Approximate counting with a floating-point counter

Miklós Csűrös

Department of Computer Science and Operations Research, University of Montréal.
csuros@iro.umontreal.ca

Abstract. When many objects are counted simultaneously in large data streams, as in the course of network traffic monitoring, or Webgraph and molecular sequence analyses, memory becomes a limiting factor. Robert Morris [*Communications of the ACM*, 21:840–842, 1978] proposed a probabilistic technique for approximate counting that is extremely economical. The basic idea is to increment a counter containing the value X with probability 2^{-X}. As a result, the counter contains an approximation of $\lg n$ after n probabilistic updates, stored in $\lg\lg n$ bits. Here we revisit the original idea of Morris. We introduce a binary floating-point counter that combines a d-bit significand with a binary exponent, stored together on $d + \lg\lg n$ bits. The counter yields a simple formula for an unbiased estimation of n with a standard deviation of about $0.6 \cdot n2^{-d/2}$.
We analyze the floating-point counter's performance in a general framework that applies to any probabilistic counter. In that framework, we provide practical formulas to construct unbiased estimates, and to assess the asymptotic accuracy of any counter.

1 Introduction

Motivation

An elementary information-theoretic argument shows that $\lceil \lg(n+1) \rceil$ bits are necessary to represent integers between 0 and n (lg denotes binary logarithm throughout the paper). Counting thus takes logarithmic space. Certain applications need to be more economical because they need to maintain many counters simultaneously while, say, tracking patterns in large data streams. Such is the case typically in network processors and embedded systems, where time and space constraints impose strict requirements on computational solutions. For instance, when measuring network traffic, per-packet updates may occur more often than allowed by Dynamic RAM access times. Faster, Static RAMs are prohibitively expensive for storing exact counters [1, 2].

Memory may become a limiting factor even on mainstream desktop computers in demanding applications such as Webgraph analyses [3, 4]. Numerous bioinformatics studies also require space-efficient solutions when searching for recurrent motifs in protein and DNA sequences. These frequent sequence motifs are associated with mobile, structural, regulatory or other functional elements, and have been studied since the first molecular sequences became available [5]. Some recent studies have concentrated on patterns involving long oligonucleotides, i.e., "words" of length 16–40 over the 4-letter DNA alphabet, revealing potentially novel regulatory features [6, 7], and general characteristics of copying processes in genome evolution [8, 9]. Hashtable-based indexing techniques [10] used in homology search and genome assembly procedures also rely on counting in order to identify repeating sequence patterns. In these applications, billions of counters need to be handled, making implementations difficult in mainstream computing environments. The need for many counters is aggravated by the fact that the counted features often have heavy-tailed frequency distributions [8, 4, 9], and there is thus no "typical" size for individual counters that could guide the memory allocation at the outset. As a numerical example, consider a study [8] of the 16-mer distribution in the human genome sequence, which has a length surpassing three billion. More than four billion (4^{16}) different words need to be counted, and the counter values span more than sixteen binary magnitudes even though the average 16-mer occurs only once or twice.

The idea of approximate counting

One way to greatly reduce memory usage is to relax the requirement of exact counts. Namely, approximate counting to n is possible using $\lg\lg n + O(1)$ bits with probabilistic techniques [11, 12]. The idea of probabilistic counting was introduced by Morris [12]. In the simplest case, a counter is initialized as $X = 0$. The counter is incremented by one at the occurrence of an event with probability 2^{-X}. The counter is meant to track the magnitude of the true number of events. More precisely, after n events, the expected value of 2^X is exactly $(n+1)$.

A generalization of the binary Morris counter is the so-called *q-ary counter* with some $r \geq 1$ and $q = 2^{1/r}$. In such a setup, the counter is incremented with probability q^{-X}. The actual event count is estimated as $f(X)$, using the transformation

$$f(x) = \frac{q^x - 1}{q - 1} = \frac{2^{x/r} - 1}{2^{1/r} - 1}.$$

The function f yields an unbiased estimate, as $\mathbb{E}f(X) = n$ after n probabilistic updates. The accuracy of a probabilistic counting method is characterized by the variance of the estimated count. For the q-ary counter,

$$\operatorname{Var} f(X) = (q-1)\frac{n(n+1)}{2}, \tag{1}$$

which is approximately $\frac{\ln 2}{2r}n^2$ for large n and r. The parameter r governs the tradeoff between memory usage and accuracy. The counter stores X (with $n = f(X)$) using $\lg r + \lg\lg n$ bits; larger r thus increases the accuracy at the expense of higher storage costs.

Approximate counting is perhaps the simplest of many related problems where probabilistic techniques can lead to efficient solutions (e.g., estimating the number of distinct elements in a multiset of size n using $\lg\lg n + O(1)$ bits

[13]). Flajolet [11] gave a precise analysis of the q-ary counter. Kirschenhoffer and Prodinger [14] performed the same analysis using Rice's method instead of the Mellin transform. Kruskal and Greenberg [15] analyzed approximate counting in a general framework (see Section 2) from a Bayesian viewpoint, assuming a known distribution for the true count.

Approximate counting revisited

The main goal of this study is to introduce a novel algorithm for approximate counting. Our *floating-point counter* is defined with the aid of a design parameter $M = 2^d$, where d is a nonnegative integer. As we discuss later, M determines the tradeoff between memory usage and accuracy, analogously to parameter r of the q-ary counter. The procedure relies on a uniform random bit generator RandomBit(). Algorithm FP-Increment below shows the incrementation procedure for a floating-point counter, initialized with $X = 0$. Notice that the first M updates are deterministic.

```
  FP-Increment($X$)                              // returns new value of $X$
1 set $t \leftarrow \lfloor X/M \rfloor$                    // bitwise right shift by $d$ positions
2 while $t > 0$ do
3     if RandomBit() $= 1$ then return $X$
4     set $t \leftarrow t - 1$
5 return $X + 1$
```

The counter value $X = 2^d \cdot t + u$, where u denotes the lower d bits, is used to estimate the actual count $f(X) = (M + u) \cdot 2^t - M$. The estimate reaches n with $d + \left\lceil \lg\lg \frac{n+M}{M-1/2} \right\rceil$ bits. The estimate's standard deviation is $\frac{c}{\sqrt{M}} n$ where c fluctuates between about 0.58 and 0.61 asymptotically (see Corollary 3 for a precise characterization). The random updates in the floating-point counter occur with exact integer powers 2^{-i}, and such random values can be generated using an average of 2 random bits. Specifically, the FP-Increment procedure uses an expected number of $\left(2 - \frac{t}{2^t - 1}\right)$ calls to RandomBit().

Notice that a q-ary counter with $r = M$ has asymptotically the same memory usage, and a standard deviation of about $\frac{0.59}{\sqrt{r}} n$ (see Eq. (1)). Our algorithm thus has similar memory usage and accuracy as q-ary counting. The floating-point counter is more advantageous in two aspects. First, the updates at the beginning are deterministic, i.e., small values are exactly represented with convenience. But more importantly, our counter can be implemented with a few elementary integer and bitwise operations, whereas a q-ary counter needs floating-point arithmetic (for random increments with irrational probabilities and the decoding by f) which may not be available on a specialized processor.

The rest of the paper is organized as follows. In order to quantify the performance of floating-point counters, we found it fruitful to develop a general analysis of probabilistic counting, which is of independent mathematical interest. Section 2 presents the general results. First, Theorem 1 shows that every probabilistic counting method has a unique unbiased estimator f with $\mathbb{E}f(X) = n$

after n probabilistic updates. Second, Theorem 2 shows that the accuracy of any such method is computable directly from the counter value. Finally, Theorem 3 gives relatively simple upper and lower bounds on the asymptotic accuracy of the unbiased estimator. Section 3 presents floating-point counters in detail, and mathematically characterizes their utility by relying on the results of Section 2. Section 3 further illustrates the theoretical analyses with simulation experiments. The proofs of the theorems are given in the Appendix, which can be safely skipped on first reading.

2 Probabilistic counting

For a formal discussion of probabilistic counting, consider the Markov chain formed by the successive counter values.

Definition 1. *A* counting chain *is a Markov chain* $(X_n : n = 0, 1, \ldots)$ *with*

$$X_0 = 0; \tag{2a}$$

$$\mathbb{P}\Big\{X_{n+1} = k+1 \,\Big|\, X_n = k\Big\} = q_k \tag{2b}$$

$$\mathbb{P}\Big\{X_{n+1} = k \,\Big|\, X_n = k\Big\} = 1 - q_k, \tag{2c}$$

where $0 < q_k \le 1$ *are the transition probabilities defining the counter.*

It is a classic result associated with probabilities in pure-birth processes [16] that the *n-step probabilities* $p_n(k) = \mathbb{P}\{X_n = k\}$ are computable by a simple recurrence (see Equations (12a–12b) later). In case of probabilistic counting, we want to infer n from the value of X_n alone through a computable function f. A given probabilistic counting method is defined by the transition probabilities and the function f. As we will see later (Theorem 1), the transition probabilities determine a unique function f that gives an unbiased estimate of the update count n.

Definition 2. *A function* $f : \mathbb{N} \mapsto \mathbb{N}$ *is an* unbiased count estimator *for a given counting chain if and only if* $\mathbb{E}f(X_n) = n$ *holds for all* $n = 0, 1, \ldots$.

In the upcoming discussions, we assume that the probabilistic counting method uses an unbiased count estimator f. The merit of a given method is gauged by its dispersion, as defined below.

Definition 3. *The* dispersion *of the counter is the coefficient of variation* $A_n = \frac{\sqrt{\operatorname{Var} f(X_n)}}{\mathbb{E}f(X_n)}$.

The theorems below provide an analytical framework for evaluating probabilistic counters. Theorem 1 shows that the unbiased estimator is uniquely defined by a relatively simple expression involving the transition probabilities. Theorem 2 shows that the uncertainty of the estimate can be determined directly from the counter value. Theorem 3 gives a practical bound on the asymptotic dispersion of the counter.

Theorem 1. *The function*

$$f(0) = 0 \tag{3a}$$

$$f(k) = \frac{1}{q_0} + \frac{1}{q_1} + \ldots + \frac{1}{q_{k-1}}. \qquad \{k > 0\} \tag{3b}$$

uniquely defines the unbiased count estimator f for any given set of transition probabilities $(q_k\colon k = 0, 1, \ldots)$. Hence, for any given counting chain, we can determine efficiently an unbiased estimator. Conversely, we can compute the counting chain for any given $f\colon \mathbb{N} \mapsto [0, \infty)$ provided that $f(0) = 0$ and $\forall k\colon f(k+1) \geq f(k) + 1$.

Definition 4. *The* variance function *for a given counting chain is defined by*

$$g(0) = 0 \tag{4a}$$

$$g(k) = \frac{1 - q_0}{q_0^2} + \frac{1 - q_1}{q_1^2} + \cdots + \frac{1 - q_{k-1}}{q_{k-1}^2} \qquad \{k > 0\} \tag{4b}$$

Theorem 2 below shows that the dispersion is computable directly from the counter value for any counting chain. The statement has a practical relevance: g quantifies the uncertainty of the estimate f, The variance function is used later to evaluate the asymptotic dispersion (see Theorem 3).

Theorem 2. *The variance function g of Definition 4 provides an unbiased estimate for the variance of f from Theorem 1. Specifically,*

$$\operatorname{Var} f(X_n) = \mathbb{E} g(X_n) \tag{5}$$

holds for all $n \geq 0$, where the moments refer to the space of n-step probabilities.

Theorem 3 is the last main result of this section. The statement relates the asymptotics of the variance function, the unbiased count estimator, and the counting chain's dispersion.

Theorem 3. *Let A_n be the dispersion of Definition 3, and let*

$$B_k = \frac{\sqrt{g(k)}}{f(k)}. \tag{6}$$

Let $\liminf_{k\to\infty} B_k = \mu$. Suppose that $\limsup_{k\to\infty} B_k = \lambda < 1$ (and, thus, $\mu < 1$). Then

$$\frac{\mu}{\sqrt{1-\mu^2}} \leq \liminf_{n\to\infty} A_n; \qquad \limsup_{n\to\infty} A_n \leq \frac{\lambda}{\sqrt{1-\lambda^2}}. \tag{7}$$

Example. Consider the case of a q-ary counter, where $q_i = q^{-i}$ with some $q > 1$. Theorem 1 automatically gives the unbiased count estimator $f(k) = \sum_{i=0}^{k-1} q_i^{-1} = \frac{q^k-1}{q-1}$. Theorem 2 yields the variance function $g(k) = \sum_{i=0}^{k-1} \big(q_i^{-2} - q_i^{-1}\big) = \frac{q^{2k}-1}{q^2-1} - \frac{q^k-1}{q-1}$. In order to use Theorem 3, observe that $\mu = \lambda = \lim_{k\to\infty} \sqrt{\frac{g(k)}{f^2(k)}} = \sqrt{\frac{q-1}{q+1}} < 1$. Therefore, we obtain the known result [11] that $\lim_{n\to\infty} A_n = \sqrt{\frac{\lambda^2}{1-\lambda^2}} = \sqrt{\frac{q-1}{2}}$.

Assessing accuracy. Interestingly, $f(k)$ corresponds to the expected value of the hitting time: $f(k) = \mathbb{E}\min\{n\colon X_n = k\}$. In other words, assuming that the chain $(X_n\colon n = 0, 1, \dots)$ "just reached" k yields an unbiased estimate. Theorem 3 links the *estimator's* coefficient of variation A_n and the ratio B_k of Eq. (6) between the *estimate's* inferred standard deviation $\sqrt{g(k)}$ and value. For any given $k = X_n$, B_k reasonably quantifies the uncertainty of the hitting time estimate $\hat{n} = f(k)$. Equation (7) states that the estimator has a slightly larger dispersion than that. In the example of the q-ary counter, the estimator's dispersion is asymptotically $\sqrt{(q+1)/2}$ times larger than B_k.

Counting in $(1+c)\lg\lg n + O(1)$ ***space.*** Theorem 1 confirms the intuition that the transition probabilities must be exponentially decreasing in order to achieve storage on $\lg\lg n + O(1)$ bits. Otherwise, with subexponential $q_k^{-1} = 2^{o(k)}$, one would have $f(k) = 2^{o(k)}$, leading to $\lg n = o(k)$. In other words, $\lg\lg n + O(1)$ bits would not suffice to store the counter value k. It is possible, however, to devise other exotic counting schemes with $O(\log\log n)$ memory usage by "reverse engineering" the memory requirements. Consider the *subexponential counter* $\mathsf{SubExp}(\beta)$ for $0 < \beta < 1$ defined by the transition probabilities $q_k = 1/\bigl(\exp((k+1)^\beta) - \exp(k^\beta)\bigr)$, yielding the unbiased estimator $f(k) = \exp(k^\beta) - 1$, and a memory requirement of $\beta^{-1}\lg\ln n + O(1)$. Using continuous approximations, $q_k^{-1} \approx \beta k^{\beta-1}\exp(k^\beta)$, and $g(k) \approx \int_{x=1}^{k} \beta^2 x^{2\beta-2}\exp(2x^\beta)\,dx$. Now, $B_k = \sqrt{g(k)}/f(k) \sim k^{(\beta-1)/2}\sqrt{\beta/2}$, since

$$\lim_{k\to\infty} \frac{B_k^2}{\frac{\beta}{2}k^{\beta-1}} \approx \lim_{k\to\infty} \frac{\int_{x=1}^{k} \beta^2 x^{2\beta-2} e^{2x^\beta}\,dx}{(e^{k^\beta}-1)^2 \frac{\beta}{2}k^{\beta-1}} = 1$$

by l'Hospital's rule. Accodingly (plug in k with $f(k) = n$, i.e., $k = \ln^{1/\beta}(n+1)$), we conjecture that the $(1+c)\lg\ln n + O(1)$ memory usage of the $\mathsf{SubExp}((c+1)^{-1})$ counter entails that A_n goes to 0 at a slow speed of $O\bigl(\sqrt{1/\log^c n}\bigr)$ (see illustration in Appendix).

Bayesian count estimation. The same general framework for approximate counting is used by Kruskal and Greenberg [15] in a Bayesian setting. In particular, they are interested in the case when the true count N is a random variable. The actual count is estimated as $\phi(k) = \mathbb{E}\bigl[N \bigm| X_N = k\bigr]$. They further consider the case of geometrically distributed N, as well as an improper prior $\mathbb{P}\{N = n\} = \epsilon$. For both distributions, $\phi(k) = \mathbb{E}\bigl[N \bigm| X_N = k\bigr] = f(k+1) - 1$ and $\mathrm{Var}\bigl[N \bigm| X_N = k\bigr] = g(k+1)$ where f and g are defined in Theorems 1 and 2 above. Kruskal and Greenberg motivate their results by suggesting that the unbiased count estimator may be difficult to find since in order to verify the condition $\mathbb{E}f(X_n) = n$, it may be complicated to calculate the expectation for an arbitrary nonlinear function f. Theorem 1 shows that this is not so: the unbiased estimator has a fairly simple form for any counting chain.

3 Floating-point counters

The counting chain for a floating-point counter is defined using a design parameter $M = 2^d$ with some nonnegative integer d:

$$\mathbb{P}\Big\{X_{n+1} = k+1 \;\Big|\; X_n = k\Big\} = 2^{-\lfloor k/M \rfloor}; \tag{8a}$$

$$\mathbb{P}\Big\{X_{n+1} = k \;\Big|\; X_n = k\Big\} = 1 - 2^{-\lfloor k/M \rfloor}. \tag{8b}$$

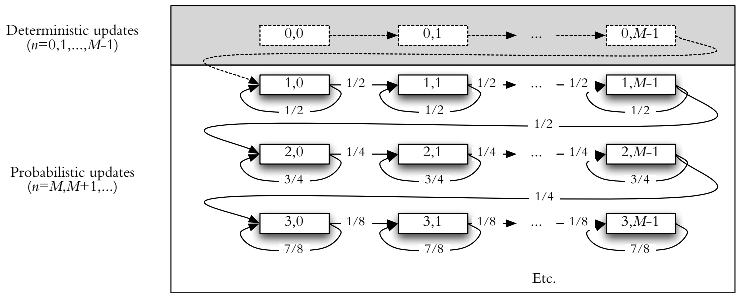


Fig. 1. States of the counting Markov chain. Each state is labeled with a pair (t, u), where $(u + M)$ are the most significant digits and t is the number of trailing zeros for the true count.

Figure 1 illustrates the states of the floating-point counter. The counter's designation becomes apparent from examining the binary representation of the counter value k. Write $k = Mt + u$ with $t = \lfloor k/M \rfloor$ and $u = k \bmod M$; i.e., u corresponds to the lower d bits of k, and t corresponds to the remaining upper bits. The pair (t, u) is essentially a floating-point representation of the true count n, where t is the exponent, and u is a d-bit significand without the hidden bit for the leading '1.' More precisely, Theorem 1 applies with $q_k = 2^{-\lfloor k/M \rfloor}$, and leads to the following corollary.

Corollary 1. *The unbiased estimator for $k = Mt + u$ is*

$$f(k) = f(t, u) = (M + u)2^t - M. \tag{9}$$

Theorem 2 yields the following variance function.

Corollary 2. *The variance function for the floating-point counter is*

$$g(k) = g(t, u) = \left(\frac{M}{3} + u\right)4^t - (M + u)2^t + \frac{2}{3}M. \tag{10}$$

Combining Corollaries 1 and 2, we get the following bounds on the dispersion.

Corollary 3. *The dispersion of the floating-point counter is asymptotically bounded as*

$$\sqrt{\frac{1}{3M-1}} \leq \liminf_{n\to\infty} A_n; \qquad \limsup_{n\to\infty} A_n \leq \sqrt{\frac{3}{8M-3}}.$$

Proof. By Equations (9) and (10), we have $\lim_{t\to\infty} \frac{g(t,u)}{f^2(t,u)} = \frac{\frac{M}{3}+u}{(M+u)^2}$. Considering the extreme values at $u = 0$ and $u = \lfloor M/3 \rfloor$ or $u = \lceil M/3 \rceil$, respectively:

$$\mu^2 = \liminf_{k\to\infty} \frac{g(k)}{f^2(k)} = \frac{1}{3M}; \quad \lambda^2 = \limsup_{k\to\infty} \frac{g(k)}{f^2(k)} \leq \lim_{t\to\infty} \frac{g(t, \frac{M}{3})}{f^2(t, \frac{M}{3})} = \frac{3}{8M}. \quad (11)$$

Plugging these limits into Theorem 3 leads to the Corollary. □

For large $M = 2^d$, the bounds of Corollary 3 become $\limsup_{n\to\infty} A_n \lessapprox 2^{-d/2}\sqrt{3/8} \approx 0.612 \cdot 2^{-d/2}$ and $\liminf_{n\to\infty} A_n \gtrapprox 2^{-d/2}\sqrt{1/3} \approx 0.577 \cdot 2^{-d/2}$.

The dispersion is thus comparable to the dispersion of a q-ary counter with $q = 2^{2^{-d}}$, which is approximately $2^{-d/2}\sqrt{0.5 \cdot \ln 2} \approx 0.589 \cdot 2^{-d/2}$. The memory requirements of the two counters are equivalent: in order to count up to $n = f(k)$, $d + \lg\lg n$ bits are necessary.

Scaled floating-point counters. A similar floating-point technique of approximate counting is described by Stanojević [2]. In his solution (for measuring network traffic with per-packet counter updates), a so-called "small active counter" splits a fixed number of ℓ bits into a d-bit significand u and an $(\ell - d)$-bit exponent t, in successive scale regimes $\sigma = 1, 2, \ldots$. The scaling factor σ is stored separately. (The same scaling factor is used for a whole set of counters, but the analysis is performed for a single counter only.) The counter is incremented with a probability of $2^{-\sigma\lfloor X/M \rfloor}$ at scaling σ, along with suitable adjustements to X at scale changes. Stanojević infers the appropriate unbiased estimator, and calculates the coefficient of variation (B_k of Theorem 3) for the hitting time $f(k) = \min\{n \colon X_n = k\}$ at a fixed scaling σ. Consider the scaled version of the floating-point counter where $q_k = q^{-\lfloor k/M \rfloor}$ with some $q > 1$ ($q = 2$ in the basic version and $q = 2^\sigma$ with scaling σ). By Theorems 1 and 2,

$$f(k) = f(t,u) = \Big(\frac{M}{q-1} + u\Big)q^t - \frac{M}{q-1}$$
$$g(k) = g(t,u) = \Big(\frac{M}{q^2-1} + u\Big)q^{2t} - \Big(\frac{M}{q-1} + u\Big)q^t + M\frac{q}{q^2-1}$$

with $t = \lfloor k/M \rfloor$ and $u = M\{k/M\} = k - tM$. Consequently, $\lg M + \lg\log_q n$ bits are needed to reach n. The extremes for B_k are attained with $u = 0$, and with $u = \lfloor M/(q+1) \rfloor$ or $u = \lceil M/(q+1) \rceil$:

$$\mu = \liminf_{k\to\infty} B_k = \lim_{t\to\infty} \frac{\sqrt{g(t,0)}}{f(t,0)} = M^{-1/2} \cdot \sqrt{\frac{q-1}{q+1}}$$
$$\lambda = \limsup_{k\to\infty} B_k \leq \lim_{t\to\infty} \frac{\sqrt{g(t, M/(q+1))}}{f(t, M/(q+1))} = M^{-1/2} \cdot \sqrt{\frac{q^2-1}{4q}}$$

By Theorem 3, the dispersion is bounded as

$$\sqrt{\frac{q-1}{(q+1)M-(q-1)}} \le \liminf_{n\to\infty} A_n; \qquad \limsup_{n\to\infty} A_n \le \sqrt{\frac{q^2-1}{4qM-(q^2-1)}}.$$

Simulations. Figures 2 and 3 compare the performance of the floating-point counters with equivalent base-q counters in simulation experiments. The equivalence is manifest on Figure 2 that illustrates the trajectories of the estimates by the different counters. Figure 3 plots statistics about the estimates across multiple experiments: the estimators are clearly unbiased, and the two counters display the same accuracy.

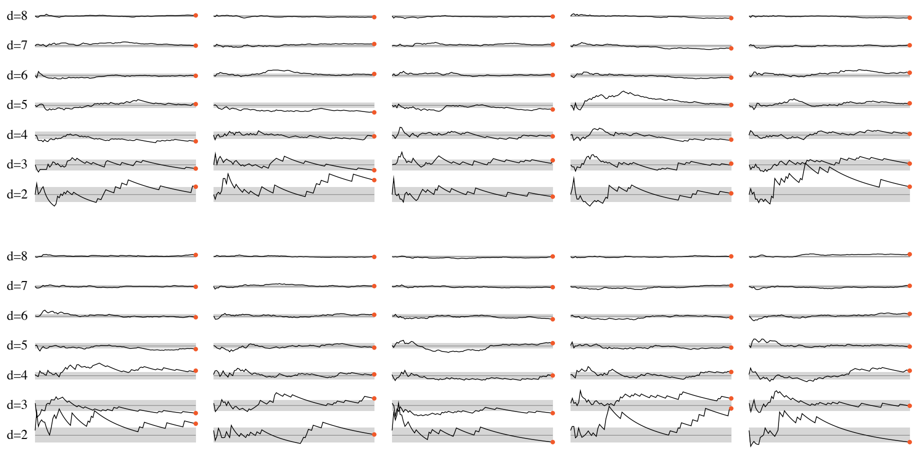
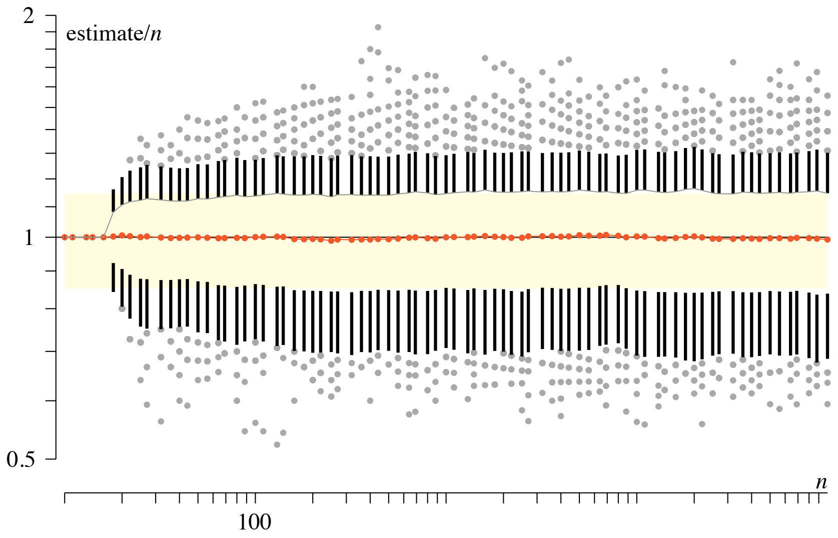


Fig. 2. Error trajectories for floating-point counters (**top**) and q-ary counters (**bottom**). Each trajectory follows the the appropriate counting chain in a random simulated run. The lines trace the relative error $(f(X_n)-n)/n$ for floating-point counters with d-bit mantissa, and comparable q-ary counters with $q = 2^{1/r}$ where $r = 2^d$. The shaded areas indicate a relative error of $\pm 0.59 \cdot 2^{-d/2}$. The dots at the end of the trajectories denote the final value for $n = 100000$.

Acknowledgments

I am very grateful to Philippe Flajolet for valuable suggestions on improving previous versions of the manuscript. This work has been supported by a grant from the Natural Sciences and Engineering Research Council of Canada.



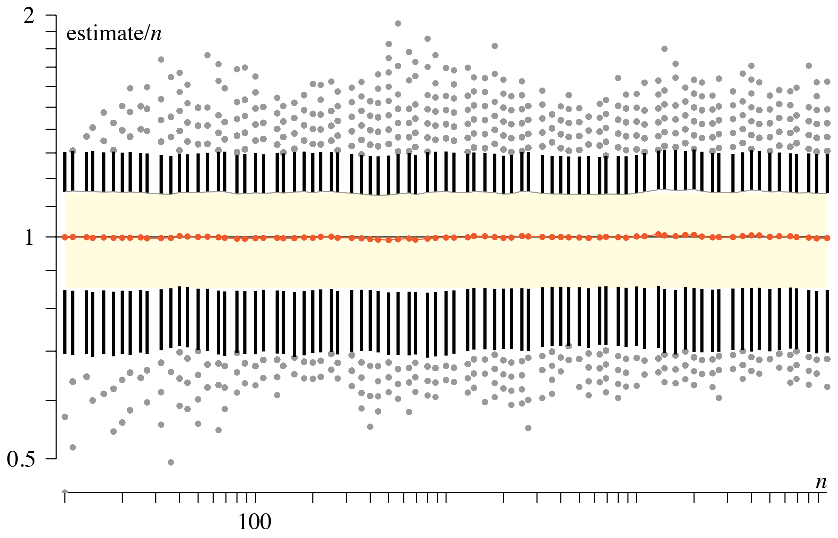


Fig. 3. Distribution of the estimates for a floating-point counter (**top**) and a comparable q-ary counter (**bottom**). Each plot depicts the result of 1000 experiments, in which a floating-point counter with $d = 4$-bit mantissa, and a q-ary counter with $q = 2^{1/16}$ were run until $n = 100,000$. The dots in the middle follow the averages; the black segments depict the standard deviations (for each σ, they are of length σ spaced at σ from the average), and grey dots show outliers that differ by more than $\pm 2\sigma$ from the average. The shading highlights the asymptotic dispersion of the q-ary counter ($\approx 0.59 \cdot 2^{-d/2}$).

References

1. Estan, C., Varghese, G.: New directions in traffic measurement and accounting: Focusing on the elephants, ignoring the mice. ACM Transactions on Computer Systems **21**(3) (2003) 270–313
2. Stanojević, R.: Small active counters. In: 26th IEEE International Conference on Computer Communications (INFOCOM), Proceedings, IEEE (2007) 2153–2161
3. Becchetti, L., Castillo, C., Donato, D., Baeza-Yates, R., Leonardi, S.: Link analysis for Web spam detection. ACM Transactions on the Web **2**(1) (2008) 2
4. Donato, D., Laura, L., Leonardi, S., Millozzi, S.: Large-scale properties of the Webgraph. European Physics Journal B **38** (2004) 239–243
5. Karlin, S.: Statistical signals in bioinformatics. Proceedings of the National Academy of Sciences of the USA **102**(38) (2005) 13355–13362
6. Jones, N.C., Pevzner, P.A.: Comparative genomics reveals unusually long motifs in mammalian genomes. Bioinformatics **22**(14) (2006) e236–e242
7. Rigoutsos, I., Huynh, T., Miranda, K., Tsirigos, A., McHardy, A., Platt, D.: Short blocks from the noncoding parts of the human genome have instances within nearly all known genes and relate to biological processes. Proceedings of the National Academy of Sciences of the USA **103**(17) (2006) 6605–6610
8. Csűrös, M., Noé, L., Kucherov, G.: Reconsidering the significance of genomic word frequencies. Trends in Genetics **23**(11) (2007) 543–546 Preprint available as q-bio/0609022 at arXiv.org.
9. Sindi, S.S., Hunt, B.R., Yorke, J.A.: Duplication count distributions in DNA sequences. Physical Review E (Statistical, Nonlinear, and Soft Matter Physics) **78**(6) (2008) 061912
10. Ning, Z., Cox, A.J., Mullikin, J.C.: SSAHA: A fast search method for large DNA databases. Genome Research **11**(10) (2001) 1725–1729
11. Flajolet, P.: Approximate counting: A detailed analysis. BIT **25** (1985) 113–134
12. Morris, R.: Counting large number of events in small registers. Communications of the ACM **21**(10) (1978) 840–842
13. Flajolet, P., Fusy, É., Gandouet, O., Meunier, F.: HyperLogLog: the analysis of a near-optimal cardinality estimation algorithm. In Jacquet, P., ed.: Conference on Analysis of Algorithms (AofA). Discrete Mathematics and Theoretical Computer Science Proceedings (2007) 127–146
14. Kirschenhoffer, P., Prodinger, H.: Approximate counting: An alternative approach. RAIRO Theoretical Informatics and Applications **25** (1991) 43–48
15. Kruskal, J.B., Greenberg, A.G.: A flexible way of counting large numbers approximately in small registers. Algorithmica **6** (1991) 590–596
16. Karlin, S., Taylor, H.M.: A First Course in Stochastic Processes. Second edn. Academic Press, San Diego, Cal. (1975)

A Subexponential counters

For the subexponential counter $\mathsf{SubExp}\big((c+1)^{-1}\big)$ with $c > 0$, the estimate's dispersion is asymptotically decreasing as

$$B_k \sim \frac{k^{-\frac{c}{2(c+1)}}}{\sqrt{2(c+1)}}.$$

Accordingly, we conjecture that $A_n = \Theta(\log^{-c/2} n)$. Figure 4 illustrates the convergence for the subexponential counter $\mathsf{SubExp}(1/2)$ (i.e., $c = 1$) at a speed of $O(\log^{-1/2} n)$. The figure also shows that the theoretical advantage does not immediately translate into a practical one: the accuracy of the $\mathsf{SubExp}(1/2)$ is better than a q-ary or floating-point counter only at very large n.

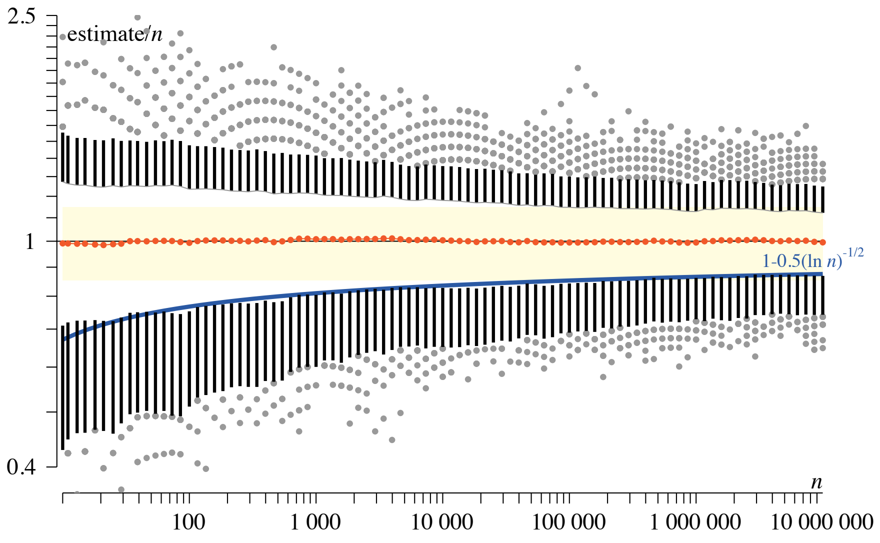


Fig. 4. Distribution of the estimates for the $\mathsf{SubExp}(1/2)$ counter with $f(k) = \exp(\sqrt{k}) - 1$. The plot depicts the result of 1000 experiments, in which the counter was run for up to 10 million steps. The dots in the middle follow the averages; the black segments depict the standard deviations (for each σ, they are of length σ spaced at σ from the average), and grey dots show outliers that differ by more than $\pm 2\sigma$ from the average. The solid line illustrates the fit for the asymptotic decrease $A_n \sim \frac{1}{2\sqrt{\ln n}}$. For a practical comparison, the shading in the middle highlights the asymptotic dispersion of the 4-bit floating-point counter (≈ 0.15).

B Proofs

In what follows, we use the shorthand notation

$$p_n(k) = \mathbb{P}\{X_n = k\}$$

for the n-step probabilities. By (2), $p_0(0) = 1$, and the recurrences

$$p_{n+1}(0) = (1 - q_0)p_n(0) \tag{12a}$$
$$p_{n+1}(k) = (1 - q_k)p_n(k) + q_{k-1}p_n(k-1) \qquad \{k > 0\} \tag{12b}$$

hold for all $n \geq 0$.

Lemma 1. *The unbiased estimator is unique.*

Proof. Since $\mathbb{E}f(0) = 0$ is imposed, and $X_0 = 0$ with certainty, $f(0) = 0$. For all n, $\mathbb{P}\{X_n > n\} = 0$, so

$$\mathbb{E}f(X_n) = \sum_{k=0}^{n} p_n(k)f(k) = n.$$

Thus, for all $n > 0$,

$$f(n) = \frac{n - \sum_{k=0}^{n-1} p_n(k)f(k)}{p_n(n)} = \frac{n - \sum_{k=0}^{n-1} p_n(k)f(k)}{q_0q_1 \cdots q_{n-1}},$$

which shows that $f(n)$ is uniquely determined by $f(0), \ldots, f(n-1)$ and the n-step probabilities. □

Proof (of Theorem 1). Define the durations $L_k(n) = \sum_{i=0}^{n-1}\{X_i = k\}$, i.e., the number of times $X_i = k$ for $i < n$. Define also $L_k = \lim_{n\to\infty} L_k(n) = \sum_{i=0}^{\infty}\{X_i = k\}$. Clearly, $\mathbb{E}L_k = 1/q_k$. By the linearity of expectations,

$$\begin{aligned}
\mathbb{E}L_k &= \mathbb{E}L_k(n) + \mathbb{E}\sum_{i=n}^{\infty}\{X_i = k\} \\
&= \mathbb{E}L_k(n) + \mathbb{E}\Big[\sum_{i=n}^{\infty}\{X_i = k\} \Bigm| X_n \leq k\Big]\mathbb{P}\{X_n \leq k\} \\
&= \mathbb{E}L_k(n) + \mathbb{P}\{X_n \leq k\}\mathbb{E}L_k,
\end{aligned}$$

where we used the memoryless property of the geometric distribution in the last step. Consequently,

$$\mathbb{E}L_k(n) = \frac{\mathbb{P}\{X_n > k\}}{q_k}. \tag{13}$$

Now,

$$\mathbb{E}\sum_{k=0}^{\infty} L_k(n) = \sum_{k=0}^{\infty} \mathbb{P}\{X_n > k\}\frac{1}{q_k} = \sum_{k=0}^{n} p_n(k)\sum_{i=0}^{k-1}\frac{1}{q_i} = \sum_{k=0}^{n} p_n(k)f(k) = \mathbb{E}f(X_n).$$

Since $\sum_{k=0}^{\infty} L_k(n) = n$, we have $\mathbb{E}f(X_n) = n$ for all n. By Lemma 1, no other function f has the same property. □

Proof (of Theorem 2). By (12), for all $n \geq 0$,

$$\begin{aligned}
\mathbb{E}f^2(X_{n+1}) &= \sum_{k=0}^{n+1} p_{n+1}(k) f^2(k) \\
&= \sum_{k=0}^{n} (1-q_k) p_n(k) f^2(k) + \sum_{k=1}^{n+1} q_{k-1} p_n(k-1) f^2(k) \\
&= \mathbb{E}f^2(X_n) - \sum_{k=0}^{n} q_k p_n(k) f^2(k) + \sum_{k=1}^{n+1} q_{k-1} p_n(k-1) \big(f(k-1) + q_{k-1}^{-1}\big)^2 \\
&= \mathbb{E}f^2(X_n) + 2\sum_{k=0}^{n} p_n(k) f(k) + \sum_{k=0}^{n} p_n(k) q_k^{-1} \\
&= \mathbb{E}f^2(X_n) + 2n + \sum_{k=0}^{n} p_n(k) q_k^{-1}.
\end{aligned}$$

Since $\operatorname{Var} f(X_n) = \mathbb{E}f^2(X_n) - \Big(\mathbb{E}f(X_n)\Big)^2 = \mathbb{E}f^2(X_n) - n^2$,

$$\operatorname{Var} f(X_{n+1}) = \operatorname{Var} f(X_n) + \sum_{k=0}^{n} p_n(k) q_k^{-1} - 1. \tag{14}$$

By (4) and (12),

$$\begin{aligned}
\mathbb{E}g(X_{n+1}) &= \sum_{k=0}^{n+1} p_{n+1}(k) g(k) \\
&= \mathbb{E}g(X_n) - \sum_{k=0}^{n} q_k p_n(k) g(k) + \sum_{k=1}^{n+1} q_{k-1} p_n(k-1) \Big(g(k-1) + \frac{1-q_{k-1}}{q_{k-1}^2}\Big) \\
&= \mathbb{E}g(X_n) + \sum_{k=0}^{n} p_n(k) \frac{1-q_k}{q_k}. \\
&= \mathbb{E}g(X_n) + \sum_{k=0}^{n} p_n(k) q_k^{-1} - 1.
\end{aligned}$$

By (14), $\operatorname{Var} f(X_{n+1}) - \operatorname{Var} f(X_n) = \mathbb{E}g(X_{n+1}) - \mathbb{E}g(X_n)$ holds for all $n \geq 0$. Since $\operatorname{Var} f(X_0) = \mathbb{E}g(X_0) = 0$, $\operatorname{Var} f(X_n) = \mathbb{E}g(X_n)$ holds for all n. □

Proof (of Theorem 3). Define

$$W_n = \frac{\operatorname{Var} f(X_n)}{\mathbb{E}f^2(X_n)} = \frac{\sum_{k=0}^{\infty} p_n(k) \cdot g(k)}{\sum_{k=0}^{\infty} p_n(k) \cdot f^2(k)}.$$

Since $\operatorname{Var} f(X_n) = \mathbb{E}f^2(X_n) - \mathbb{E}^2 f(X_n)$ and $\mathbb{E}f(X_n) = n$,

$$\begin{aligned} W_n &= \frac{\operatorname{Var} f(X_n)}{\operatorname{Var} f(x_n) + n^2} \\ A_n^2 &= \frac{\operatorname{Var} f(X_n)}{n^2} = \frac{W_n}{1 - W_n}. \end{aligned}$$

Now, $0 \leq W_n \leq 1$, and the monotonicity of $\varphi(z) = \frac{z}{1-z}$ on $z \in [0, 1)$ implies that

$$\liminf_{n\to\infty} A_n = \sqrt{\frac{\liminf_{n\to\infty} W_n}{1 - \liminf_{n\to\infty} W_n}} \tag{15a}$$

$$\limsup_{n\to\infty} A_n = \sqrt{\frac{\limsup_{n\to\infty} W_n}{1 - \limsup_{n\to\infty} W_n}}, \tag{15b}$$

where $\limsup_{n\to\infty} A_n = \infty$ or $\lim_{n\to\infty} A_n = \infty$ is allowed with $\limsup_{n\to\infty} W_n = 1$ or $\lim_{n\to\infty} W_n = 1$.

Let $\epsilon > 0$ be an arbitrary threshold. By the definition of λ, there exists K such that

$$\frac{g(k)}{f^2(k)} < (1 + \epsilon)\lambda^2$$

for all $k > K$. Therefore,

$$\begin{aligned} W_n &= \frac{\sum_{k=0}^{K} p_n(k)g(k) + \sum_{k>K} p_n(k) \cdot g(k)}{\sum_{k=0}^{K} p_n(k)f^2(k) + \sum_{k>K} p_n(k)f^2(k)} \\ &< \frac{\sum_{k=0}^{K} p_n(k)g(k) + (1 + \epsilon)\lambda^2 \sum_{k>K} p_n(k)f^2(k)}{\sum_{k>K} p_n(k)f^2(k)} \\ &= (1 + \epsilon)\lambda^2 + \frac{\sum_{k=0}^{K} p_n(k)g(k)}{\sum_{k>K} p_n(k)f^2(k)}. \end{aligned}$$

Since $q_k > 0$ for all k, $\lim_{n\to\infty} p_n(k) = 0$ for all k. Consequently, $\lim_{n\to\infty} \sum_{k=0}^{K} p_n(k)g(k) = 0$. As $\lim_{n\to\infty} \sum_{k>K} p_n(k)f^2(k) = \infty$, there exists N such that

$$W_n < (1 + 2\epsilon)\lambda^2 \qquad \text{for all } n > N. \tag{16}$$

As ϵ is arbitrary,

$$\limsup_{n\to\infty} W_n \leq \lambda^2$$

must hold, and (15b) implies $\limsup_{n\to\infty} A_n \leq \sqrt{\frac{\lambda^2}{1-\lambda^2}}$.

The lower bound is proven analogously. Let $\epsilon > 0$ be an arbitrary threshold. Let K be such that $\frac{g(k)}{f^2(k)} > (1 - \epsilon)\mu^2$ for all $k > K$. So,

$$W_n > \frac{(1 - \epsilon)\mu^2 \sum_{k>K} p_n(k)f^2(k)}{\sum_{k=0}^{K} p_n(k)f^2(k) + \sum_{k>K} p_n(k)f^2(k)}.$$

For n large enough, $W_n > (1-2\epsilon)\mu^2$ holds. Since ϵ is arbitrarily small,

$$\liminf_{n\to\infty} W_n \geq \mu^2,$$

and (15a) gives $\liminf_{n\to\infty} A_n \geq \sqrt{\frac{\mu^2}{1-\mu^2}}$. $\square$