A Tour of Understanding Hyperloglog

Zhenghua Lyu
zlyu@vmware.com

Greenplum Developer @ VMware MAPBU

December 5, 2021

Abstract

Hyperloglog [2] is an efficient (both time and space) algorthm to estimate the cardinality (the number of distinct values) of multiset by scanning only one run of data with small memory consumption. This tutorial analyzes the Hyperloglog algorithm in details. It tries best to show every step of the formula deduction to help understand. Some warming up sections are provided for the key math tools used.

Contents

1 Introduction

At the end of May 2020, Github recommends a repo hyperloglog extension for Postgres for me. Then I opened it and got the original paper [2] and found that the author is just the same as the book Analytical Combinatorics [4]: Professor Philippe Flajolet. This bursted my interest to understand the paper. I kept reading and searching for resources of many other papers to help understand. During the whole progress, I learn a lot and decide to write this tutorial in details.

2 Mathematical Tools

2.1 Generating Function

Generating Function is the key tool for analytical combinatorics [4]. Given a set contains many elements, if we can define a size function that maps an element to a natural number, we then get a combinatorial class. Class $\mathcal{A}$ is:

- a set contains many elements $\{a\}$
- with a size function for each element, denote $size(a) = |a| \geq 0$

Then we can write then ordinary generating function $g(z)$ as:

$$g(z) = \sum_{a \in \mathcal{A}} z^{|a|} = \sum_{n \geq 0} A_n z^n \tag{1}$$

Thus A_n is just the number of all elements with size equals n. We encode all the information in just a simple function $g(z)$. There are also other kinds of generating function:

- exponential generating function:$g(z) = \sum_{a \in \mathcal{A}} z^{|a|}/|a|! = \sum_{n \geq 0} A_n z^n/n!$
- poisson generating function: $g(\lambda) = \sum_{n \geq 0} A_n e^{-\lambda} \lambda^n / n!$

Different kinds of generating functions are used in solving different problems. Plenty of examples can be found in [4] and [10].

2.2 Cauchy's Integral Theorem and Residue Theorem

Theorem 1. ***Residue Theorem:***
Let $\mathbb{D}$ be a simply connected domain and let f be analytic in $\mathbb{D}$, except for isolated singularities. Let C be a simple closed curve in $\mathbb{D}$ (oriented counterclockwise, no singularities of f in C), and let $z_1, z_2, \ldots, z_n$ be those isolated singularities of f that lie inside of C. Then

$$\int_C f(z)dz = 2\pi i \sum_{k=1}^{n} Res(f, z_k). \tag{2}$$

2.3 Gamma Function

Gamma Function plays important role in the asymptotic analysis, it has many forms, one is:

$$\Gamma(z) = \int_0^\infty x^{z-1} e^{-x} dx \tag{3}$$

The book [11] has a whole chapter to discuss Gamma Function in details.

Lemma 1.

$$n \in \mathbb{N}^+ \implies \Gamma(n) = (n-1)! \tag{4}$$

Lemma 2.

$$z \in \mathbb{C}, Im(z) \to \infty \implies \Gamma(z) = 0 \tag{5}$$

Lemma 2 is mentioned in [1], ***very fast decrease of*** $\mathbf{\Gamma(s)}$ ***when*** $\mathbf{Im(s)}$ ***tends to infinity***. [1] mentioned that this claim was from [11]. This can be verified by plot the graph in Picutre 1 and Picture 2.

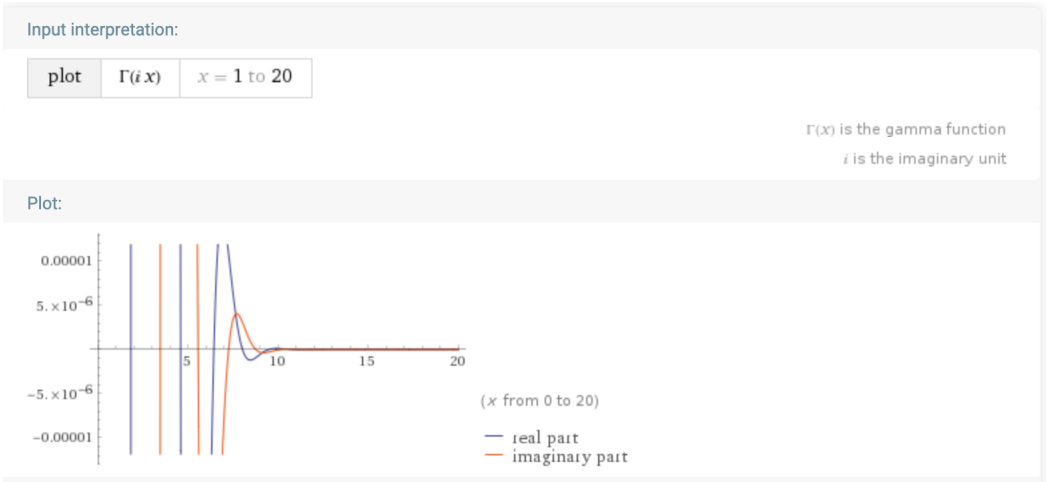

Figure 1: Graph $\Gamma(ix)$ $x \in [1, 20]$

Wolframalpha is a very helpful to verify some basic facts.

2.4 Mellin Transformation

Mellin Transformation is a mapping defined as below:

$$\{\mathcal{M}f\}(s) = f^*(s) = \int_0^\infty f(x) x^{s-1} dx \tag{6}$$

With fundamental strip $< \alpha, \beta >$, then the Mellin Inversion Transformation is:

$$f(x) = \{\mathcal{M}^{-1} f^*\}(s) = \frac{1}{2\pi i} \int_{c-i\infty}^{c+i\infty} x^{-s} f^*(s) ds, \quad \forall c \in \langle \alpha, \beta \rangle \tag{7}$$

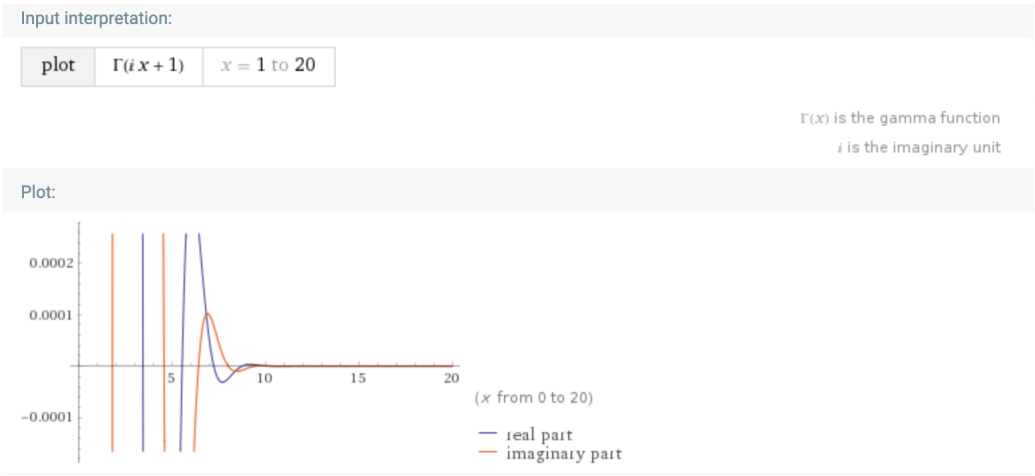

Figure 2: Graph $\Gamma(1+ix)$ $x \in [1, 20]$

It is very useful for asymptotic analysis [3].
We then go through some important properties of Mellin Transformation.

Lemma 3. ***Gamma Function:***

$$\begin{aligned} &f(x) = e^{-x} \implies \\ &\{\mathcal{M}f\}(s) = f^*(s) = \int_0^\infty e^{-x} x^{s-1} dx = \Gamma(s) \end{aligned} \tag{8}$$

Lemma 4. ***Linear Scale:***

$$\begin{aligned} &\{\mathcal{M}f(x)\}(s) = f^*(s) \implies \\ &\{\mathcal{M}f(ax)\}(s) = a^{-s} f^*(s) \end{aligned} \tag{9}$$

Proof.

$$\begin{aligned} \{\mathcal{M}f(ax)\}(s) &= \int_0^\infty f(ax) x^{s-1} dx \\ &= \int_0^\infty f(t) (\frac{t}{a})^{s-1} d(\frac{t}{a}) \\ &= a^{-s} \int_0^\infty f(t) t^{s-1} dt \\ &= a^{-s} f^*(s) \end{aligned} \tag{10}$$

□

Lemma 5. ***Harmonic Sum:***

$$\begin{aligned} &\{\mathcal{M}f(x)\}(s) = f^*(s) \implies \\ &\{\mathcal{M}F(x)\}(s) = \{\mathcal{M} \sum_k \lambda_k f(\mu_k x)\}(s) = (\sum_k \lambda_k \mu_k^{-s}) f^*(s) \end{aligned} \tag{11}$$

Proof.

$$\begin{aligned}\{\mathcal{M}F(x)\}(s) &= \{\mathcal{M}\sum_k \lambda_k f(\mu_k x)\}(s) \\ &= \sum_k \lambda_k \{\mathcal{M}f(\mu_k x)\}(s) \\ &= \sum_k \lambda_k \mu_k^{-s} f^*(s) \\ &= f^*(s)\sum_k \lambda_k \mu_k^{-s}\end{aligned} \tag{12}$$

□

3 Approximate Counting Algorithm

Paper [8] proposed the Approximate Counting Algorithm and paper [1] gave a detailed analysis.

The problem is how to use small memory of registers to count a huge number. The idea is to add 1 in the counter according to some probability that depends on the current state. The algorithm is easy to understand using state machine in Picture 3:

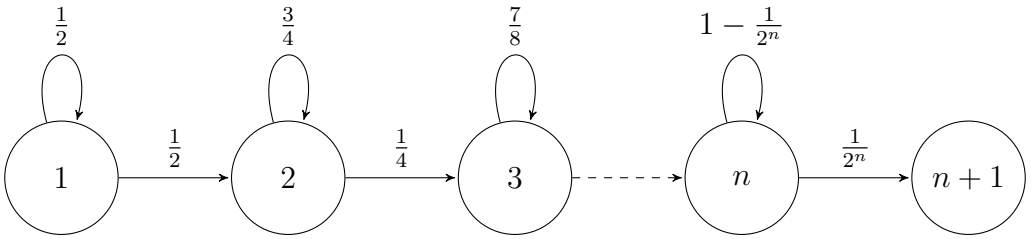

Figure 3: State Machine of Approximate Counting Algorithm

The start state is 1, which means the counter is initialized as 1. And for each event, if its value is n, it either adds 1 by the probability 2^{-n}, or either just keeps the value (ignore the event) by the probability $1-2^{-n}$. The algorithm then uses the value (state) of the register to estimate n. The idea is to estimate $\log(n)$ based on the counter's value and thus finally get n.

Here I only prove the following two lemmas based on the state machine. Given n(the event number), then the counter's value C_n is a random variable in the range $\{1, 2, \ldots, n+1\}$. The distribution probablity is $P_{n,C}$. With the state machine model, we have:

$$\begin{aligned}&P_{0,1} = 1, \\ &P_{1,1} = \frac{1}{2}, P_{1,2} = \frac{1}{2} \\ &P_{n+1,C} = \frac{1}{2}P_{n,C},\ for\ C = 1 \\ &P_{n+1,C} = (1-2^{-C})P_{n,C} + 2^{1-C}P_{n,C-1},\ for\ C \in [2, n+1] \\ &P_{n+1,C} = 2^{1-C}P_{n,C-1},\ for\ C = n+2\end{aligned} \tag{13}$$

Lemma 6. *If n events happen and the counter's value is C_n, then $\mathbb{E}(2^{C_n}) = n + 2$.*

Proof. Prove the lemma by induction.

Base case: $n = 1$, $\mathbb{E}(2^{C_1}) = P_{1,1} \times 2^1 + P_{1,2} \times 2^2 = 3 = n + 2$.

Induction step: Suppose for n, $\mathbb{E}(2^{C_n}) = \sum_{k=1}^{n+1} 2^k P_{n,k} = n + 2$. Now we are trying to prove for $n + 1$: $\mathbb{E}(2^{C_{n+1}})$.

$$
\begin{aligned}
\mathbb{E}(2^{C_{n+1}}) &= \sum_{k=1}^{n+2} 2^k P_{n+1,k} \\
&= 2P_{n+1,1} + 2^{n+2} P_{n+1,n+2} + \sum_{k=2}^{n+1} 2^k P_{n+1,k} \\
&= P_{n,1} + 2P_{n,n+1} + \sum_{k=2}^{n+1} 2^k ((1 - 2^k) P_{n,k} + 2^{1-k} P_{n,k-1}) \qquad (14) \\
&= P_{n,1} + 2P_{n,n+1} + \sum_{k=2}^{n+1} 2^k P_{n,k} - P_{n,k} + 2P_{n,k-1} \\
&= \sum_{k=1}^{n+1} 2^k P_{n,k} - \sum_{k=1}^{n+1} P_{n,k} + 2 \sum_{k=1}^{n+1} P_{n,k} = n + 2 + 1 = (n + 1) + 2
\end{aligned}
$$

□

Lemma 7. *If n events happen and the counter's value is C_n, then $\mathbb{V}(2^{C_n}) = n(n + 1)/2$.*

Skip the lemma's proof because it just uses the same skill as the above proof.

4 Hyperloglog Algorithm

Given an ideal multiset $\mathcal{M}$, *Hyperloglog*($\mathcal{M}$) returns roughly the cardinality of the multiset which can also be known the number of distinct values in $\mathcal{M}$.

The algorithm is so easy that anyone who know some C language could finish it in minutes, the pseudocode is shown below picture 4:

```
for tuple in M:
   # x is the bit array of the hash value
   (x[0], x[1], x[2], ... x[b-1], x[b], ..,)2= hash(tuple)
   bucket_id = (x[0], x[1], x[2], ... x[b-1]) # the first b bit
   pos = the_first_1_bit_pos(x[b], x[b+1], ...)
   reg[bucket_id] = max(reg[bucket_id], pos)
Z = 1/sum(2^(-r) for r in reg)
return fix_coef*Z
```

Figure 4: Pseudocode of Hyperloglog

The estimate is:

$$E = \frac{\alpha_m m^2}{\sum_{j=1}^{m} 2^{-M^{(j)}}} \quad with\ \alpha_m = \left(m \int_0^\infty \left(\log_2 \left(\frac{2+u}{1+u} \right) \right)^m du \right)^{-1} \tag{15}$$

Paper [6] talks about the engineering of Hyperloglog algorithm. Paper [5] compares many different kinds of algorithms to do cardinality estimate. The slide [9] talks about several algorithms on cardinality estimate including hyperloglog and an improved algorithm hyperbitbit (without analysis yet).

4.1 Analysis of Expectation

Based on the above algorithm, claim that the estimate is asymptotically almost unbiased:

$$\frac{1}{n}\mathbb{E}_n(E) \underset{n\to\infty}{=} 1 + \delta_1(n) + o(1) \tag{16}$$

In the formula 16, $|\delta_1(n)| < 5 \times 10^{-5}$ if given $m \geq 16$.
The idea to prove the above claim is:

1. Based on discrete probability expectation formula, find the close form of $\mathbb{E}_n(Z)$
2. For any fixed n, the above $\mathbb{E}_n(Z)$ is just a number, then take $\mathbb{E}_n(Z)$ as an infinite series on n
3. Get the poisson generating function of $\mathbb{E}_n(Z)$, and then try to do asymptotic analysis for the poisson generating function, this step will use some math tools like Mellin transformation
4. Get the asymptotic representation of $\mathbb{E}_n(Z)$, thus finish the job

Now let's go through the details of the above steps.

4.1.1 The close form of $\mathbb{E}_n(Z)$

Let's write the indicator to analyze at the first of this subsubsection:

$$Z = \left(\sum_{j=1}^{m} 2^{-M^{(j)}} \right)^{-1} \tag{17}$$

The indicator in formula 17 is the harmonic mean of m random variables. Random variable $M^{(j)}$ is the value of jth register, which means the max position of the first 1 bit in the hash value of all the data in the jth bucket. We can write Z as $Z(M^{(1)}, \ldots, M^{(m)})$, a function of m random variables. So in order to get the expectation of Z, we have to get the joint distribution of the m random variables: $\Pr(M^{(1)} = k_1, M^{(2)} = k_2, \ldots, M^{(m)} = k_m)$.

To analyze the joint probability, first try to study the situation with just one bucket. For v distinct values to put in one bucket, what is the probability of the max position of the first

1 bit: $\Pr(\max(\rho_1, \rho_2, \ldots, \rho_v))$, where $\{\rho_i\}$ can be taken as i.i.d random variables.

$$\begin{aligned}\Pr(\max(\rho_1, \rho_2, \ldots, \rho_v) \leq k) &= \Pr(\rho_1 \leq k \wedge \rho_2 \leq k \wedge \cdots \wedge \rho_v \leq k) \\ &= \prod_{i=1}^{v} \Pr(\rho_i \leq k) \\ &= \prod_{i=1}^{v} (1 - \Pr(\rho_i > k))\end{aligned} \tag{18}$$

For $\Pr(\rho_i > k)$, the event $(\rho_i > k)$ means the first k bits are all 0, $\Pr(\rho_i > k) = \frac{1}{2^k}$. If we let $F(k) = \Pr(\max(\rho_1, \rho_2, \ldots, \rho_v) \leq k)$, then

$$\begin{aligned}\Pr_v(M^{(j)} = k) &= \Pr(\max(\rho_1, \rho_2, \ldots, \rho_v) = k) \\ &= F(k) - F(k-1) \\ &= (1 - \frac{1}{2^k})^v - (1 - \frac{1}{2^{k-1}})^v\end{aligned} \tag{19}$$

Next we can come to the joint probability:

- Let bucket i gets n_i distinct values, note $\sum_{i=1}^{n} n_i = n$
- Let the value in register i be k_i, note $k_i \geq 1$
- $\Pr_{n_i}(k_i) = (1 - \frac{1}{2^{k_i}})^{n_i} - (1 - \frac{1}{2^{k_i - 1}})^{n_i}$

A specific configuration of n_i's probability is:

$$\begin{aligned}&\frac{1}{m^n}\binom{n}{n_1}\binom{n-n_1}{n_2}\cdots\binom{n-n_1-n_2-\cdots-n_{m-1}}{n_m} \\ =&\frac{1}{m^n}\frac{n!}{n_1!n_2!\ldots n_m!} = \frac{1}{m^n}\binom{n}{n_1 n_2 \ldots n_m}\end{aligned} \tag{20}$$

Joint probability is:

$$\begin{aligned}&\Pr(M^{(1)} = k_1, M^{(2)} = k_2, \ldots, M^{(m)} = k_m) \\ =&\sum_{n_1+n_2+\cdots+n_m=n} \frac{1}{m^n}\binom{n}{n_1 n_2 \ldots n_m} \prod_{i=1}^{m} \left((1 - \frac{1}{2^{k_i}})^{n_i} - (1 - \frac{1}{2^{k_i-1}})^{n_i}\right)\end{aligned} \tag{21}$$

With the probability at hands, we can write down the close form of $\mathbb{E}_n(Z)$:

$$\mathbb{E}_n(Z) = \sum_{k_1,k_2,\ldots,k_m \geq 1} \left(\sum_{j=1}^{m} 2^{-M^{(j)}}\right)^{-1} \sum_{n_1+n_2+\cdots+n_m=n} \binom{n}{n_1 n_2 \ldots n_m} \frac{1}{m^n} \prod_{i=1}^{m} \left((1 - \frac{1}{2^{k_i}})^{n_i} - (1 - \frac{1}{2^{k_i-1}})^{n_i}\right) \tag{22}$$

Finally, we get the close form of $\mathbb{E}_n(Z)$, for each fixed n, it is a deterministic number. The formula 22 is what we are to analyze.

4.1.2 Possion Generating Function

Generating Function is the tool to analyze sequences. Taking the above $\mathbb{E}_n(Z)$ is a sequence on n, we consider using generating function to analyze it. For this pattern, Possion Generating Function [4,7] is more suitable.

$$\mathbb{E}_{\mathcal{P}(\lambda)}(Z) = \sum_{n\geq 0} \mathbb{E}_n(Z) e^{-\lambda} \frac{\lambda^n}{n!} \tag{23}$$

Notice that the $\mathbb{E}_n(Z)$'s formula 22's first part $\sum_{k_1,k_2,\ldots,k_m\geq 1} \left(\sum_{j=1}^{m} 2^{-M^{(j)}}\right)^{-1}$ does not involve n, so the sum over n in 23 can be dragged into the second part of formula 22. Let's focus on this first.

$$\begin{aligned}
&\sum_{n\geq 0} \sum_{n_1+n_2+\cdots+n_m=n} e^{-\lambda} \frac{\lambda^n}{n!} \binom{n}{n_1 n_2 \ldots n_m} \frac{1}{m^n} \prod_{i=1}^{m} \left((1-\frac{1}{2^{k_i}})^{n_i} - (1-\frac{1}{2^{k_i-1}})^{n_i} \right) \\
=&\sum_{n_1,n_2,\ldots,n_m\geq 0} e^{-\lambda} (\frac{\lambda}{m})^n \frac{1}{n_1! n_2! \ldots n_m!} \prod_{i=1}^{m} \left((1-\frac{1}{2^{k_i}})^{n_i} - (1-\frac{1}{2^{k_i-1}})^{n_i} \right) \\
=&\sum_{n_1,n_2,\ldots,n_m\geq 0} e^{-\lambda} (\frac{\lambda}{m})^n \prod_{i=1}^{m} \left(\frac{(1-\frac{1}{2^{k_i}})^{n_i}}{n_i!} - \frac{(1-\frac{1}{2^{k_i-1}})^{n_i}}{n_i!} \right) \\
=&\sum_{n_1,n_2,\ldots,n_m\geq 0} e^{-\lambda} (\frac{\lambda}{m})^{n_1+n_2+\cdots+n_m} \prod_{i=1}^{m} \left(\frac{(1-\frac{1}{2^{k_i}})^{n_i}}{n_i!} - \frac{(1-\frac{1}{2^{k_i-1}})^{n_i}}{n_i!} \right) \\
=&\sum_{n_1,n_2,\ldots,n_m\geq 0} e^{-\lambda} \prod_{i=1}^{m} \left(\frac{(\frac{\lambda}{m})^{n_i}(1-\frac{1}{2^{k_i}})^{n_i}}{n_i!} - \frac{(\frac{\lambda}{m})^{n_i}(1-\frac{1}{2^{k_i-1}})^{n_i}}{n_i!} \right) \\
=&e^{-\lambda} \prod_{i=1}^{m} \sum_{n_i\geq 0} \left(\frac{(\frac{\lambda}{m}(1-\frac{1}{2^{k_i}}))^{n_i}}{n_i!} - \frac{(\frac{\lambda}{m}(1-\frac{1}{2^{k_i-1}}))^{n_i}}{n_i!} \right) \\
=&e^{-\lambda} \prod_{i=1}^{m} (e^{\frac{\lambda}{m}(1-\frac{1}{2^{k_i}})} - e^{\frac{\lambda}{m}(1-\frac{1}{2^{k_i-1}})}) \\
=&e^{-\lambda} \prod_{i=1}^{m} e^{\frac{\lambda}{m}} (e^{-\frac{\lambda}{m2^{k_i}}} - e^{-\frac{\lambda}{m2^{k_i-1}}}) \\
=&\prod_{i=1}^{m} (e^{-\frac{\lambda}{m2^{k_i}}} - e^{-\frac{\lambda}{m2^{k_i-1}}})
\end{aligned} \tag{24}$$

So if we let $g(x) = e^{-x} - e^{-2x}$, then we have

$$\mathbb{E}_{\mathcal{P}(\lambda)}(Z) = \sum_{k_1,k_2,\ldots,k_m\geq 1} \left(\sum_{j=1}^{m} 2^{-k_j} \right)^{-1} \prod_{i=1}^{m} g(\frac{\lambda}{m2^{k_i}}) \tag{25}$$

4.1.3 Asymptotic analysis under the Poisson model

The key skill to make one step further is:$\frac{1}{a} = \int_0^\infty e^{-at}dt$, with this we can turn the first part $\left(\sum_{j=1}^m 2^{-M^{(j)}}\right)^{-1}$ into the the exponential function so that we can factorize it. Also, it opens the door to use Mellin transformation.

Set $\lambda = mx$, we have

$$\begin{aligned}
\mathbb{E}_{\mathcal{P}(mx)}(Z) &= \sum_{k_1,k_2,\ldots,k_m\geq 1}\left(\sum_{j=1}^m 2^{-k_j}\right)^{-1}\prod_{i=1}^m g(\frac{x}{2^{k_i}}) \\
&= \sum_{k_1,k_2,\ldots,k_m\geq 1}\int_0^\infty \prod_{i=1}^m g(\frac{x}{2^{k_i}})e^{-t\sum_{j=1}^m 2^{-k_j}}dt \\
&= \int_0^\infty \sum_{k_1,k_2,\ldots,k_m\geq 1}\prod_{i=1}^m g(\frac{x}{2^{k_i}})e^{-t\sum_{j=1}^m 2^{-k_j}}dt \\
&= \int_0^\infty \sum_{k_1,k_2,\ldots,k_m\geq 1}\prod_{i=1}^m g(\frac{x}{2^{k_i}})e^{-t2^{-k_i}}dt \\
&= \int_0^\infty \prod_{i=1}^m \sum_{k_i\geq 1} g(\frac{x}{2^{k_i}})e^{-t2^{-k_i}}dt \\
&= \int_0^\infty G(x,t)^m dt
\end{aligned} \tag{26}$$

where we have set $G(x,t) = \sum_{k\geq 1} g(\frac{x}{2^k})e^{-\frac{t}{2^k}}$, and

$$\begin{aligned}
&\mathbb{E}_{\mathcal{P}(mx)}(Z) = \int_0^\infty G(x,t)^m dt \implies \\
&\mathbb{E}_{\mathcal{P}(mx)}(Z) = \int_0^\infty G(x,ux)^m d(ux) \implies \\
&H(x) = x\int_0^\infty G(x,ux)^m du \implies \\
&\mathbb{E}_{\mathcal{P}(x)}(Z) = H(\frac{x}{m})
\end{aligned} \tag{27}$$

If you can get the asymptotic result of the above $H(x)$, then we know the poisson generating function's asymptotic result, thus we might use the skill of depoissonization to know the asymptotic result of $\mathbb{E}_n(Z)$.

4.1.4 Mellin Transform to Analyze $G(x, xu)$

If we write $G(x,xu)$ as a function on x with parameter u, we have:

$$h_u(x) = G(x,xu) = \sum_{k=1}^\infty g(2^{-k}x)e^{-2^{-k}xu} = \sum_{k\geq 1} q(x2^{-k}), with\ q(x) = g(x)e^{-xu} \tag{28}$$

Then applying Mellin Transformation, and use lemma 3 and lemma 5, we have

$$\begin{aligned}\{\mathcal{M}h_u\}(s) &= h_u^*(s) \\ &= \left(\sum_{k\geq 1} 2^{ks}\right) q^*(s) \\ &= \frac{2^s\Gamma(s)}{1-2^s}((1+u)^{-s} - (2+u)^{-s})\end{aligned} \tag{29}$$

The corresponding fundamental strip of formula 29 is $\langle -1, 0\rangle$[1].

Next step is to use Mellin Inversion Transformation, Cauthy's Integral Theorem and Residue Theorem to get the bound of the function $h_u(x)$ and then $H(x)$.

Let's take a look at the poles of $h_u^*(s)$:

- Poles from $1-2^s$: $\{\eta_k := 2ik\pi/ln(2), k \in \mathbb{Z}\}$
- Poles from $\Gamma(s)$: Gamma function has simple poles at non-positive integers

The Mellin Inversion Transformation of formula 29 is:

$$h_u(x) = \frac{1}{2\pi i}\int_{-1/2-i\infty}^{-1/2+i\infty} h_u^*(s)x^{-s}ds \tag{30}$$

Consider the following integral when given some T through the path shown in Picture5:

$$\underbrace{\frac{1}{2\pi i}\int_{-1/2-Ti}^{-1/2+Ti} h_u^*(s)x^{-s}ds}_{\blacktriangle} + \underbrace{\frac{1}{2\pi i}\int_{-1/2+Ti}^{1+Ti} h_u^*(s)x^{-s}ds}_{\blacksquare} + \underbrace{\frac{1}{2\pi i}\int_{1+Ti}^{1-Ti} h_u^*(s)x^{-s}ds}_{\blacklozenge} + \underbrace{\frac{1}{2\pi i}\int_{1-Ti}^{-1/2-Ti} h_u^*(s)x^{-s}ds}_{\bigstar} \tag{31}$$

When $T \to \infty$, in the integral of 31:

- $\blacktriangle = h_u(x)$
- $\blacksquare$ and $\bigstar$ both are 0 using the lemma 2[2]

With residue theorem, we have

$$\begin{aligned}&\blacktriangle + \blacksquare + \blacklozenge + \bigstar \underset{T\to\infty}{=} -\sum_{k\in\mathbb{Z}} Res(h_u^*(s)x^{-s}, \eta_k) \implies \\ &\blacktriangle + \blacklozenge \underset{T\to\infty}{=} -\sum_{k\in\mathbb{Z}} Res(h_u^*(s)x^{-s}, \eta_k) \implies \\ &h_u(x) = \underbrace{-\sum_{k\in\mathbb{Z}} Res(h_u^*(s)x^{-s}, \eta_k)}_{Part1} + \underbrace{\frac{1}{2\pi i}\int_{1-i\infty}^{1+i\infty} h_u^*(s)x^{-s}ds}_{Part2}\end{aligned} \tag{32}$$

[1] I do not understand how to get the fundamental strip yet. Just skip this by accepting its correctness.

[2] This is another point that I do not quite understand.

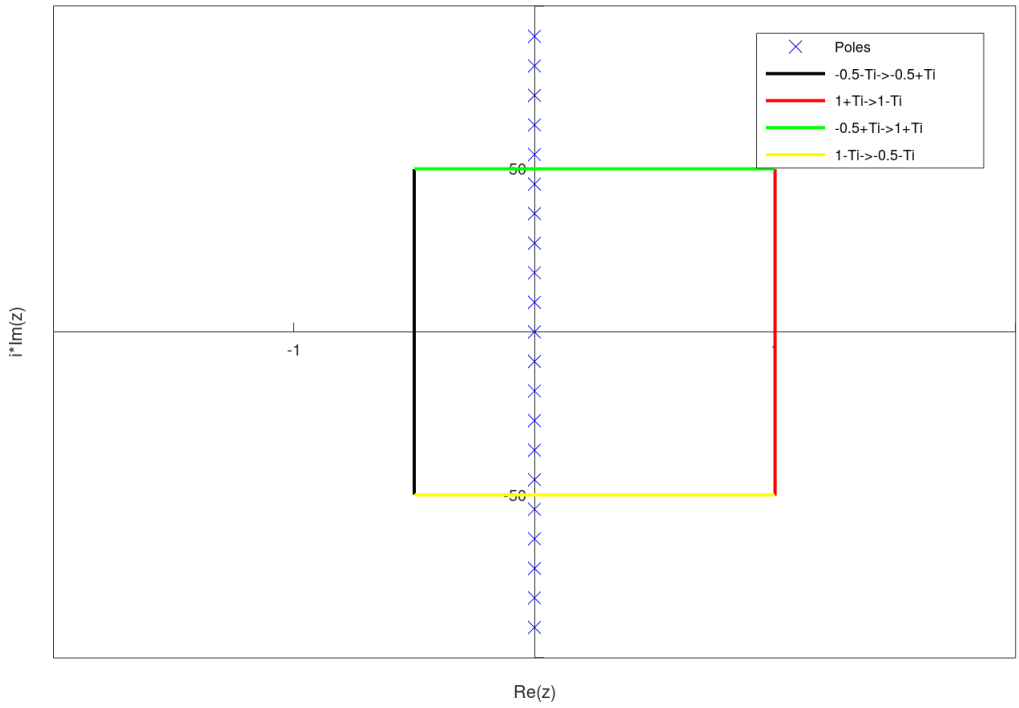

Figure 5: Integral path

Now let's bound the above *part*1 and *part*2 in equation 32.

Note that $h_u^*(s)x^{-s} = \frac{2^s\Gamma(s)}{1-2^s}((1+u)^{-s} - (2+u)^{-s})x^{-s}$, now take a look at $((1+u)^{-s} - (2+u)^{-s})$, first use inequality to bound this, the skill is to consider it as a result of definite integration:

$$\begin{aligned}
|(1+u)^{-s} - (2+u)^{-s}| &= |e^{-s\log(1+u)} - e^{-s\log(2+u)}| \\
&= |\int_{1+u}^{2+u} (e^{-s\log(x)})' dx| \\
&= |\int_{1+u}^{2+u} e^{-s\log(x)} \frac{-s}{x} dx| \\
&\leq \int_{1+u}^{2+u} |e^{-s\log(x)}/x||s| dx \\
&= |s| \int_{1+u}^{2+u} |\frac{e^{-s\log(x)}}{x}| dx, \ \textit{with } u > 0, \ \textit{and } \mathcal{R}e(s) \geq 0 \\
&\leq |s| \int_{1+u}^{2+u} \frac{1}{x} dx = |s| \log\left(\frac{2+u}{1+u}\right), \ \textit{with } u > 0, \ \textit{and } \mathcal{R}e(s) \geq 0
\end{aligned} \tag{33}$$

For *part*2, it is the integral on the line $\mathcal{R}e(s) = 1$, directly apply the above inequality 33, we get:

$$\begin{aligned}\left|part2\right| &= \left|\frac{1}{2\pi i}\int_{1-i\infty}^{1+i\infty}\frac{2^s\Gamma(s)}{1-2^s}((1+u)^{-s}-(2+u)^{-s})x^{-s}ds\right| \\ &\leq \frac{1}{2\pi}\log\left(\frac{2+u}{1+u}\right)\int_{1-i\infty}^{1+i\infty}\left|\frac{2^s}{1-2^s}\right|\left|s\Gamma(s)\right|\left|x^{-s}\right|ds,\ with\ \mathcal{R}e(s)=1\end{aligned} \tag{34}$$

For $\left|\frac{2^s}{1-2^s}\right|$, we can plot its graph as below picture 6, and from the graph[3]. we know $\left|\frac{2^s}{1-2^s}\right| \leq 2$.

```
In [1]: import numpy as np
        from matplotlib.pyplot import plot

        sigma = np.arange(-100, 100, 0.01)
        s = 1 + complex(0, 1)*sigma
        y = abs((2**s)/(1-2**s))
        plot(sigma, y)

Out[1]: [<matplotlib.lines.Line2D at 0x109f910d0>]
```

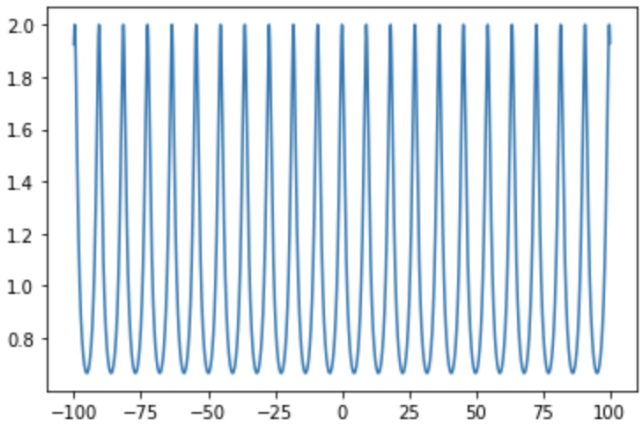

Figure 6: Graph of $\left|\frac{2^s}{1-2^s}\right|$

For x^{-s}, it is easy to show $x^{-s} = x^{-1-\sigma i} \implies \left|x^{-s}\right| = \frac{1}{x}\left|e^{i(-\sigma\log(x))}\right| \leq \frac{1}{x}$.

For the integral involving Gamma Function, I use wolframalpha to compute it, the result is in picture7 and we have $\int_{1-i\infty}^{1+i\infty}|s\Gamma(s)|ds \approx 3.51445$.

Combine all the above, we have $part2 = \mathcal{O}(\frac{1}{x})\log|\frac{2+u}{1+u}|$. This tells us $part2$ decreases fast with x getting large[4].

We now come to $part1$, first let's compute the residues of $h_u^*(s)x^{-s}$ in the poles in the imaginary axis. Because $\Gamma(0) = \infty$, for residue at 0 we have to separately consider it:

[3]It can also be deduced using algebra method, here I just skip it.

[4]Some consts are not the same as in the original paper [2], but the result is the same under Big O notation.

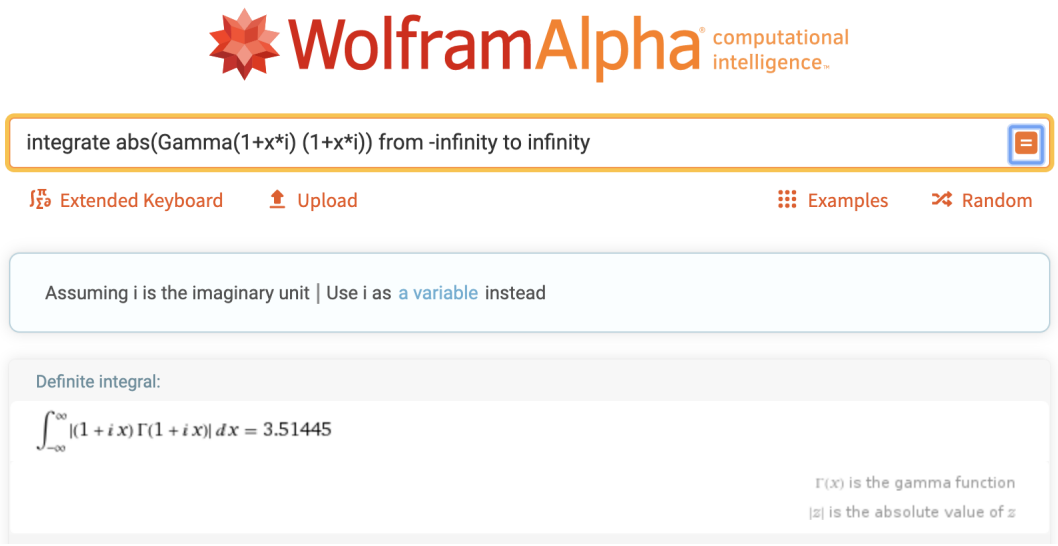

Figure 7: Result of integral $\int_{1-i\infty}^{1+i\infty} |s\Gamma(s)|ds$

$$
\begin{aligned}
Res(h_u^*(s)x^{-s}, 0) &= -\frac{1}{\log 2}\log\left(\frac{2+u}{1+u}\right) \\
Res(h_u^*(s)x^{-s}, \eta_k) &= -\frac{1}{\log 2}x^{-\eta_k}\Gamma(\eta_k)\left((1+u)^{-\eta_k} - (2+u)^{-\eta_k}\right),\ with\ k \in \mathbb{Z}_{\neq 0}
\end{aligned}
\tag{35}
$$

The residue at 0 is computed by Wolframalpha[5], see the picture 8.

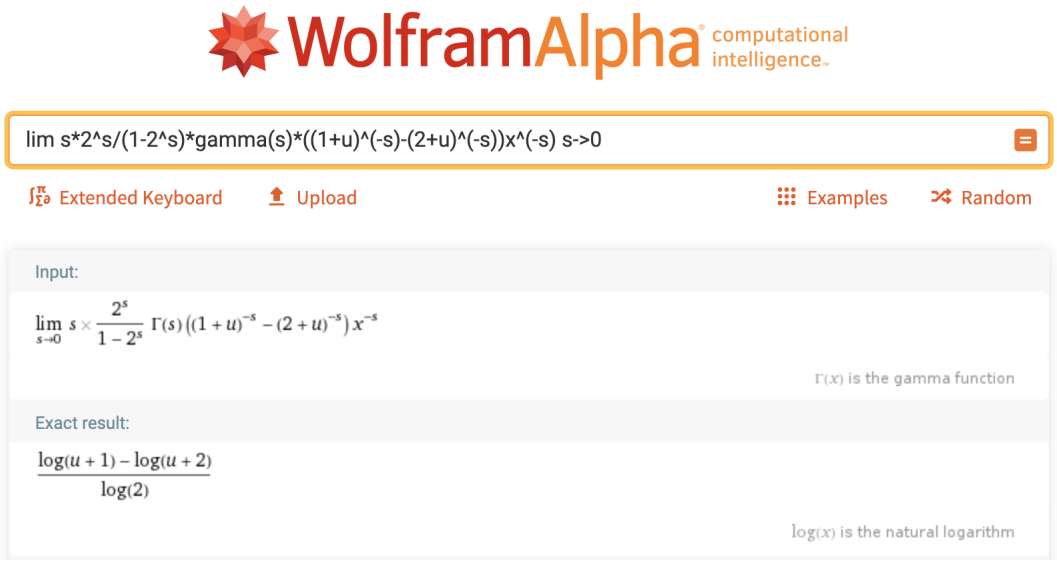

Figure 8: $Res(h_u^*(s)x^{-s}, 0)$

[5]How to handle $\Gamma(0)$ in limit computing is beyond my skill.

For residues not at 0, because $Gamma(s)$ is anlytical, so I use L'Hospital's rule[6] to get the paper's result[7]:

$$\begin{aligned}
Res(h_u^*(s)x^{-s},\eta_k) &= x^{-\eta_k}\Gamma(\eta_k)\left((1+u)^{-\eta_k}-(2+u)^{-\eta_k}\right)\lim_{s\to\eta_k}\frac{(s-\eta_k)2^s}{1-2^s}\\
&= x^{-\eta_k}\Gamma(\eta_k)\left((1+u)^{-\eta_k}-(2+u)^{-\eta_k}\right)\lim_{s\to\eta_k}\frac{s-\eta_k}{2^{-s}-1}\\
&= x^{-\eta_k}\Gamma(\eta_k)\left((1+u)^{-\eta_k}-(2+u)^{-\eta_k}\right)\lim_{s\to\eta_k}\frac{1}{-\log 2e^{-s\log 2}}\\
&= -\frac{1}{\log 2}x^{-\eta_k}\Gamma(\eta_k)\left((1+u)^{-\eta_k}-(2+u)^{-\eta_k}\right),\ with\ k\in\mathbb{Z}_{\neq 0}\\
&= -\frac{1}{\log 2}e^{-2\pi ki\log x}\Gamma(\eta_k)\left((1+u)^{-\eta_k}-(2+u)^{-\eta_k}\right),\ with\ k\in\mathbb{Z}_{\neq 0}
\end{aligned} \tag{36}$$

Use the inequality 33, we can get:

$$\left|Res(h_u^*(s)x^{-s},\eta_k)\right| \le 2\pi\log_2\frac{2+u}{1+u}\left|k\Gamma(\eta_k)\right| \tag{37}$$

Some typical value for $Y_k = \left|k\Gamma(\eta_k)\right|$ is $Y_1 = 0.0000517, Y_2 = 0.00000000378, Y_3 = 2.397...\times 10^{-13}, \ldots$. It decreases very fast as k getting large. Thus we may conjecture that the $part1$ is only determined by the residue at 0. I still use wolframalpha to verify this[8]: $\sum_{k\in\mathbb{Z}_{\neq 0}}\left|Res(h_u^*(s)x^{-s},\eta_k)\right| \approx 0.00010345394$ in Picture9.

With bound of $part1$ and $part2$, we now come to the asymptotics of $h_u(x)$ for some fixed $u>0$ when $x\to+\infty$:

$$G(x,xu) = \log_2\left(\frac{2+u}{1+u}\right)(1+\mathcal{O}(x^{-1})+\epsilon),\ with\ |\epsilon|\le 7\times 10^{-4} \tag{38}$$

4.1.5 Final asymptotics of the Poisson averages

After knowing $G(x,xu)$, we come to $H(x) = x\int_0^\infty G(x,ux)^m du$. Let $f(u) = \log_2\left(\frac{2+u}{1+u}\right)$ then we have:

$$\begin{aligned}
\frac{1}{x}H(x) &\underset{x\to\infty}{=} \int_0^\infty f(u)^m(1+\epsilon+o(1))^m du\\
&= \int_0^\infty f(u)^m du + \epsilon_m + o(1),\ with\ very\ small\ \epsilon_m
\end{aligned} \tag{39}$$

The integeral $U_m = \int_0^\infty f(u)^m du$ converges for $m = 2^b$. I use wolframalpha to get some values of it: $U_2 = 1.4237, U_4 = 0.469, U_8 = 0.1998$.

[6] Although I get the result, but I do not know the exact preassumption for complex limit using L'Hospital's rule.

[7] With a different sign, I believe it is a typo.

[8] In fact, I do not follow the bound formula in the original paper [2].

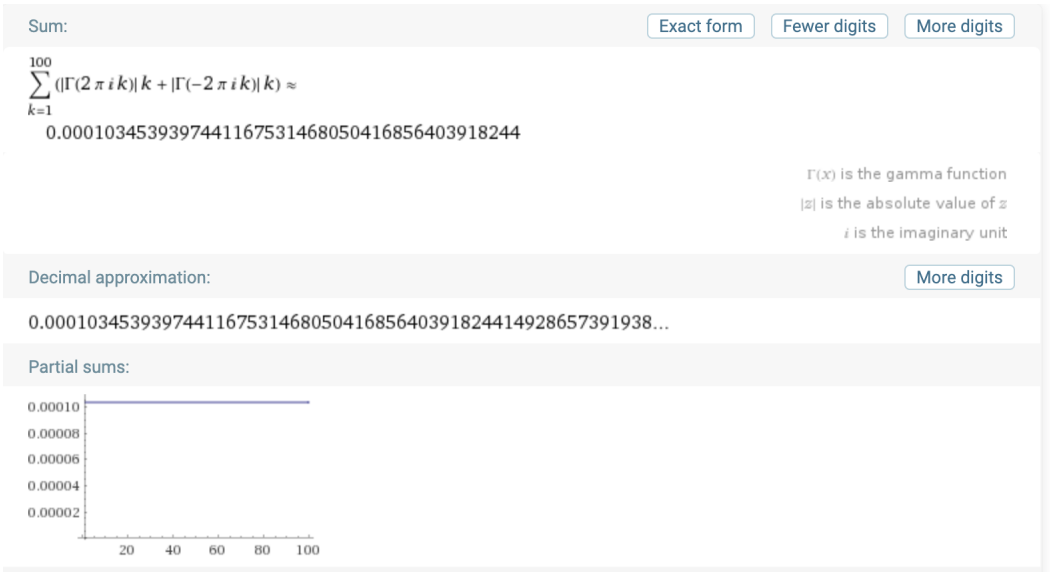

Figure 9: $\sum_{k\in\mathbb{Z}_{\neq 0}} \left| Res(h_u^*(s)x^{-s}, \eta_k) \right|$

Then we get the poisson generating function's asymptotics:

$$\begin{aligned} \mathbb{E}_{\mathcal{P}(\lambda)} &= H(\frac{\lambda}{m}) \\ &= \frac{\lambda}{m}\left(\int_0^\infty f(u)^m du + \epsilon_m + o(1)\right) \\ &= \frac{\lambda}{m}\left(\frac{1}{m\alpha_m} + \epsilon_m + o(1)\right) \end{aligned} \tag{40}$$

Where in the above equation40, $\alpha_m = \left(m\int_0^\infty \left(\log_2\left(\frac{2+u}{1+u}\right)\right)^m du\right)^{-1}$.

The final step is to apply the ***Analytical depoissonization*** theorem. This theorem says, when the poisson generating function satisfies some condition, we can have the asymptotic result: $\mathbb{E}_n = \mathbb{E}_{\mathcal{P}(n)}(Z) + \mathcal{O}(1)$. **I am not going to verify the condition is suitable for the hyperloglog case, for that details please read the paper [2]'s section2.3.** Paper [7] is the original paper on this.

By skipping the proof of the condition satisfication, we get:

$$\mathbb{E}_n(Z) = \mathbb{E}_{\mathcal{P}(n)}(Z) + \mathcal{O}(1) \approx \frac{n}{m^2\alpha_m} \tag{41}$$

Thus we deisgn the hyperloglog estimator as $E = \alpha_m m^2 Z$.

4.2 Analysis of Variance

This part is almost the same progress as the mean analysis. So I do not dive into the details her but only copy the result from the paper [2], and use wolfram alpha to compute the constant.

$$\begin{aligned}\mathbb{V}_n(Z) &\approx \mathbb{V}_{\mathcal{P}(n)}(Z) \\ &\approx \frac{n^2}{m^2}\left(\int_0^\infty uf(u)^m du - \left(\int_0^\infty f(u)^m du\right)^2\right)\end{aligned} \tag{42}$$

Let the special integral to be $J_k(m) = \int_0^\infty u^k f(u)^m du$, the equation42 can be written as $\mathbb{V}_n(Z) \approx \frac{n^2}{m^2}\left(J_1(m) - J_0(m)\right)^2 = \frac{n^2}{m^2} J_0(m)^2 \left(\frac{J_1(m)}{J_0(m)^2} - 1\right)$. Also $\alpha_m = \frac{1}{mJ_0(m)}$. With all these re-written and $E = \alpha_m m^2 Z$, we have[9]:

$$\begin{aligned}&\mathbb{V}_n(E) = \mathbb{V}_n(\alpha_m m^2 Z) = \alpha_m^2 m^4 \mathbb{V}_n(Z) \approx \alpha_m^2 m^2 n^2 J_0(m)^2 \left(\frac{J_1(m)}{J_0(m)^2} - 1\right) \implies \\ &\frac{1}{n}\sqrt{\mathbb{V}_n(E)} \approx \alpha_m m J_0(m)\sqrt{\frac{J_1(m)}{J_0(m)^2} - 1} = \sqrt{\frac{J_1(m)}{J_0(m)^2} - 1} = \frac{\sqrt{m}\sqrt{\frac{J_1(m)}{J_0(m)^2} - 1}}{\sqrt{m}}\end{aligned} \tag{43}$$

Thus with $\beta_m = \sqrt{m}\sqrt{\frac{J_1(m)}{J_0(m)^2} - 1}$, we have finished the proof of relative accuracy of the hyperloglog in the paper [2]:

$$\frac{1}{n}\sqrt{\mathbb{V}_n(E)} = \frac{\beta_m}{\sqrt{m}} + \delta_2(n) + o(1) \tag{44}$$

Paper [2] uses Laplace method to find the asymptotics of $J_k(m)$. I skip that details and only give some value of β_m using wolframalpha here: $\beta_{16} = 1.106, \beta_{32} = 1.071, \beta_{64} = 1.0545, \beta_{256} = 1.0429$.

4.3 Analysis of Space

Only a rough idea is provided here to try to understand space performance: ***We estimate the log of cardinality based on the register's value, so the memory should be loglog level.***

References

[1] P. Flajolet. Approximate counting: a detailed analysis. *BIT Numerical Mathematics*, 25(1):113–134, 1985.

[2] P. Flajolet, É. Fusy, O. Gandouet, and F. Meunier. Hyperloglog: the analysis of a near-optimal cardinality estimation algorithm. 2007.

[3] P. Flajolet, X. Gourdon, and P. Dumas. Mellin transforms and asymptotics: Harmonic sums. *Theoretical computer science*, 144(1):3–58, 1995.

[9] The original paper [2] missed a denominator of m^2 in its equation 26. I think it is a typo.

[4] P. Flajolet and R. Sedgewick. *Analytic combinatorics*. cambridge University press, 2009.

[5] H. Harmouch and F. Naumann. Cardinality estimation: An experimental survey. *Proceedings of the VLDB Endowment*, 11(4):499–512, 2017.

[6] S. Heule, M. Nunkesser, and A. Hall. Hyperloglog in practice: algorithmic engineering of a state of the art cardinality estimation algorithm. In *Proceedings of the 16th International Conference on Extending Database Technology*, pages 683–692, 2013.

[7] P. Jacquet and W. Szpankowski. Analytical depoissonization and its applications. *Theoretical Computer Science*, 201(1-2):1–62, 1998.

[8] R. Morris. Counting large numbers of events in small registers. *Communications of the ACM*, 21(10):840–842, 1978.

[9] R. Sedgewick. Cardinality estimation, Jan 2018.

[10] R. Sedgewick and P. Flajolet. *An introduction to the analysis of algorithms*. Pearson Education India, 2013.

[11] E. T. Whittaker and G. N. Watson. *A course of modern analysis*. Dover Publications, 2020.