# *An optimal cardinality estimation algorithm based on order statistics and its full analysis*

Jérémie Lumbroso

*Équipe APR, LIP6/UPMC, 4 place Jussieu, F-75005 Paris, France*
*Algorithms Project, INRIA Rocquencourt, F-78153 Le Chesnay, France*

Building on the ideas of Flajolet and Martin (1985), Alon et al. (1987), Bar-Yossef et al. (2002), Giroire (2005), we develop a new algorithm for cardinality estimation, based on order statistics which, according to Chassaing and Gerin (2006), is optimal among similar algorithms. This algorithm has a remarkably simple analysis that allows us to take its *fine-tuning* and the *characterization of its properties* further than has been done until now. We prove that, asymptotically, it is *strictly unbiased* (contrarily to Probabilistic Counting, Loglog, Hyperloglog), we verify that its relative precision is about $1/\sqrt{m-2}$ when $m$ words of storage are used, and we fully characterize the limit law of the estimates it provides, in terms of gamma distribution—this is the first such algorithm for which the limit law has been established. We also develop a Poisson analysis for the pre-asymptotic regime. In this way, we are able to devise a complete algorithm, covering all cardinalities ranges from 0 to very large.

## 1 Introduction

Given a large *multiset* $S$ of size $|S|$ (the number of elements contained in $S$), we are interested in a basic quantity, the *cardinality* $n$ which is the number of *distinct* elements in $S$.

The problem of determining or estimating the cardinality of a (large) multiset is historically tied to database query optimization, but it also has a wide variety of other practical applications in many fields, ranging from data mining to network security (as is further detailed in Flajolet's survey [5]). Solving it exactly without additional knowledge on the nature of the data requires linear space. For the orders of magnitude considered, multisets containing millions, if not billions, of elements, this is impractical.

Probabilistic algorithms, which advantageously trade accuracy for speed, have proven remarkably elegant and useful: see Alon et al.'s synthesis study [1]. The algorithms we are interested in all resort to hash functions to transform the input data into what can be virtually considered as uniform random variables (see Knuth [11]). An *observable* is a function of the underlying *set* of hashed values; that is, a quantity independent of replications.

From here, two broad categories of observables have emerged, corresponding to two different ways of viewing the hashed values.

- The "continuous view", where the hashed values are seen as real numbers on the set $[0, 1]$, and only the relative order of elements is taken into account: observables studied under this model are called *order statistics*. In particular the $k$-th order statistic, for any given (constant) $k$, is the $k$-th smallest value of the underlying set. These are relevant because, for instance, the $k$-th order statistic of $n$ uniformly random variables is an observable and its expected value is $k/(n+1)$, thus it must convey some information on the cardinality $n$. Algorithms which fit into this category are due to Bar-Yossef et al. [2], and more recently to Giroire [9, 10], Chassaing and Gerin [3].

- The "discrete view", in which the hashed values are considered as strings of bits. Observables are then based on *bit-patterns*, such as the longest run of zeroes at the beginning of a string. The papers by Flajolet et al. [4, 6, 7] belong to this category, as does (in a way) *linear counting*, a special method introduced by Whang et al. in [12], but one that consumes linear space.

Algorithms based on pattern observation produce estimators that, for well understood reasons, tend to involve small periodic fluctuations. While inconsequential from a practical standpoint, these introduce a residual error which means the estimators are not totally asymptotically unbiased. Our algorithm, by virtue of being based on order statistics, does not have this limitation and is truly *asymptotically unbiased*.

Giroire's thesis [9] extensively studies several estimators based on order statistics observables; Chassaing and Gerin in [3] suggest (without a detailed analysis) that another estimator, the inverse of an arithmetic mean, is optimal—in the sense that its variance is less than that of any other estimator based on order statistics. This is the estimator which we have chosen to use in our algorithm. It gives it a relative precision which is $1/\sqrt{m-2}$ (where $m$ is the amount of memory used, see Theorem 2).

**Counting with hash functions.** Hash functions are very relevant to cardinality estimations algorithms. On the one hand, they simulate uniform random variables, which allow for relatively easy calculations; on the other hand, this simulated randomness is *reproductible*, in the sense that with a given hash function the same element will always be hashed to the same value. So in effect, hash functions allow us to study the underlying *set* (of cardinality $n$) of a multiset as though we were dealing with $n$ uniform random variables. There is an additional point: we need to choose a function of the hashed values that is insensitive to repetitions, for example, the *minimum* of all hashed values.

The minimum of $n$ uniform random variables has expectation $1/(n+1)$; taking the inverse of this minimum to estimate cardinality seems straightforward enough. Unfortunately, this alone turns out to be less than useful for two reasons: the standard deviation of this minimum is of the same order as its expectation, which means that it fluctuates a lot; concurrently, the inverse function diverges in $0$, and the inverse of the minimum has infinite expectation.

Fluctuations can be circumvented by taking the average of several experiments (here $m$ minima), each of which would be performed using a different hash function and the same multiset. While this works, the overhead of hashing *each* element of the multiset $m$ times is vastly prohibitive[(i)].

**Stochastic averaging.** As early as one of the first papers on probabilistic cardinality estimation, Flajolet and Martin introduced a technique they called *stochastic averaging* [7], to simulate $m$ different hash functions with a single hash function, by splitting the input stream into several substreams.

To this end, we assign each element a given substream uniformly at random, with the condition that every repetition of a given element must be assigned the same substream—which is why the averaging is termed *stochastic*. Because every instance of an element has to be given the same substream, we again resort to hash functions. More specifically, given the hashed value $h(x) \in (0,1)$, we use $\lfloor mh(x) \rfloor + 1$ as

[(i)] Especially as we have in mind algorithms with a precision of one or a few percent, which require to take values of $m$ in the range of thousands.

the substream, and we then take the fractional part, $(mh(x) - \lfloor mh(x) \rfloor) \in (0,1)$, as the hashed value[ii].

When $m = 2^l$ is a power of two (as we will assume in our final implementation, in Section 5), this can be done very efficiently by sampling the $l$ high bits of the hashed value of the element (these bits will then have to be discarded or they will introduce a correlation that will bias the estimate in unpredictable ways).

**Approach and results.** Our main purpose is to concurrently develop an *algorithm* for cardinality estimations and its *analysis*. The analysis here serves several purposes:

1. to develop an asymptotically unbiased estimator (Theorem 1);
2. to determine its accuracy, measured in terms of standard error (Theorem 2) and characterize the risk of error by way of the limit distribution of the estimate (Theorem 3);
3. to take into account the non-asymptotic regime so as to obtain a fully operational algorithm that provably covers all cardinality ranges from 0 to extremely large (Theorem 4 and Proposition 4.2).

The core algorithm is given in Figure 1. The mean and variance analysis (bias correction and standard error estimation) are the subject of Section 2, as is the limit distribution result. The core algorithm presents a non-asymptotical regime that is not directly usable: Section 3 details the analysis of an intermediate regime ($n$ and $m$ proportional), which is amenable to a Poisson approach and allows for the necessary corrections.

Section 4 reaps the crop. It adds a correction for the case when $n$ is very small (Subsection 4.1), develops the corrections provided by the Poisson analysis (Subsection 4.2), finally resulting in the complete algorithm, which covers the entire range of small-to-very-large values as announced.

```
CORE ALGORITHM (fully analyzed in Section 2)

Parameter: $m$ control parameter
Input: a stream $S = (s_1, \ldots, s_N)$

initialize $m$ registers $M_1$ through $M_m$ to 1
forall $x \in S$ do
    $A := h(x)$                             {hash $x$, with $h(x) \in (0,1)$}
    $j := \lfloor mA \rfloor + 1$                   {index of the substream assigned to $x$}
    $M_j := \min(M_j, mA - \lfloor mA \rfloor)$     {update the minimum of the $j$-th substream}
return $\mathcal{Z}^* = \dfrac{m(m-1)}{M_1 + \ldots + M_m}$   {estimation function}
```

**Fig. 1:** Pseudo-code for the core algorithm, introducing some notations we will use throughout this article. An extended estimation function applies either one of our two corrections, linear counting or Poisson correction, depending on the observed load—that is the proportion of non-empty registers.

[ii] Alternatively, we could use two hash functions: $h_1$ to select the substream and $h_2$ to give the value attached to an element, but it is hard enough to find (and then costly to compute) two hash functions with the sought properties—let alone two which, in addition, will also have to be independent.

## 2 Analysis of the core algorithm

The core algorithm (given in Figure 1, and so called because our final algorithm includes several additional significant corrections), as we have said, relies on stochastic averaging. Because of the way we split the input stream, the size of the resulting substreams is *not deterministic*. This is an issue in calculating the estimator's moments. We are lead to obtain a multi-dimensional integral parameterized by $n$, which we approximate using Laplace's method. Finally, after mean and variance estimates, we conclude this section with a characterization of the limit (large $n$) distribution of the estimator.

The minimum $M$ of $n$ independent random variables uniformly distributed over $[0, 1]$ has a distribution characterized by

$$\mathbb{P}_n[M \in [x, x + \mathrm{d}x]] = n(1 - x)^{n-1}\,\mathrm{d}x. \tag{1}$$

The expectation of $M$ satisfies $\mathbb{E}_n[M] = 1/(n + 1)$, but this observation alone is not enough, because the expectation of the inverse of $M$ diverges.

**Splitting the stream.** Stochastic averaging involves uniformly splitting the input stream into substreams, then proceeding as though each of these substreams is representative of the entire input stream. In our algorithm, this means that if $n$ is the cardinality of the input stream, we expect each of the substreams to have a cardinality close to $n/m$.

From an analytical viewpoint, splitting the $n$ elements of the stream into $m$ substreams, places us in the classical *balls-in-urns model*. Let $N_j$ be the (random) number of elements assigned the $j$-th substream, the probability distribution of the vector $N = (N_1, \ldots, N_m)$ is multinomial

$$\mathbb{P}_n[N_1 = n_1, \ldots, N_m = n_m] = \frac{1}{m^n}\binom{n}{n_1, \ldots, n_m}. \tag{2}$$

Combining (1) and (2) gives us the joint distribution of the $m$ minima $(M_1, \ldots, M_m)$ associated to the $m$ substreams. Let $\mathcal{M}_j$ be the event: $N_j = n_j$ *and* $M_j \in [x_j, x_j + \mathrm{d}x_j]$. Then,

$$\mathbb{P}_n\left[\bigcap_{j=1}^{m} \mathcal{M}_j\right] = \frac{1}{m^n}\binom{n}{n_1, \ldots, n_m}\prod_{j=1}^{m} n_j(1 - x_j)^{n_j - 1}\,\mathrm{d}x_j. \tag{3}$$

In this context, the random variable $\mathcal{Z}$ defined by

$$\mathcal{Z} := \frac{m}{M_1 + \ldots + M_m}$$

is a viable cardinality estimator[(iii)] as we will show in the remainder of this section.

### 2.1 Moments of the inversed mean of minima

**Lemma 1.** *The $r$-th moment of the random variable $\mathcal{Z}$ (inverse mean of minima) is given by*

$$\mathbb{E}_n[\mathcal{Z}^r] = \frac{(n)_m}{n^{m-r}}\int_{[0,\frac{n}{m}]^m}\left(1 - \frac{1}{n}\sum_{j=1}^{m} t_j\right)^{n-m}\frac{\mathrm{d}t_1 \cdots \mathrm{d}t_m}{(t_1 + \ldots + t_m)^r} \tag{4}$$

[(iii)] Or more precisely, we expect $\mathcal{Z}$ to estimate $n/m$ rather than $n$, i.e., $m\mathcal{Z}$ is expected to estimate $n$. In fact as we prove below (Theorem 1), the unbiased estimator is $(m - 1)\mathcal{Z}$.

*using the following notation for falling factorials,* $(n)_m := n(n-1)\cdots(n-m-1)$.

*Proof.* This follows from (3), summed over all possible values of $n_1, \ldots, n_m$ and simplified using the multinomial formula after differentiation,

$$\sum_{n_1,\ldots,n_m} \binom{n}{n_1,\ldots,n_m} n_1 a_1^{n_1-1} \cdots n_m a_m^{n_m-1} = \frac{\partial^m}{\partial a_1 \cdots \partial a_m} \left(a_1 + \ldots + a_m\right)^n$$

$$= \frac{n!}{(n-m)!} \left(a_1 + \ldots + a_m\right)^{n-m}.$$

Next, the normalization $x_j = \frac{m}{n} t_j$ brings the $r$-th moment under its form in (4). □

**Lemma 2.** *Under the regime where $n$ tends to infinity, while $m$ remains fixed,*

$$\mathbb{E}_n[\mathcal{Z}^r] = \frac{n^r}{r \cdot \binom{m-1}{r}} (1 + o(1)). \tag{5}$$

*Proof.* This result is obtained from the exact formula of Lemma 1, by using Laplace's method: we split the integration domain $\left[0, \frac{n}{m}\right]^m$ into two,

$$I_C = \left[0, \frac{\delta(n)}{m}\right]^m \qquad \text{and} \qquad I_T = \left[0, \frac{n}{m}\right]^m \setminus I_C \tag{6}$$

where $I_C$ is the *central domain*, in which the integral turns out to be concentrated and can be approximated, and $I_T$ is the *tail* which can be bounded by a vanishingly small term; the quantity $\delta(n)$ is some function to be chosen later.

*Central approximation:* when integrating on $I_C$, we have $0 \leqslant \sum t_i \leqslant \delta(n)$. The approximation

$$\left(1 - \frac{x}{n}\right)^n = \mathrm{e}^{-x}(1 + o(1)) \tag{7}$$

is only verified for $x = o(\sqrt{n})$; we thus pick $\delta(n) = n^{1/10}$, for instance. We can then use the asymptotic equivalence (7).

With the integral representation of the Gamma function on integers, valid for $y \in \mathbb{R}_{>0}$ and $n \in \mathbb{N}$, we get

$$\int_0^\infty \mathrm{e}^{-ay} a^{r-1} \, \mathrm{d}a = \frac{(r-1)!}{y^r}. \tag{8}$$

as it crucially allows us to separate the integration variables of the denominator. After which, it is a simple matter of grouping exponentials, switching integral and sum, to finally obtain (we have suppressed a factor $a^{r-1}$ for space purposes)

$$\int_{I_C} \left(1 - \frac{1}{n} \sum_{i=1}^m t_i\right)^{n-m} \prod_{i=1}^m \exp\left(-a t_i\right) \mathrm{d}t_i \sim \left(\int_0^{\frac{\delta(n)}{m}} \mathrm{e}^{-(a+1)w} \, \mathrm{d}w\right)^m = \left(\frac{1 - \mathrm{e}^{-(a+1)\frac{\delta(n)}{m}}}{a+1}\right)^m$$

keeping in mind that $n$ is considered to be large (compared to $m$). Because $\mathrm{e}^{-(a+1)\frac{\delta(n)}{m}} \leqslant \mathrm{e}^{-\frac{\delta(n)}{m}}$ is exponentially small, we further have the approximation,

$$a^{r-1}\left(\frac{1-\mathrm{e}^{-(a+1)\frac{\delta(n)}{m}}}{a+1}\right)^m = \frac{a^{r-1}+O\left(\mathrm{e}^{-\frac{\delta(n)}{m}}\right)}{(a+1)^m}. \tag{9}$$

*Tail estimates:* when integrating on $I_T$, at least one of the variables $t_i$ is larger than $\delta(n)/m$,

$$\left(1-\frac{1}{n}\sum_{i=1}^m t_i\right)^n \frac{1}{t_1+\ldots+t_m} \leqslant \left(1-\frac{1}{n}\sum_{i=1}^m t_i\right)^n \leqslant \left(1-\frac{\delta(n)}{nm}\right)^n \sim \mathrm{e}^{-\frac{\delta(n)}{m}}$$

hence an upper bound of the tail that is an exponentially small term,

$$\int_{I_T}\left(1-\frac{1}{n}\sum_{i=1}^m t_i\right)^n \frac{\mathrm{d}t_1\cdots\mathrm{d}t_m}{t_1+\ldots+t_m} \leqslant \frac{n^m}{m^m}\mathrm{e}^{-\frac{\delta(n)}{m}}. \tag{10}$$

A similar exponentially small bound holds with $m$ replaced by $n-m$, as in (4).

*Final result:* by combining equations (9) and (10), and integrating on $a$, we can now give an asymptotic equivalent to the entire integral,

$$\int_{[0,\frac{n}{m}]^m}\left(1-\frac{1}{n}\sum_{j=1}^m t_j\right)^{n-m} \frac{\mathrm{d}t_1\cdots\mathrm{d}t_m}{(t_1+\ldots+t_m)^r} = \int_0^\infty \frac{a^{r-1}\,\mathrm{d}a}{(a+1)^m}(1+o(1)).$$

Hence the equivalence (5) of the statement. □

## 2.2 Asymptotically unbiased estimator

According to Lemma 2, the expected value of the estimator $\mathcal{Z}$ is asymptotically equivalent to $n/(m-1)$, which means we can trivially define a new asymptotically *unbiased* estimator.

**Theorem 1.** *The estimator $\mathcal{Z}^\star$ defined by*

$$\mathcal{Z}^\star := (m-1)\mathcal{Z} = \frac{(m-1)m}{M_1+\ldots+M_m}, \tag{11}$$

*is asymptotically unbiased, in the sense that:*

$$\mathbb{E}_n[\mathcal{Z}^\star] = n(1+o(1)). \tag{12}$$

*Proof.* Follows directly from Lemma 2. □

**Theorem 2.** *The precision of estimator $\mathcal{Z}^\star$, expressed in terms of standard error, satisfies*[iv]

$$\frac{\sigma_n[\mathcal{Z}^\star]}{n} \sim \frac{1}{\sqrt{m-2}}. \tag{13}$$

[iv] We use '$\sim$' to represent asymptotic equivalence; that is, $f(n)\sim g(n)$ as $n\to\infty$, if $\lim\limits_{n\to\infty} f(n)/g(n)=1$.

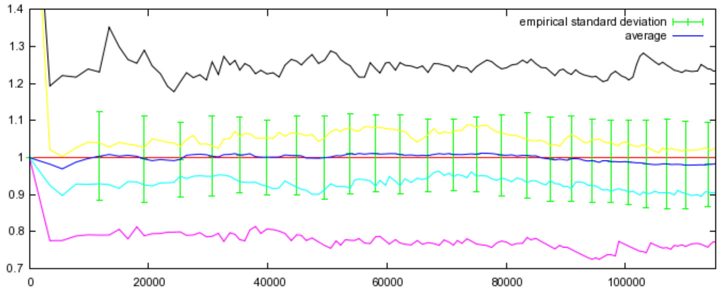

**Fig. 2:** A summary of one hundred executions (each with a different hash function) of the core algorithm on an English document of size $\approx 10^9$ words, containing about $n = 115000$ different words, with $m = 100$. The $x$-axis is labeled with the exact cardinality, the $y$-axis with the normalized estimator ($\mathcal{Z}^\star/n$). The average is very close to $x = 1$, which illustrates Theorem 1. The sample standard deviation appears to be within ±10%, which is in agreement with Theorem 2, as $1/\sqrt{m-2} \approx 0.101$. Additional curves give an idea of the dispersion: 90% of all estimates are within the two extreme curves. [The algorithm used here incorporates corrections developed below.]

*Proof.* From the previous results, we derive

$$\mathbb{V}_n[\mathcal{Z}] = \mathbb{E}_n\left[\mathcal{Z}^2\right] - \mathbb{E}_n[\mathcal{Z}]^2 \sim \frac{n^2}{(m-1)(m-2)} - \frac{n^2}{(m-1)^2} = \frac{n^2}{(m-1)^2(m-2)},$$

thus the result, by the definition of $\mathcal{Z}^\star$ given in equation (11). □

Figure 2 experimentally validates the results of Theorems 1 and 2.

## 2.3 Limit law of the main algorithm

The simplicity of estimator $\mathcal{Z}^\star$ allows us to relatively simply calculate its limit law. Our starting point is the same: we recall that when $M^{(\nu)}$ is the minimum of $\nu$ random variables uniform in $[0,1]$,

$$\mathbb{P}\left[M^{(\nu)} \in [x, x + \mathrm{d}x]\right] = \nu(1-x)^{\nu-1}\,\mathrm{d}x, \tag{14}$$

and the expectation of $M$ is of order $1/n$, so that for large $n$

$$\lim_{n\to\infty} \mathbb{P}\left[\nu M^{(\nu)} \in [t, t + \mathrm{d}t]\right] = \mathrm{e}^{-t}\,\mathrm{d}t,$$

which represents the familiar exponential distribution and results plainly from the usual exponential approximation.

It is well known that the sum of $m$ independent exponentially (with parameter $\alpha = 1$) distributed random variables is gamma-distributed, with density

$$w_m(t) = \mathrm{e}^{-t}\frac{t^{m-1}}{(m-1)!}. \tag{15}$$

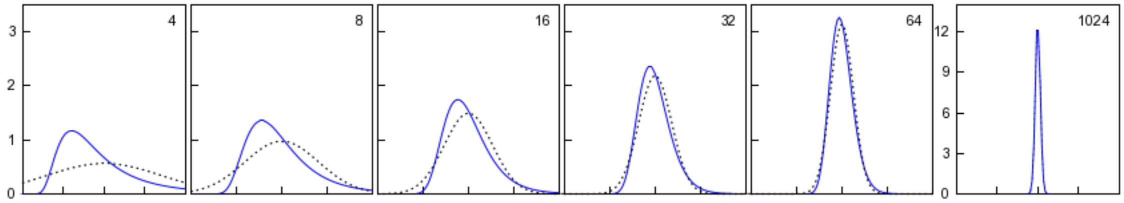

**Fig. 3:** Plots the density of the limit distribution of $\mathcal{Z}^\star/n$ (solid blue) and a normal law matching the mean and standard deviation (dotted black), for $m = 4, 8, 16, 32, 64$ and $1024$.

From here, we call and deal with the sum $S := M_1 + \ldots + M_m$ which involves partial dependency, because the $M_j$ are bound[v]. We can recycle previous computations to get the Laplace transform of $S$ as

$$\begin{aligned}\mathbb{E}\left[\mathrm{e}^{-wS}\right] &= \sum_{n_1,\ldots,n_m} \frac{1}{m^n}\binom{n}{n_1,\ldots,n_m}\int_{[0,1]^m} \mathrm{e}^{-w(x_1+\ldots+x_m)}\prod_{i=1}^{m} n_i(1-x_i)^{n_i-1}\,\mathrm{d}x_i\\ &= \frac{(n)_m}{m^n}\int_{[0,1]^m}\mathrm{e}^{-w(x_1+\ldots+x_m)}\left(m-\sum_{i=1}^{m}x_i\right)^{n-m}\mathrm{d}x_1\cdots\mathrm{d}x_m.\end{aligned}$$

The same approximations we used in Subsection 2.1 apply here as well, to the effect that on $[0,1]^m$,

$$\mathbb{E}\left[\mathrm{e}^{-wS}\right] \sim \left(\int_0^\infty \mathrm{e}^{-t}\mathrm{e}^{-w\frac{m}{n}t}\,\mathrm{d}t\right)^m = \left(\mathbb{E}\left[\mathrm{e}^{-wY\frac{m}{n}}\right]\right)^m \tag{16}$$

where $Y$ is an exponentially distributed random variable with parameter $\alpha = 1$.

Thus by the shape of its Laplace transform, the quantity $S$ asymptotically behaves like the sum of $m$ *independent* random variables, each exponentially distributed with parameter $\lambda = m/n$. The rescaled quantity $S^* := n/m \cdot S$ asymptotically obeys a gamma law with density $w_m(t)$ as given in equation (15).

Now if $S^*$ is gamma distributed according to (15), then $1/S^*$ has a distribution with the induced density

$$\overline{w}_m(u) = \mathrm{e}^{-1/u}\,\frac{u^{-m-1}}{(m-1)!}. \tag{17}$$

Hence:

**Theorem 3.** *For a fixed $m > 1$, as $n$ tends to infinity, the estimator $\mathcal{Z}^\star$ satisfies*

$$\lim_{n\to\infty}\mathbb{P}_n\left[\frac{\mathcal{Z}^\star}{n}\leqslant y\right] = \int_0^{y/(m-1)} \mathrm{e}^{-1/u}\,\frac{u^{-m-1}}{(m-1)!}\,\mathrm{d}u.$$

*Proof.* An immediate consequence of the calculation (17) above and of the continuity theorem for Laplace transforms. □

[v] Recall that $M_j = M^{(N_j)}$. Thus indeed, the $M_j$ are interdependent: each $M_j$ is the minimum of $N_j$ uniform random variables, for some random $N_j$ which satisfies $\sum_j N_j = n$.

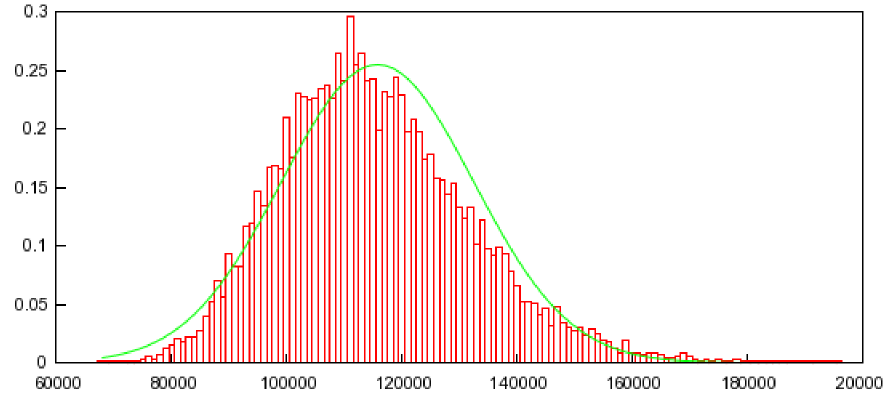

**Fig. 4:** Bar diagram of the distribution of $k = 10000$ estimates of the cardinality of an English dictionary with $n = 115794$ and $m = 50$, to which has been super-imposed the curve of a rescaled normal law of expectated value $\mu = 115794$ and standard deviation $\sigma = 1/\sqrt{48}$. The distribution of the estimates (for this value of $m$) is somewhat skewed with regards to the normal law, in accordance with the theoretical results for the limit distribution.

With this theorem, we can quantify the probabilities of our estimates substantially deviating from the norm.

For example, for values of $m$ of practical use, namely $m \geqslant 32$, the probability of deviating from more than three times the standard deviation is expected to be less than 0.27% and the distribution of the estimates is expected to be barely distinguishable to what the normal law predicts. This phenomenon is illustrated in both Figures 3 and 4.

On the theoretical side, we note that, by restricting the argument of the Laplace transform to imaginary values and making use of Berry-Esseen inequalities, we could bound the speed of convergence to the inverse-gamma law: we expect this speed to be roughly of order $1/\sqrt{n}$. Also simulations reveal the possibility of a stronger convergence in the sense of a *local limit law* [8, §9.9], meaning that the individual probabilities are themselves well approximated by the density of the inverse-gamma law: this could conceivably be established by applying the saddle point method to the inverse Laplace integral transform. These two aspects will be further investigated in my forthcoming PhD thesis.

## 3 Poisson analysis of the intermediate regime

Stochastic averaging, the technique which involves simulating the use of $m$ hash functions by uniformly splitting the input stream into substreams, brings computational efficiency at the cost of a delayed asymptotical regime. This drawback, however, can be compensated for: experimental results have shown that estimates within the non-asymptotical regime are strongly biased, but that they do not appear any more dispersed than estimates within the asymptotical regime; this suggested that the bias can, to some extent, be corrected so as to provide estimates with a similar precision to those of the asymptotical regime.

The analysis of this section validates the existence of this *intermediate regime* and gives us a precise theoretical basis from which to effect the bias correction.

**Poisson model.** As we now focus on pre-asymptotic calculations, $n$ will be assumed to be relatively small compared to $m$. We consider the $m$ substreams to be urns, in which we distribute $n$ values according to a Poisson law (as it is well known that the Poisson distribution is a good approximation of the binomial urn allocation), and we consider the values to be real valued uniform variables of the unit interval.

Formally, let $\lambda := n/m$. We have $m$ urns, and in each urn falls $N_j \in \mathrm{Poi}\,(\lambda)$ values. Each of these $N_j$ values is a uniform random variable on $[0,1]$, called $U_1$ through $U_{N_j}$. We will be interested in the minimal value to have fallen in each urn $j$; if no value has fallen in urn $j$, by convention (and because it makes algorithmical and mathematical sense), we have that the minimum of that urn is 1.

Suppose a given urn $j$ is *not empty*: its minimum $M_j$ belongs to the interval $[x, x+\,\mathrm{d}x]$ with probability

$$\sum_{\nu=1}^{\infty} \mathbb{P}[N_j = \nu]\,\mathbb{P}_\nu[M_j \in [x, x+\,\mathrm{d}x]] = \sum_{\nu=1}^{\infty} \mathrm{e}^{-\lambda}\frac{\lambda^\nu}{\nu!}\nu(1-x)^{\nu-1}\,\mathrm{d}x = \lambda\mathrm{e}^{-\lambda x}\,\mathrm{d}x.$$

In addition, urn $j$ can also be *empty*: this means that $M_j$ has an extra probability $\mathrm{e}^{-\lambda}$ of being equal to 1. Thus we finally get

$$\mathbb{P}[M_j \in [x, x+\,\mathrm{d}x]] = \lambda\mathrm{e}^{-\lambda x}\,\mathrm{d}x + \mathrm{e}^{-\lambda}\mathbf{1}_{\{x=0\}} \tag{18}$$

where $\mathbf{1}_\varepsilon$ is the indicator function of the event $\varepsilon$.

Now let $\mathcal{D}_k$ be the event: *exactly $k$ urns are non-empty, and the rest are empty*. Under this hypothesis, the expectation of our estimator (up to a multiplicative factor) can be expressed as

$$\mathbb{E}_{\mathcal{D}_k}\left[\frac{1}{M_1+\ldots+M_m}\right] = \binom{n}{k}\int_{[0,1]^k}\left(\prod_{j=1}^{k}\lambda\mathrm{e}^{-\lambda x_j}\right)\left(\mathrm{e}^{-\lambda}\right)^{m-k}\frac{\mathrm{d}x_1\cdots\,\mathrm{d}x_k}{x_1+\ldots+x_k+(m-k)}.$$

Using the integral representation (8) as before, we obtain

$$\mathbb{E}_{\mathcal{D}_k}\left[\frac{1}{M_1+\ldots+M_m}\right] = \int_0^\infty \binom{n}{k}\left(\lambda\cdot\frac{1-\mathrm{e}^{-a-\lambda}}{a+\lambda}\right)^k\left(\mathrm{e}^{-a-\lambda}\right)^{m-k}\,\mathrm{d}a. \tag{19}$$

Naturally, equation (19) begs only to be summed over all possible values of $k$,

$$\mathbb{E}\left[\frac{1}{M_1+\ldots+M_m}\right] = \sum_{k=0}^{m}\mathbb{E}_{\mathcal{D}_k}\left[\frac{1}{M_1+\ldots+M_m}\right] = \int_0^\infty f(a)^m\,\mathrm{d}a, \tag{20}$$

with

$$f(a) := \frac{\lambda + a\mathrm{e}^{-a-\lambda}}{a+\lambda}. \tag{21}$$

The resulting integral of (20) can then be approximated using Laplace's method.

**Laplace approximation.** We expand $f(a)^m = \exp\left(m\log(f(a))\right)$ and consider that $a$ is in the neighborhood of 0, as usual to be quantified precisely later, so that we can use the natural logarithm's expansion,

$$\log(f(a)) = -Ca + o(a^2), \qquad C := \frac{\mathrm{e}^{-\lambda}-1}{\lambda} \tag{22}$$

then by combining (20) and (22), we have for a suitable $\delta(m)$

$$\int_0^\infty f(a)^m \,\mathrm{d}a \sim \int_0^{\delta(m)} \mathrm{e}^{-mCa} \,\mathrm{d}a. \tag{23}$$

A further development through Laplace's method yields Theorem 4.

**Theorem 4.** *In the intermediate regime, $n/m = \lambda$ with $\lambda$ bounded away from* 0*, and bounded from above by a positive constant, with $\mathcal{Z}^\star$ the (usual) estimator defined in* (11)

$$\mathbb{E}_n[\mathcal{Z}^\star] \sim \frac{n}{1-\mathrm{e}^{-\lambda}}. \tag{24}$$

We observe, as confirmed by simulations (see Figure 5), that in this regime, the core algorithm overestimates the actual cardinality (e.g.: by about 58% when $n \approx m$) and that this situation strongly degrades as $\lambda$ becomes smaller than 1.

Our core algorithm observes some estimate $\zeta$, which, in the pre-asymptotic regime, is roughly equal to the right hand side of (24). As we want to retrieve a corrected estimate, $n$, we need to solve the following equation in $\lambda$

$$y = \frac{\lambda}{1-\mathrm{e}^{-\lambda}} \tag{25}$$

where $y := \zeta/m$.

From subsequent developments, a couple of things appear noteworthy. First, this correction needs only be applied for $y \in [2.3, 6]$; second, equation (25) can be solved approximatively (and satisfactorily) through several means: a quadratic form either obtained through an asymptotic development of (25), or through empirical determination; the classical iteration process, which converges quite quickly.

## 4 Correcting the non-asymptotic regime of stochastic averaging

The stochastic averaging algorithm only provides sufficiently precise estimates once it has entered its asymptotic regime. We have just seen in Section 3 how to develop an intermediate range correction.

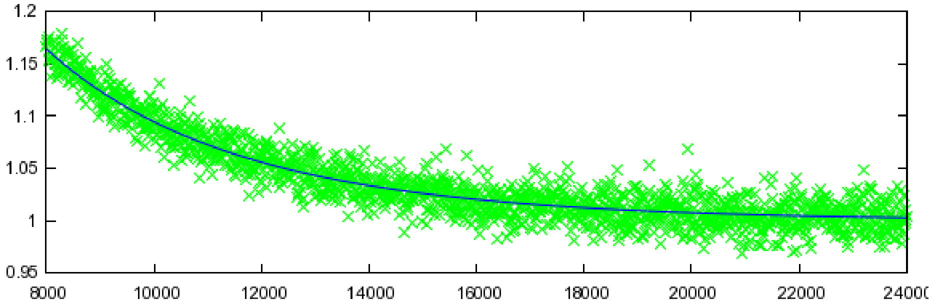

**Fig. 5:** A plot of the accuracy of estimates function of the exact cardinality and, super-imposed, the curve of the function $f(y) := 1/(1-\mathrm{e}^{-\lambda})$. Note the latter follows the center of the distribution perfectly. (Here $m = 4086$.)

Even smaller cardinalities however, which are important in practice[(vi)], must be estimated through other means—one possible solution would be to maintain an exact cardinality count up until a certain threshold.

## 4.1 Small cardinality estimation

It turns out that a subtle shift in viewpoint provides the most elegant and efficient solution. In stochastic averaging, we distribute $n$ values into $m$ sub-intervals, and then average the minimum of each of the $m$ intervals. When $n$ is too small with respect to $m$, this average is distorted because too many sub-intervals are empty. In this case, a sharper estimation of cardinality is obtained from observing *how many* sub-intervals are empty.

This idea was first developed by Whang and al. [12] in their "*linear counting*" algorithm; it has already been used as a palliative measure in Loglog and Hyperloglog. For completeness, we offer in the annex a succinct analysis of the algorithm.

**Proposition 4.1.** *Let $n$ and $m$ tend to infinity in such a way that their ratio $\lambda$ is bounded from above. Let $V$ be the number of empty buckets after hashing $n$ values in $m$ buckets, then*

$$\mathbb{E}[\log V] = \log(m \cdot \exp(-\lambda)) + o(1).$$

This property directly suggests a cardinality estimator.

**Proposition 4.2.** *Let $n$ and $m$ tend to infinity in such a way that their ratio $\lambda$ is bounded from above. Let $\phi$ be the cardinality estimator defined by*

$$\phi(V, m) := -m \log \frac{V}{m}$$

*where $V$ is the empirical number of empty buckets observed during the hashing of $n$ values into $m$ buckets. The estimator $\phi$ is asymptotically unbiased, in the sense that $\mathbb{E}[\phi] \sim n$.*

*Proof.* From proposition 4.1, by linearity of expectation. □

## 4.2 Joining methods together

In this article, we have given three different methods to estimate cardinality, and to each cardinality range (low, mid, and large) corresponds a method which is most efficient. Through our analyses, we have a relatively complete view of the properties of each of these methods, and we now need to assemble them to devise a single algorithm capable of estimating the entire range of possible cardinalities.

This portion of our work is in part suggested by our theoretical work, but also relies on empirical observations—both made from large amounts of actual data (extracted, for instance, from the Gutenberg Project) and from randomly generated data. In the process, we also experimentally validate our results.

Figure 6 is a plot of the error (ratio between the cardinality estimate and the exact cardinality) as a function of the exact cardinality—the threshold for switching from linear counting to the core algorithm has been determined approximatively. Three regimes are apparent:

[(vi)] To put things in perspective: when monitoring a network— that is, estimating the number of active connections—very small and small cardinalities are the rule; however very large cardinalities, while the exception, must also be readily detectable as they are fairly often symptomatic of an attack.

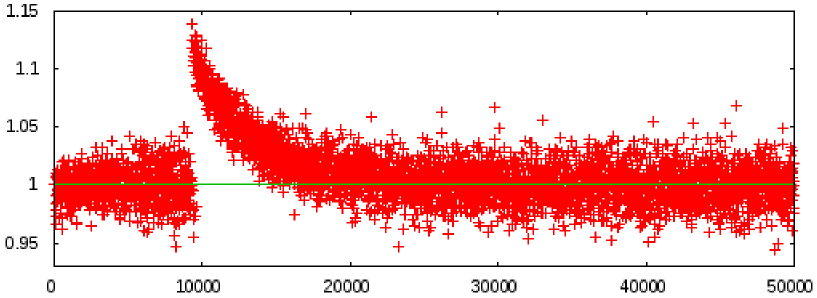

**Fig. 6:** The error of cardinality estimates ranging from 0 to 50000, with $m = 4086$, first using the linear counting algorithm (Subsection 4.1) and then the core algorithm (Section 2).

- first, for $n$ ranging from 0 to about $2.5m$, the linear counting method is used and we notice that the error increases progressively (estimates are dispersed progressively further from 1);
- starting at $n = 2.5m$, our core algorithm is used, and for $n$ ranging from $2.5m$ to about $6m$, we notice a bias—and indeed not a loss of precision, because the estimates are not any more dispersed than further along the plot: they are just shifted with respect to the actual cardinality;
- from $n = 6m$ on, our core algorithm has reached its asymptotical regime and the resulting estimates are satisfyingly precise.

In short, this means that we roughly have two parameters to determine: the threshold at which we switch from linear counting to the core algorithm, and the correction function for the intermediate region.

**Switching threshold.** As an increasing number of the $m$ substreams are assigned at least one value, the accuracy of the estimates provided by linear account decreases. Thus, in choosing when to switch methods, we have to determine when the linear counting method starts to become less accurate than the core algorithm approximately corrected (as suggested by the Poisson analysis).

This was done empirically as a two step process. First, by determining when to switch methods in

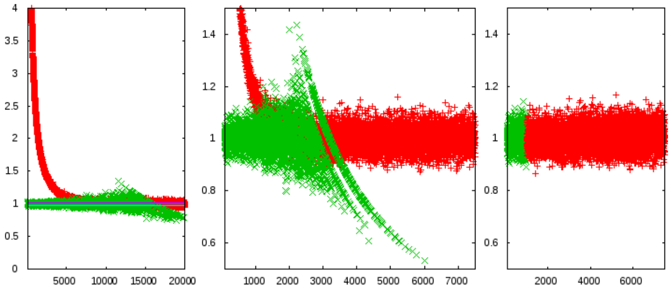

**Fig. 7:** Experimentally determining the threshold at which to switch estimation methods; plot of the relative accuracy ($y$-axis) function of the exact cardinality $n$ ($x$-axis). Note that the value of $m$ varies from the first plot to the other two (for scale purposes).

function of $n$, as is illustrated by Figure 7: the second plot stresses that the linear counting should not be past a certain point as it very much degrades; the third plot is a snapshot of an experiment where the cut-off threshold was moved until the accuracy was most satisfactory. But through this we get a threshold that is a function of the exact value of $n$, whereas this is precisely what the algorithm is attempting to estimate, thus is it not a parameter that is known when choosing which method to use. Consequently, second, we need to "translate" this threshold into a quantity that *is* plainly measurable by the algorithm: in our case, the *load factor* (proportion of substreams that have been assigned at least one element, or in the balls-in-urns view, the proportion of urns that are not empty). Further experimentation reveals that the linear counting is best used up to an 86% load, at which point we are better off switching methods.

**Correcting the core algorithm's pre-asymptotic regime.** As stated previously, our Poisson analysis gives us a precise, *implicit*, equation that characterizes the bias of the pre-asymptotic regime. Two obvious ways of exploiting this equation to reverse the bias are: using a polynomial expansion in $n/m$ or using iteration to get a reasonably good approximation of the equation's solution.

However at this point, we have settled for a prior, simple solution in which we have fit the curve of the bias with a certain function quadratic in $m/n$, which we use for bias inversion (for instance, this was activated in the last plot of Figure 7). This function presents the flaw that it may introduce numerical errors when its parameter (the biased estimate) is too large, so we stop using it once the load has reached $100\%$.

**Extensions.** As we have already mentioned, a *local* limit law seems within reach, and this information would shed even more light on our algorithm. In addition, we would like to find out whether the Poisson correction developped in Section 3 can be pushed to the extent that we no longer need the separate linear counting method of analyzed in Subsection 4.1—while answering this particular question is of little practical interest, it would be intellectually satisfying.

Finally, we are interested in providing a similar limit law result for the Hyperloglog algorithm, as well as reuse and extend our analyses to algorithms which use cardinality estimation as a black box (text classification, bimodal network traffic analysis, etc.).

## References

[1] Noga Alon, Yossi Matias, and Mario Szegedy. The space complexity of approximating the frequency moments. *Journal of Computer and System Sciences*, 58(1):137–147, 1999.

[2] Ziv Bar-Yossef, Ravi Kumar, and D. Sivakumar. Reductions in streaming algorithms, with an application to counting triangles in graphs. In *Proceedings of the thirteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 623–632. Society for Industrial and Applied Mathematics, 2002.

[3] Philippe Chassaing and Lucas Gerin. Efficient estimation of the cardinality of large data sets. In *Proceedings of the 4th Colloquium on Mathematics and Computer Science, Nancy*, volume AG of *Discrete Mathematics & Theoretical Computer Science Proceedings*, pages 419–422, 2006. Full paper available at `http://arxiv.org/abs/math.ST/0701347`.

[4] Marianne Durand and Philippe Flajolet. LOGLOG counting of large cardinalities. In G. Di Battista and U. Zwick, editors, *Annual European Symposium on Algorithms (ESA03)*, volume 2832 of *Lecture Notes in Computer Science*, pages 605–617, 2003.

[5] Philippe Flajolet. Counting by coin tossings. In M. Maher, editor, *Proceedings of ASIAN'04 (Ninth Asian Computing Science Conference)*, volume 3321 of *Lecture Notes in Computer Science*, pages 1–12, 2004. (Text of Opening Keynote Address.).

[6] Philippe Flajolet, Éric Fusy, Olivier Gandouet, and Frédéric Meunier. Hyperloglog: the analysis of a near-optimal cardinality estimation algorithm. In Philippe Jacquet, editor, *Analysis of Algorithms 2007 (AofA07)*, volume AH of *Discrete Mathematics and Theoretical Computer Science Proceedings*, pages 127–146, 2007.

[7] Philippe Flajolet and G. Nigel Martin. Probabilistic counting algorithms for data base applications. *Journal of Computer and System Sciences*, 31(2):182–209, October 1985.

[8] Philippe Flajolet and Robert Sedgewick. *Analytic Combinatorics*. Cambridge University Press, 2009. 824 pages (ISBN-13: 9780521898065); also available electronically from the authors' home pages.

[9] Frédéric Giroire. *Réseaux, algorithmique et analyse combinatoire de grands ensembles*. PhD thesis, Université Paris VI, 2006.

[10] Frédéric Giroire. Order statistics and estimating cardinalities of massive data sets. *Discrete Applied Mathematics*, 157(2):406–427, 2009.

[11] Donald E. Knuth. *The Art of Computer Programming*, volume 3: Sorting and Searching. Addison-Wesley, 2nd edition, 1998.

[12] K-Y. Whang, B.T. Vander-Zanden, and H.M. Taylor. A linear-time probabilistic counting algorithm for database applications. *ACM Transactions on Database Systems*, 15(2):208–229, 1990.

# A Proof of Proposition 4.1

Let $m$ be the number of buckets and $n$, the number of distinct values that are hashed. When $m$ and $n$ both tend to infinity while their ratio $\lambda := n/m$ remains within a closed interval of $\mathbb{R}_{>0}$, the number $V$ of buckets to which no value is hashed is such that

$$\mathbb{E}[V] \sim m \cdot \exp(-\lambda) \qquad \text{and} \qquad \sigma[V] \sim c(\lambda)\sqrt{m} \quad \text{with} \quad c(\lambda) := \sqrt{\exp(-\lambda) - \exp(-2\lambda)}. \tag{26}$$

Chebyshev's inequality implies that, for all real $t > 0$,

$$\mathbb{P}\left[|V - m \cdot \exp(-\lambda)| \geqslant t \cdot c(\lambda)\sqrt{m}\right] \leqslant \frac{1}{t^2}. \tag{27}$$

The starting point of this proof is the formula

$$\mathbb{E}[\log V] = \sum_{i=1}^{\infty} \log i \cdot \mathbb{P}[V = i]\,. \tag{28}$$

Let $t_0$ be a function of $m$ (to be determined *a posteriori*), $r_a$ and $r_b$ be bounds defined by

$$r_a := \left\lfloor m \cdot \exp(-\lambda) - t_0 \cdot c(\lambda)\sqrt{m} \right\rfloor \qquad \text{and} \qquad r_b := \left\lfloor m \cdot \exp(-\lambda) + t_0 \cdot c(\lambda)\sqrt{m} \right\rfloor .$$

We are going to split the sum of (28) according to three intervals, $[1, r_a]$, $[r_a + 1, r_b]$ and $[r_b + 1, \infty[$, then use inequality (27) to show that the two extremal sums, the "*tails*", are negligible.

*Tails:* thus

$$\begin{aligned}\sum_{i=1}^{r_a} \log i \cdot \mathbb{P}[V = i] &\leqslant \log\left(m \cdot \mathrm{e}^{-\lambda}\right) \sum_{i=1}^{r_a} \mathbb{P}[V = i] \\ &\leqslant \log\left(m \cdot \mathrm{e}^{-\lambda}\right) \mathbb{P}[V \leqslant r_a] \\ &\leqslant \log\left(m \cdot \mathrm{e}^{-\lambda}\right) \mathbb{P}\left[t_0 \cdot c(\lambda)\sqrt{m} \leqslant m \cdot \mathrm{e}^{-\lambda} - V\right] \\ &\leqslant \log\left(m \cdot \mathrm{e}^{-\lambda}\right) \frac{1}{{t_0}^2}. \end{aligned} \tag{29}$$

And by a similar method, we also show that

$$\sum_{i=r_b+1}^{\infty} \log i \cdot \mathbb{P}[V = i] = \sum_{i=r_b+1}^{m} \log i \cdot \mathbb{P}[V = i] \leqslant \frac{\log m}{{t_0}^2}. \tag{30}$$

By choosing $t_0$ appropriately (by which we mean $t_0 \gg \sqrt{\log m}$), both sums are bounded by a term that tends to 0 when $m$ tends to infinity—and thus both sums are negligible when $m$ is sufficiently large.

*Central approximation:* The central part requires more stringent bounding. We first determine an upper bound,

$$\sum_{i=r_a+1}^{r_b} \log i \cdot \mathbb{P}[V = i] \leqslant \log r_b \sum_{i=r_a+1}^{r_b} \mathbb{P}[V = i] \leqslant \log r_b \tag{31a}$$

and then a lower bound,

$$\sum_{i=r_a+1}^{r_b} \log i \cdot \mathbb{P}[V = i] \geqslant \log r_a \cdot \left(1 - \frac{1}{{t_0}^2}\right). \tag{31b}$$

Finally,

$$\log\left(m \cdot \mathrm{e}^{-\lambda} \pm t_0 \cdot c(\lambda)\sqrt{m}\right) = \log\left(m \cdot \mathrm{e}^{-\lambda}\left(1 \pm \frac{t_0 \cdot c(\lambda)}{\sqrt{m} \cdot \mathrm{e}^{-\lambda}}\right)\right) = \log\left(m \cdot \mathrm{e}^{-\lambda}\right) + O\left(\frac{t_0}{\sqrt{m}}\right). \tag{31c}$$

Therefore, by choosing $t_0 \ll \sqrt{m}$, we centrally approximate the integrand by the asymptotic equivalent $\log\left(m \cdot \mathrm{e}^{-\lambda}\right)$.

*Finally:* combining equations (29), (30) and (31c), with $t_0 = m^{1/10}$, we obtain

$$\mathbb{E}[\log V] = \log\left(m \cdot \exp\left(-\lambda\right)\right) + o(1), \tag{32}$$

which is the desired result.