

# Implementation Notes for entropy estimation based on NIST SP 800-90B non-IID track

Gen'ya SAKURAI

September 21, 2025

## 1

 Implementation notes for numerical computation of entropy estimates

### 1.1 Notes for the Markov estimate

In this section, we follow the convention using a column vector of probabilities  $\mathbf{P}$  and a left stochastic matrix  $\mathbf{T}$ .

$$\mathbf{P} \equiv \begin{bmatrix} p_0 \\ p_1 \end{bmatrix} \quad (1)$$

$$\mathbf{T} \equiv \begin{bmatrix} t_{0,0} & t_{0,1} \\ t_{1,0} & t_{1,1} \end{bmatrix} \quad (2)$$

Note here that 6.3.3 of NIST SP 800-90B[1] uses the convention of a row vector of probabilities and a right stochastic matrix. So the transition matrix components  $t_{i,j}$  in this section will be interpreted as  $P_{j,i}$  in 6.3.3 of NIST SP 800-90B[1].

Considering a sequence of binary-valued samples of length  $\lambda$ , there are following four possible combinations of the first sample value and the last sample value:

- a)  $0 \rightarrow 0$
- b)  $0 \rightarrow 1$
- c)  $1 \rightarrow 0$
- d)  $1 \rightarrow 1$

For each combination, the probability of occurrence of the most likely sequence is expressed by the following equation, by using the parameters  $\mu$  and  $\nu$ :

a)

$$f_a(\mathbf{T}, \mu, \nu) \equiv p_0 t_{00}^\nu t_{11}^{\lambda-1-2\mu-\nu} t_{01}^\mu t_{10}^\mu \quad (3)$$

$$\begin{cases} 0 \leq 2\mu < \lambda - 1 \\ 0 \leq \nu \leq \lambda - 1 \\ 0 \leq \lambda - 1 - 2\mu - \nu \leq \lambda - 1 \end{cases} \quad (4)$$

b)

$$f_b(\mathbf{T}, \mu, \nu) \equiv p_0 t_{00}^\nu t_{11}^{\lambda-2-2\mu-\nu} t_{01}^\mu t_{10}^{\mu+1} \quad (5)$$

$$\begin{cases} 0 \leq 2\mu < \lambda - 1 \\ 0 \leq \nu < \lambda - 1 \\ 0 \leq \lambda - 2 - 2\mu - \nu \leq \lambda - 2 \end{cases} \quad (6)$$

c)

$$f_c(\mathbf{T}, \mu, \nu) \equiv p_1 t_{00}^\nu t_{11}^{\lambda-2-2\mu-\nu} t_{01}^{\mu+1} t_{10}^\mu \quad (7)$$

$$\begin{cases} 0 \leq 2\mu < \lambda - 1 \\ 0 \leq \nu < \lambda - 1 \\ 0 \leq \lambda - 2 - 2\mu - \nu \leq \lambda - 2 \end{cases} \quad (8)$$

d)

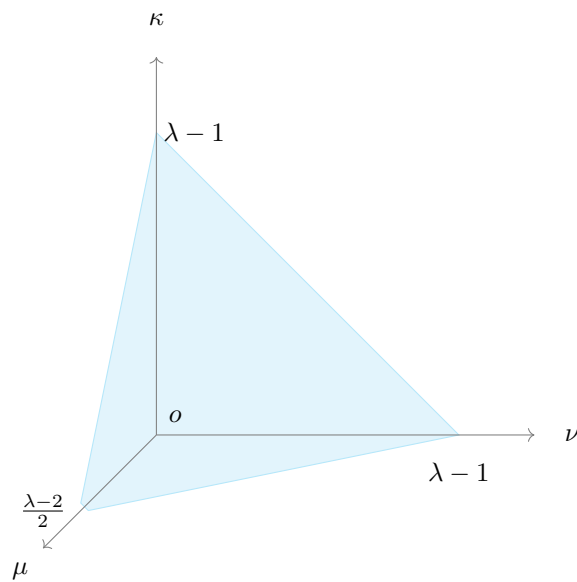
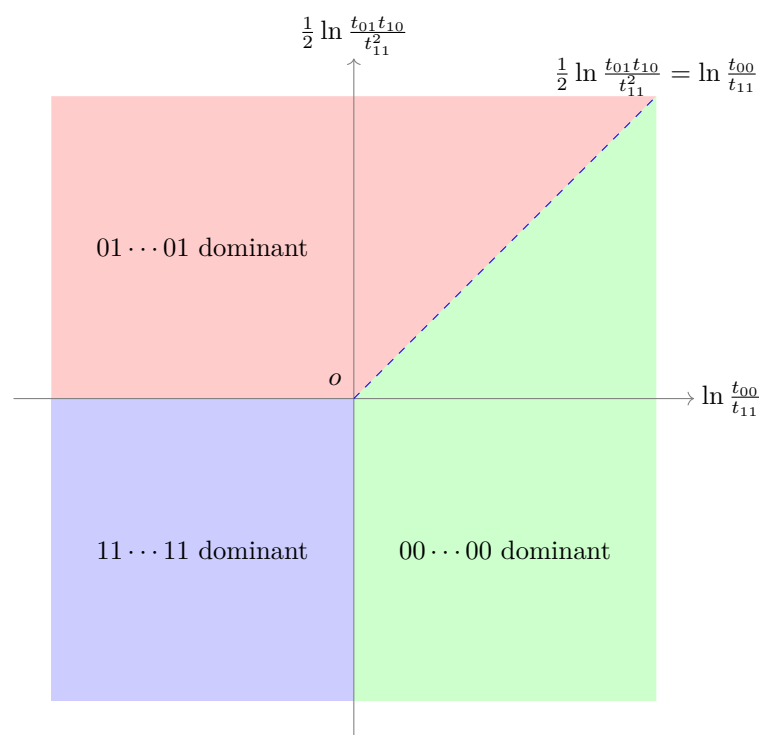
$$f_d(\mathbf{T}, \mu, \nu) \equiv p_1 t_{00}^\nu t_{11}^{\lambda-1-2\mu-\nu} t_{01}^\mu t_{10}^\mu \quad (9)$$

$$\begin{cases} 0 \leq 2\mu < \lambda - 1 \\ 0 \leq \nu \leq \lambda - 1 \\ 0 \leq \lambda - 1 - 2\mu - \nu \leq \lambda - 1 \end{cases} \quad (10)$$

If we denote  $\kappa$  as the exponent of  $t_{11}$ , then the possible values of  $\mu, \nu$ , and  $\kappa$  are expressed as integer coordinates in the approximately triangular region (to be precise, truncated triangle) shown in Figure 1.

In order to calculate the extrema of  $f_a, f_b, f_c$ , and  $f_d$ , assuming that  $\mu$ , and  $\nu$  are continuous variables, we first consider the partial derivatives of  $f_a(\mathbf{T}, \mu, \nu)$  with respect to  $\mu$ , and  $\nu$ .

$$\begin{aligned} \frac{\partial}{\partial \mu} f_a(\mathbf{T}, \mu, \nu) &= (-2 \ln t_{11} + \ln t_{01} + \ln t_{10}) f_a(\mathbf{T}, \mu, \nu) \\ &= \ln \frac{t_{01} t_{10}}{t_{11}^2} f_a(\mathbf{T}, \mu, \nu) \end{aligned} \quad (11)$$

Figure 1 Parameter plane of the exponents  $\kappa, \mu, \nu$ Figure 2 Phase diagram of most likely sequence based on  $f_a(\mathbf{T}, \mu, \nu)$  with respect to components of stochastic matrix  $\mathbf{T}$ 

Taking into account eqs. (11) and (12), and the property that the value  $f_a(\mathbf{T}, \mu, \nu)$  is non-negative, a rough phase diagram will be obtained as shown in Figure 2, by taking the horizontal axis as  $\ln \frac{t_{00}}{t_{11}}$ , and the vertical axis as  $\frac{1}{2} \ln \frac{t_{01} t_{10}}{t_{11}^2}$ .

First, if we consider the third quadrant, for example, the conditional probability  $t_{11}$  is greater than or equal to  $t_{00}$  and  $\sqrt{t_{01} t_{10}}$ , then the sequence  $11 \dots 11$  will be dominant.

Next, for the other quadrants, let us consider whether  $00 \dots 00$  or  $01 \dots 01$  is the dominant sequence. In particular, for the first quadrant, since right hand sides of eqs. (11) and (12) are positive, let us examine whether  $00 \dots 00$  or  $0101 \dots 0101$  is the dominant sequence. Since at least  $11 \dots 11$  is not a dominant sequence, we can limit the discussion with  $\kappa = 0, 1$ , the parameter region of possible values of  $\mu$  and  $\nu$  is restricted to a line segment in Figure 1. In this case, the total differentiation of  $f_a$  is then expressed by the following equation:

$$\begin{aligned} \Delta f_a &= \left( \frac{\partial f_a}{\partial \mu} + \frac{\partial f_a}{\partial \nu} \frac{d\nu}{d\mu} \right) \Delta \mu \\ &= \left( \ln \frac{t_{01} t_{10}}{t_{00}^2} \right) \Delta \mu \end{aligned} \quad (13)$$

From this equation, if RHS of eq. (13) is positive,  $f_a$  increases with respect to  $\mu$  and  $01 \dots 01$  becomes the dominant sequence, and conversely, if RHS of eq. (13) is negative,  $00 \dots 00$  becomes the dominant sequence. This will be determined by the following coefficient:

$$\ln \frac{t_{01} t_{10}}{t_{00}^2} = \ln \frac{t_{01} t_{10}}{t_{11}^2} - 2 \ln \frac{t_{00}}{t_{11}} \quad (14)$$

Figure 2 shows a dashed diagonal straight line in the first quadrant, on which the RHS of eq. (14) is zero. In other words, in the region above this line,  $01 \dots 01$  is the dominant sequence, and  $00 \dots 00$  is the dominant sequence in the region below the line.

From the above, we have to consider the following three dominant sequences:

- i)  $00 \dots 00$  dominant
- ii)  $01 \dots 01$  dominant
- iii)  $11 \dots 11$  dominant

For each combination of  $f_a, f_b, f_c, f_d$ , and above three dominant sequences i), ii), and iii), the following expressions are obtained giving the maximum probability with respect to  $\mu$  and  $\nu$ :

- a)-i)  $00 \dots 00$  dominant

$$\max_{\mu, \nu} (f_a(\mathbf{T}, \mu, \nu)) = p_0 t_{00}^{\lambda-1} \quad (15)$$

- a)-ii)  $01 \dots 01$  dominant

$$\max_{\mu, \nu} (f_a(\mathbf{T}, \mu, \nu)) = p_0 t_{00} t_{01}^{(\lambda-2)/2} t_{10}^{(\lambda-2)/2}, \text{ or} \quad (16)$$

$$\max_{\mu, \nu} (f_a(\mathbf{T}, \mu, \nu)) = p_0 t_{11} t_{01}^{(\lambda-2)/2} t_{10}^{(\lambda-2)/2} \quad (17)$$

b)-i) 00...00 dominant

$$\max_{\mu, \nu}(f_b(\mathbf{T}, \mu, \nu)) = p_0 t_{00}^{\lambda-2} t_{10} \quad (19)$$

b)-ii) 01...01 dominant

$$\max_{\mu, \nu}(f_b(\mathbf{T}, \mu, \nu)) = p_0 t_{01}^{(\lambda-2)/2} t_{10}^{\lambda/2} \quad (20)$$

b)-iii) 11...11 dominant

$$\max_{\mu, \nu}(f_b(\mathbf{T}, \mu, \nu)) = p_0 t_{11}^{\lambda-2} t_{10} \quad (21)$$

c)-i) 00...00 dominant

$$\max_{\mu, \nu}(f_c(\mathbf{T}, \mu, \nu)) = p_1 t_{00}^{\lambda-2} t_{01} \quad (22)$$

c)-ii) 01...01 dominant

$$\max_{\mu, \nu}(f_c(\mathbf{T}, \mu, \nu)) = p_1 t_{01}^{\lambda/2} t_{10}^{(\lambda-2)/2} \quad (23)$$

c)-iii) 11...11 dominant

$$\max_{\mu, \nu}(f_c(\mathbf{T}, \mu, \nu)) = p_1 t_{11}^{\lambda-2} t_{01} \quad (24)$$

d)-i) 00...00 dominant

$$\max_{\mu, \nu}(f_d(\mathbf{T}, \mu, \nu)) = p_1 t_{00}^{\lambda-3} t_{01} t_{10} \quad (25)$$

d)-ii) 01...01 dominant

$$\max_{\mu, \nu}(f_d(\mathbf{T}, \mu, \nu)) = p_1 t_{00} t_{01}^{(\lambda-2)/2} t_{10}^{(\lambda-2)/2}, \text{ or} \quad (26)$$

$$\max_{\mu, \nu}(f_d(\mathbf{T}, \mu, \nu)) = p_1 t_{11} t_{01}^{(\lambda-2)/2} t_{10}^{(\lambda-2)/2} \quad (27)$$

d)-iii) 11...11 dominant

$$\max_{\mu, \nu}(f_d(\mathbf{T}, \mu, \nu)) = p_1 t_{11}^{\lambda-1} \quad (28)$$

Note here that we considered the property that  $\lambda$  is even in the above expressions. Also note that all boundary conditions are considered in the above calculation, combination of first sample values and last sample values.

From the above, we have to compare 14 expressions listed in Table 1.

Table 1 Probabilities of the most or second most likely sequences of length  $\lambda$

No.	Sequence	Probability	$-\log_2(\text{Probability})/\lambda$	Notes
1	0000...0000	$p_0 \times t_{00}^{\lambda-1}$	$-\left[\frac{\lambda-1}{\lambda} \log_2 t_{00} + \frac{1}{\lambda} \log_2 p_0\right]$	<sup>a</sup> , <sup>b</sup>
2	0101...0101001010...1010	$p_0 \times t_{00} \times t_{01}^{(\lambda-2)/2} \times t_{10}^{(\lambda-2)/2}$	$-\left[\frac{\lambda-2}{2\lambda} \log_2 (t_{01} t_{10}) + \frac{1}{\lambda} \log_2 (p_0 \times t_{00})\right]$	<sup>b</sup>
3	0101...0101101010...1010	$p_0 \times t_{11} \times t_{01}^{(\lambda-2)/2} \times t_{10}^{(\lambda-2)/2}$	$-\left[\frac{\lambda-2}{2\lambda} \log_2 (t_{01} t_{10}) + \frac{1}{\lambda} \log_2 (p_0 \times t_{11})\right]$	
4	0111...1110	$p_0 \times t_{11}^{\lambda-3} \times t_{01} \times t_{10}$	$-\left[\frac{\lambda-3}{\lambda} \log_2 t_{11} + \frac{1}{\lambda} \log_2 (p_0 \times t_{01} \times t_{10})\right]$	
5	0000...0001	$p_0 \times t_{00}^{\lambda-2} \times t_{10}$	$-\left[\frac{\lambda-2}{\lambda} \log_2 t_{00} + \frac{1}{\lambda} \log_2 (p_0 \times t_{10})\right]$	
6	0101...0101	$p_0 \times t_{01}^{(\lambda-2)/2} \times t_{10}^{\lambda/2}$	$-\left[\frac{\lambda-2}{2\lambda} \log_2 (t_{01} t_{10}) + \frac{1}{\lambda} \log_2 (p_0 \times t_{10})\right]$	<sup>a</sup> , <sup>b</sup>
7	0111...1111	$p_0 \times t_{11}^{\lambda-2} \times t_{10}$	$-\left[\frac{\lambda-2}{\lambda} \log_2 t_{11} + \frac{1}{\lambda} \log_2 (p_0 \times t_{10})\right]$	<sup>a</sup> , <sup>c</sup>
8	1000...0000	$p_1 \times t_{00}^{\lambda-2} \times t_{01}$	$-\left[\frac{\lambda-2}{\lambda} \log_2 t_{00} + \frac{1}{\lambda} \log_2 (p_1 \times t_{01})\right]$	<sup>a</sup> , <sup>b</sup>
9	1010...1010	$p_1 \times t_{01}^{\lambda/2} \times t_{10}^{(\lambda-2)/2}$	$-\left[\frac{\lambda-2}{2\lambda} \log_2 (t_{01} t_{10}) + \frac{1}{\lambda} \log_2 (p_1 \times t_{01})\right]$	<sup>a</sup> , <sup>b</sup>
10	1111...1110	$p_1 \times t_{11}^{\lambda-2} \times t_{01}$	$-\left[\frac{\lambda-2}{\lambda} \log_2 t_{11} + \frac{1}{\lambda} \log_2 (p_1 \times t_{01})\right]$	
11	1000...0001	$p_1 \times t_{00}^{\lambda-3} \times t_{01} \times t_{10}$	$-\left[\frac{\lambda-3}{\lambda} \log_2 t_{00} + \frac{1}{\lambda} \log_2 (p_1 \times t_{01} \times t_{10})\right]$	
12	1010...1010100101...0101	$p_1 \times t_{00} \times t_{01}^{(\lambda-2)/2} \times t_{10}^{(\lambda-2)/2}$	$-\left[\frac{\lambda-2}{2\lambda} \log_2 (t_{01} t_{10}) + \frac{1}{\lambda} \log_2 (p_1 \times t_{00})\right]$	
13	1010...1010110101...0101	$p_1 \times t_{11} \times t_{01}^{(\lambda-2)/2} \times t_{10}^{(\lambda-2)/2}$	$-\left[\frac{\lambda-2}{2\lambda} \log_2 (t_{01} t_{10}) + \frac{1}{\lambda} \log_2 (p_1 \times t_{11})\right]$	<sup>c</sup>
14	1111...1111	$p_1 \times t_{11}^{\lambda-1}$	$-\left[\frac{\lambda-1}{\lambda} \log_2 t_{11} + \frac{1}{\lambda} \log_2 p_1\right]$	<sup>a</sup> , <sup>b</sup>

<sup>a</sup> Included in 6.3.3 of NIST SP 800-90B[1].

<sup>b</sup> Explicitly included in paragraph 516 of AIS 20/31 Version 2.35 - DRAFT [2], by using the relation  $m \equiv \lambda - 1$ .

<sup>c</sup> Implicitly included in paragraph 516 of AIS 20/31 Version 2.35 - DRAFT [2], by using the relation  $m \equiv \lambda - 1$  and by relabeling the state space  $\Omega = \{0, 1\}$ .

## 1.2 Notes for the collision estimate

In this section, we try to rewrite the following equation using elementary functions:

$$F(1/z) = \Gamma(3, z)z^{-3} \exp(z) \quad (29)$$

In NIST SP 800-90B[1], it is documented to evaluate incomplete gamma function using continued fraction. However, we try to obtain simpler expression using known properties of incomplete gamma function without using continued fraction.

First we try to use the following equations (see 8.4.8 and 8.4.11 in [3]).

$$\Gamma(n+1, z) = n! \exp(-z) e_n(z), \quad (30)$$

$$e_n(z) = \sum_{k=0}^n \frac{z^k}{k!} \quad (31)$$

With these two equations,  $F(1/z)$  can be rewritten to as follows:

$$\begin{aligned} F(1/z) &= \Gamma(3, z)z^{-3} \exp(z) \\ &= 2! \exp(-z) e_2(z) z^{-3} \exp(z) \\ &= 2z^{-3} e_2(z) \\ &= 2z^{-3} \sum_{k=0}^2 \frac{z^k}{k!} \\ &= 2z^{-3} \left( 1 + z + \frac{z^2}{2} \right) \\ &= z^{-1} (2z^{-2} + 2z^{-1} + 1) \end{aligned} \quad (32)$$

By replacing the argument  $1/z$  with  $q$ , the following expression can be obtained:

$$F(q) = q(2q^2 + 2q + 1) \quad (33)$$

From this expression, the range of  $F(q)$  is  $[0, \frac{5}{4}]$ , with respect to the domain  $q \in [0, 0.5]$ .

If we denote  $g(p)$  as the right hand side (RHS) of equation to be solved, in step 7 of 6.3.2 of NIST SP 800-90B[1],  $g(p)$  can be rewritten by using Eq.(33).

$$\begin{aligned} g(p) &\equiv pq^{-2} \left[ 1 + \frac{1}{2}(p^{-1} - q^{-1}) \right] F(q) - pq^{-1} \frac{1}{2}(p^{-1} - q^{-1}) \\ &= pq^{-2} \left[ 1 + \frac{1}{2}(p^{-1} - q^{-1}) \right] q(2q^2 + 2q + 1) - pq^{-1} \frac{1}{2}(p^{-1} - q^{-1}) \\ &= pq^{-1} \left[ 1 + \frac{1}{2}(p^{-1} - q^{-1}) \right] (2q^2 + 2q + 1) - pq^{-1} \frac{1}{2}(p^{-1} - q^{-1}) \\ &= pq^{-1} \left[ 1 + \frac{1}{2}(p^{-1} - q^{-1}) \right] - pq^{-1} \frac{1}{2}(p^{-1} - q^{-1}) \\ &\quad + pq^{-1} \left[ 1 + \frac{1}{2}(p^{-1} - q^{-1}) \right] (2q^2 + 2q) \\ &= pq^{-1} + 2p \left[ 1 + \frac{1}{2}(p^{-1} - q^{-1}) \right] (q + 1) \\ &= pq^{-1} + 2p \left[ (q + 1) + \frac{1}{2}p^{-1}(q + 1) - \frac{1}{2}(1 + q^{-1}) \right] \\ &= pq^{-1} + p[2(q + 1) + p^{-1}(q + 1) - (1 + q^{-1})] \\ &= p[2(q + 1) + p^{-1}(q + 1) - 1] \\ &= p[2q + 1 + p^{-1}(q + 1)] \\ &= (2pq + p + q + 1) \\ &= (2pq + 2) \\ &= 2(pq + 1) \\ &= 2[p(1 - p) + 1] \\ &= 2 \left[ -\left(p - \frac{1}{2}\right)^2 + \frac{5}{4} \right] \end{aligned} \quad (34)$$

Figure 3 shows Eq.(34) graphically.

If  $\bar{X}'$  is in range  $[2, \frac{5}{2}]$ , the solution  $p$  of step 7 of 6.3.2 of NIST SP 800-90B[1] can be expressed by the following equation:

$$p = \frac{1}{2} + \sqrt{\frac{5}{4} - \frac{\bar{X}'}{2}}. \quad (35)$$

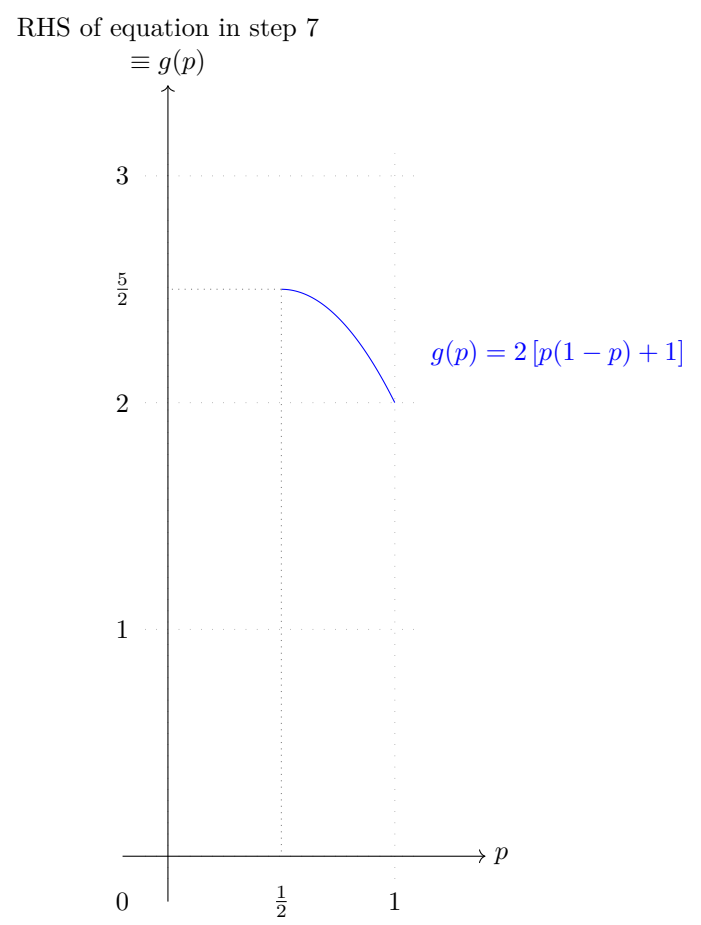


Figure 3 The right hand side of the equation in step 7 of 6.3.2 of NIST SP 800-90B

### 1.3 Notes for the compression estimate

As documented in [4], Eq.(36) should be replaced by Eq.37.

$$G(z) = \frac{1}{\nu} \sum_{t=d+1}^L \sum_{u=1}^t \log_2(u) F(z, t, u) \quad (36)$$

$$G(z) = \frac{1}{\nu} \sum_{t=d+1}^{\lfloor L/b \rfloor} \sum_{u=2}^t \log_2(u) F(z, t, u) \quad (37)$$

$F(z, t, u)$  in Eq.(37) can be expressed by the following equation:

$$F(z, t, u) = \begin{cases} z^2(1-z)^{u-1} & \text{if } u < t \\ z(1-z)^{t-1} & \text{if } u = t \end{cases} \quad (38)$$

We have to evaluate Eq.(37), but this expression requires relatively large computational complexity due to its nested summation. First Figure 4 shows parameters  $u, t$ , where nested summation must be taken to compute  $G(z)$ .

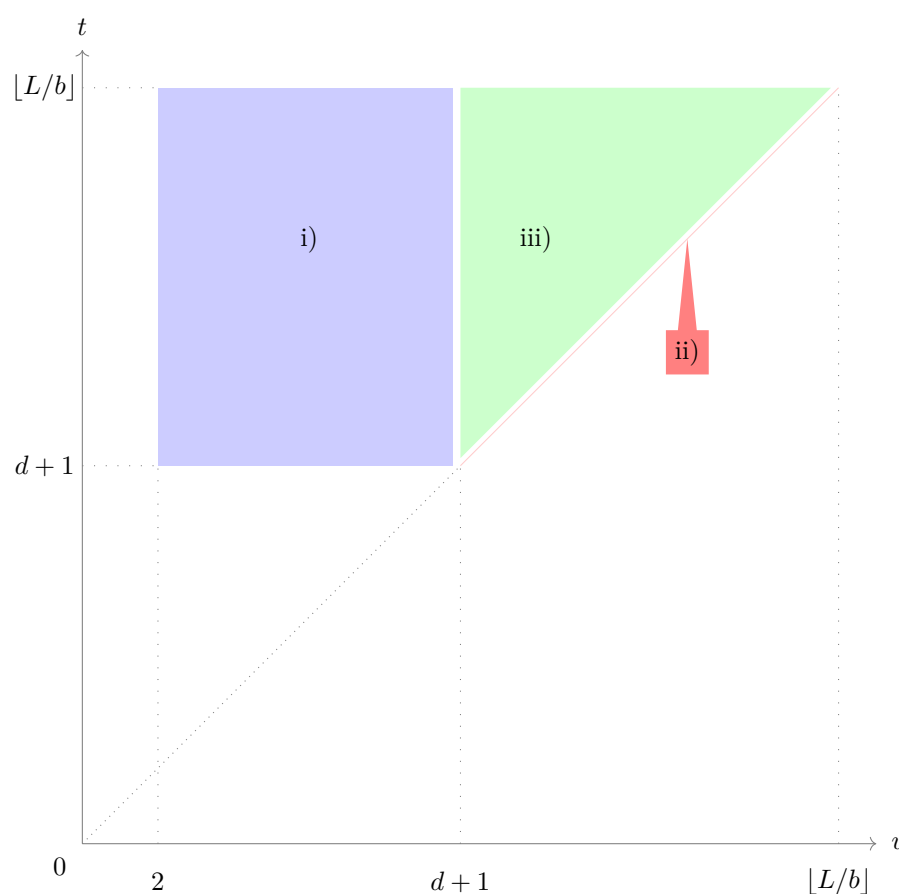


Figure 4 Parameters  $u, t$ , where nested summation must be taken

$t$ -dependency of  $F(z, t, u)$  must be considered when  $u = t$ . From the above, the parameters domain where summation must be taken can be divided into the following three groups:

i)

$$\begin{cases} 2 \leq u < d+1 \\ d+1 \leq t \leq \lfloor L/b \rfloor \end{cases} \quad (39)$$

ii)

$$\begin{cases} (u, t) = (\tau, \tau) \\ d+1 \leq \tau \leq \lfloor L/b \rfloor \end{cases} \quad (40)$$

iii)

$$\begin{cases} d+1 \leq u < t \\ d+1 < t \leq \lfloor L/b \rfloor \end{cases} \quad (41)$$

For groups i) and iii), the summation with respect to  $t$  can be taken first. By using the relation  $\nu = \lfloor L/b \rfloor - d$ , Eq.(37) can be rewritten to as follows:

$$\begin{aligned} G(z) &= \sum_{u=2}^d \log_2(u) z^2 (1-z)^{u-1} \\ &+ \frac{1}{\nu} \sum_{u=d+1}^{\lfloor L/b \rfloor} \log_2(u) z (1-z)^{u-1} \\ &+ \frac{1}{\nu} \sum_{u=d+1}^{\lfloor L/b \rfloor - 1} (\lfloor L/b \rfloor - u) \log_2(u) z^2 (1-z)^{u-1} \end{aligned} \quad (42)$$

With this expression, the computational complexity can be decreased from  $\mathcal{O}(\nu^2)$  to  $\mathcal{O}(\nu)$ .

### 1.4 Notes for t-Tuple estimate, LRS estimate, MultiMMC Prediction and LZ78Y prediction estimate

In the following groups of entropy estimate, we have to handle  $t$ -tuples (pairs, triples, etc.)

a)  $t$ -Tuple estimate (NIST SP 800-90B 6.3.5)

b) Longest Repeated Substrings (LRS) estimate (NIST SP 800-90B 6.3.6)

### 1.5 Notes for Multi Most Common in Window prediction estimate

In step 3-a-i of 6.3.7 of NIST SP 800-90B, it is required to compute the mode in the previous window of  $w_j$  before  $s_i$ . As the order of  $w_j$  is 1000, the computational complexity is estimated to be about  $\mathcal{O}(1000n)$ , and expected to be time-consuming. We attempt to reduce time-complexity by preparing histograms of certain lengths in advance, and in step 3-a-i, we use the histograms that fit within the target window, and compute the unavailable parts on-demand basis.

## 2 Additional implementation notes for t-Tuple estimate, LRS estimate using Longest Common Prefix

### 2.1 An issue of naive implementation for t-Tuple estimate and LRS estimate

The naive implementation of t-Tuple estimate and LRS estimate consumes huge memory  $\approx$  few giga-bytes, to store unique t-Tuple. Here, it is stated that Longest Common Prefix can be used for t-Tuple and LRS estimates, and naive algorithm is provided[5]. Note here that [6] provides good introduction for Longest Common Prefix and its calculating algorithm. Next optimized algorithms are provided to perform t-Tuple and LRS estimates[7]. However, it is unclear about how to derive the whole algorithms, except for “Count Aggregation” (hereafter denoted as “cascading algorithm”).

So let us consider a more intuitive algorithm to perform t-Tuple and LRS estimates in the following subsection.

### 2.2 Derivation of more intuitive algorithms for t-Tuple estimate and LRS estimate

Provided that Longest Common Prefix array ( $LCP[1], \dots, LCP[L]$ ) has been calculated from input  $\mathbf{S}$ ,  $LCP[j]$  changes with respect to index  $j$ , so let us consider the following three cases:

- a)  $LCP[j-1] < LCP[j]$  ( $j \in [2, L]$ ),
- b)  $LCP[j-1] = LCP[j]$  ( $j \in [2, L]$ ),
- c)  $LCP[j-1] > LCP[j]$  ( $j \in [2, L]$ ).

We may call case a) as “climb up” case, case b) as “traverse” case, and case c) as “climb down” case.

For brevity, let us use  $\eta_j$  to denote  $LCP[j]$ , or simply  $\eta$  by omitting index  $j$ . Let us consider a run length of LCP array at index  $j$ , and denote  $\lambda_{\eta_j}$  as the run length.

Normally several minimum and maximum points can be found in LCP array, so let us consider the range from one local minimum to the next as a single “peak”, and introduce an array  $\tilde{q}_{\text{work}}[\eta]$  representing run length value with respect to the “height”  $\eta$  (which is equal to a value in LCP array) in currently climbing peak. Considering that normally there are multiple peaks in LCP array, let us introduce an array  $\tilde{q}_{\text{master}}[\eta]$  storing the maximum run length for each  $\eta$  appeared in the LCP array.

For case a), let us store the previous run length value  $\lambda_{\eta_{j-1}}$  to  $\tilde{q}_{\text{work}}[\eta_{j-1}]$ , and assign new run length value  $\lambda_{\eta_j}$  as 1. Note that the part of algorithm is highlighted in light blue in Algorithms 1 and 2.

For case b), let us increment  $\lambda_{\eta_j}$  by 1. Note that the part of algorithm is highlighted in light green in Algorithms 1 and 2.

For case c), the run lengths of LCP values for ( $LCP[j] + 1, \dots, LCP[j-1]$ ) can be determined by Algorithm 3, which is “cascading algorithm”. Note that the part of algorithm is highlighted in pale purple in Algorithms 1 and 2.

For case c) or “climb down” case, update  $\tilde{q}_{\text{master}}$  to store longer run lengths, and also run lengths for higher  $\eta$  values, by comparing  $\tilde{q}_{\text{work}}$  and  $\tilde{q}_{\text{master}}$ .

Finally the required array  $Q[i]$  where  $i \in [1, t]$  can be calculated by the following relation:

$$Q[i] = \tilde{q}_{\text{master}}[i] + 1. \quad (43)$$



### 2.3 Additional implementation notes for t-Tuple estimate using Longest Common Prefix

---

**Algorithm 1** Rewritten steps 1 and 2 in  $t$ -Tuple Estimate

---

**Input:** input dataset  $\mathbf{S} = (s_1, \dots, s_L)$  where  $s_i \in A = \{x_1, \dots, x_k\}$

**Output:** an integer array of  $i$ -tuple counts for  $i \in \{1, 2, \dots, t\}$

```

1: Compute the Longest Common Prefix array  $LCP[j](j \in \{1, 2, \dots, L\})$  from  $\mathbf{S}$ .
2:  $\eta_{\max} \leftarrow \max_{j \in \{1, 2, \dots, L\}} LCP[j]$ 

▷  $\eta_{\max}$  coincide with  $\nu$  in Longest Repeated Substring (LRS) Estimate.



3: for  $\eta \leftarrow 1, \max(\eta_{\max}, 1)$  do
4:    $\tilde{q}_{\text{master}}[\eta] \leftarrow 0$ 
5:    $\tilde{q}_{\text{work}}[\eta] \leftarrow 0$ 
6: end for
7:  $\eta_{\text{prev}} \leftarrow LCP[1]$ 
8:  $\lambda \leftarrow 1$ 
9: for  $j \leftarrow 2, L$  do
10:   $\eta_{\text{current}} \leftarrow LCP[j]$ 
11:  if  $\eta_{\text{prev}} < \eta_{\text{current}}$  then
12:    if  $0 < \eta_{\text{prev}}$  then
13:       $\tilde{q}_{\text{work}}[\eta_{\text{prev}}] \leftarrow \lambda$ 
14:    end if
15:     $\lambda \leftarrow 1$ 
16:  else if  $\eta_{\text{prev}} = \eta_{\text{current}}$  then
17:    if  $\eta_{\text{current}} = 0$  then

▷ This condition corresponds to consecutive 0-values in array LCP.


18:       $\lambda \leftarrow 1$ 

▷ consecutive 0-values in array LCP mean different 1-tuple.


19:    else
20:       $\lambda \leftarrow \lambda + 1$ 
21:    end if
22:  else
23:    AccumulateLambda( $\tilde{q}_{\text{work}}, \eta_{\text{prev}}, \eta_{\text{current}}, \lambda$ )
24:    UpdateLambda( $\lambda, \tilde{q}_{\text{work}}, \eta_{\text{current}}$ )
25:    UpdateQTildeMaster( $\tilde{q}_{\text{master}}, \tilde{q}_{\text{work}}, \eta_{\text{prev}}, \eta_{\text{current}}$ )
26:    ResetQTildeWork( $\tilde{q}_{\text{work}}, \eta_{\text{prev}}, \eta_{\text{current}}$ )
27:  end if
28:   $\eta_{\text{prev}} \leftarrow \eta_{\text{current}}$ 
29: end for

30: AccumulateLambda( $\tilde{q}_{\text{work}}, \eta_{\text{prev}}, 0, \lambda$ )
31: UpdateQTildeMaster( $\tilde{q}_{\text{master}}, \tilde{q}_{\text{work}}, \eta_{\text{prev}}, 0$ )
32:  $t \leftarrow \max(\{\eta \in \{1, \dots, \eta_{\max}\} \mid \tilde{q}_{\text{master}}[\eta] \geq 34\})$ 

▷ Find the largest  $t$  such that the number of occurrences of the most common  $t$ -tuple in  $\mathbf{S}$  is at least 35. This can be rewritten as shown above using  $\tilde{q}_{\text{master}}$ .


33: return CalcQ( $\tilde{q}_{\text{master}}, t$ )

```

---

## 2.4 Additional implementation notes for LRS estimate using Longest Common Prefix

---

**Algorithm 2** Rewritten steps 1, 2 and 3 in Longest Repeated Substring (LRS) Estimate

---

**Input:** input dataset  $\mathbf{S} = (s_1, \dots, s_L)$  where  $s_i \in A = \{x_1, \dots, x_k\}$

**Output:** string *status*, a real value  $\hat{p}$

```

1: Compute the Longest Common Prefix array  $LCP[j](j \in \{1, 2, \dots, L\})$  from  $\mathbf{S}$ .
2:  $\eta_{\max} \leftarrow \max_{j \in \{1, 2, \dots, L\}} LCP[j]$ 
3: if  $\eta_{\max} < 1$  then
4:   return (“ $\nu$  not found”, NaN)
                                     ▷  $\nu$  cannot be found, therefore estimate cannot be computed.

5: end if
6:  $\nu \leftarrow \eta_{\max}$ 
                                     ▷ It is ensured that  $\nu \geq 1$ .

7: for  $\eta \leftarrow 1, \nu$  do
8:    $\tilde{q}_{\text{master}}[\eta] \leftarrow 0$ 
9:    $\tilde{q}_{\text{work}}[\eta] \leftarrow 0$ 
10:   $\sigma[\eta] \leftarrow 0$ 
11: end for
12:  $\eta_{\text{prev}} \leftarrow LCP[1]$ 
13:  $\lambda \leftarrow 1$ 
14: for  $j \leftarrow 2, L$  do
15:    $\eta_{\text{current}} \leftarrow LCP[j]$ 
16:   if  $\eta_{\text{prev}} < \eta_{\text{current}}$  then

17:     if  $0 < \eta_{\text{prev}}$  then
18:        $\tilde{q}_{\text{work}}[\eta_{\text{prev}}] \leftarrow \lambda$ 
19:     end if
20:      $\lambda \leftarrow 1$ 

21:   else if  $\eta_{\text{prev}} = \eta_{\text{current}}$  then

22:     if  $\eta_{\text{current}} = 0$  then
23:        $\lambda \leftarrow 1$ 
24:     else
25:        $\lambda \leftarrow \lambda + 1$ 
26:     end if
27:   else

28:     AccumulateLambda( $\tilde{q}_{\text{work}}, \eta_{\text{prev}}, \eta_{\text{current}}, \lambda$ )
29:     UpdateNumerator( $\sigma, \tilde{q}_{\text{work}}, \eta_{\text{prev}}, \eta_{\text{current}}$ )
30:     UpdateLambda( $\lambda, \tilde{q}_{\text{work}}, \eta_{\text{current}}$ )
31:     UpdateQTildeMaster( $\tilde{q}_{\text{master}}, \tilde{q}_{\text{work}}, \eta_{\text{prev}}, \eta_{\text{current}}$ )
32:     ResetQTildeWork( $\tilde{q}_{\text{work}}, \eta_{\text{prev}}, \eta_{\text{current}}$ )

33:   end if
34:    $\eta_{\text{prev}} \leftarrow \eta_{\text{current}}$ 
35: end for

36: AccumulateLambda( $\tilde{q}_{\text{work}}, \eta_{\text{prev}}, 0, \lambda$ )
37: UpdateNumerator( $\sigma, \tilde{q}_{\text{work}}, \eta_{\text{prev}}, 0$ )
38: UpdateQTildeMaster( $\tilde{q}_{\text{master}}, \tilde{q}_{\text{work}}, \eta_{\text{prev}}, 0$ )
39:  $u \leftarrow \min(\{\eta \in \{1, \dots, \nu\} \mid \tilde{q}_{\text{master}}[\eta] < 34\})$ 
   ▷ Find the smallest  $u$  such that the number of occurrences of the most common  $u$ -tuple in  $\mathbf{S}$  is less than 35. This can be rewritten as shown
   above using  $\tilde{q}_{\text{master}}$ .
40:  $\hat{p} \leftarrow \text{CalcPHat}(\sigma, u, \nu)$ 
   ▷ Compute the estimated average collision probability per string symbol as  $P_{\max, W} = (P_W)^{\frac{1}{W}}$ . Let  $\hat{p} = \max(P_{\max, u}, \dots, P_{\max, \nu})$ .
41: return (“Success”,  $\hat{p}$ )

```

---

## 2.5 Auxiliary routines

---

### Algorithm 3 AccumulateLambda

---

```

1: procedure ACCUMULATELAMBDA( $\tilde{\mathbf{q}}_{\text{work}}, \eta_{\text{prev}}, \eta_{\text{current}}, \lambda$ )
2:    $\tilde{q}_{\text{work}}[\eta_{\text{prev}}] \leftarrow \lambda$ 
3:   for  $\eta \leftarrow (\eta_{\text{prev}} - 1)$  downto  $\max(\eta_{\text{current}}, 1)$  do
4:      $\tilde{q}_{\text{work}}[\eta] \leftarrow \tilde{q}_{\text{work}}[\eta] + \tilde{q}_{\text{work}}[\eta + 1]$ 
▷ sum up  $\lambda$  and update the array  $\tilde{q}$ 
5:   end for
6: end procedure

```

---



---

### Algorithm 4 UpdateLambda

---

```

1: procedure UPDATELAMBDA( $\lambda, \tilde{\mathbf{q}}_{\text{work}}, \eta_{\text{current}}$ )
2:   if  $\eta_{\text{current}} = 0$  then
3:      $\lambda \leftarrow 1$ 
4:   else
5:      $\lambda \leftarrow (\tilde{q}_{\text{work}}[\eta_{\text{current}}] + 1)$ 
6:   end if
7: end procedure

```

---



---

### Algorithm 5 UpdateQTildeMaster

---

```

1: procedure UPDATEQTILDEMASTER( $\tilde{\mathbf{q}}_{\text{master}}, \tilde{\mathbf{q}}_{\text{work}}, \eta_{\text{prev}}, \eta_{\text{current}}$ )
2:   for  $\eta \leftarrow \eta_{\text{prev}}$  downto  $(\eta_{\text{current}} + 1)$  do
3:     if  $\tilde{q}_{\text{master}}[\eta] < \tilde{q}_{\text{work}}[\eta]$  then
4:        $\tilde{q}_{\text{master}}[\eta] \leftarrow \tilde{q}_{\text{work}}[\eta]$ 
▷ update  $\eta$ -tuple count
5:     end if
6:   end for
7: end procedure

```

---



---

### Algorithm 6 ResetQTildeWork

---

```

1: procedure RESETQTILDEWORK( $\tilde{\mathbf{q}}_{\text{work}}, \eta_{\text{prev}}, \eta_{\text{current}}$ )
2:   for  $\eta \leftarrow \eta_{\text{prev}}$  downto  $\max(\eta_{\text{current}}, 1)$  do
3:      $\tilde{q}_{\text{work}}[\eta] \leftarrow 0$ 
▷ reset to zero for future computation
4:   end for
5: end procedure

```

---



---

### Algorithm 7 CalcQ

---

```

1: function CALCQ( $\tilde{\mathbf{q}}_{\text{master}}, t$ )
2:   for  $\eta \leftarrow 1, t$  do
3:      $Q[\eta] \leftarrow \tilde{q}_{\text{master}}[\eta] + 1$ 
4:   end for
5:   return  $\mathbf{Q} = (Q[1], \dots, Q[t])$ 
6: end function

```

---



---

### Algorithm 8 UpdateNumerator

---

```

1: procedure UPDATENUMERATOR( $\sigma, \tilde{\mathbf{q}}_{\text{work}}, \eta_{\text{prev}}, \eta_{\text{current}}$ )
2:   for  $\eta \leftarrow \eta_{\text{prev}}$  downto  $(\eta_{\text{current}} + 1)$  do
▷ Here,  $\tilde{q}_{\text{work}}[\eta_{\text{current}}]$  is a snapshot at a given point in time, therefore, updating  $\sigma[\eta]$  where  $\eta > \eta_{\text{current}}$  makes sense.
3:     if  $\tilde{q}_{\text{work}}[\eta] + 1 \geq 2$  then
4:        $\sigma[\eta] \leftarrow \sigma[\eta] + \binom{\tilde{q}_{\text{work}}[\eta] + 1}{2}$ 
5:     end if
6:   end for
7: end procedure

```

---



---

### Algorithm 9 CalcPHat

---

```

1: function CALCPHAT( $\sigma, u, \nu$ )
2:   for  $W \leftarrow u, \nu$  do
3:      $P[W] \leftarrow \frac{\sigma[W]}{\binom{L-W+1}{2}}$ 
4:   end for
5:    $\hat{p} \leftarrow \max \left( (P[u])^{\frac{1}{u}}, \dots, (P[\nu])^{\frac{1}{\nu}} \right)$ 
6:   return  $\hat{p}$ 
7: end function

```

---

### 3 References

- [1] Meltem Sönmez Turan, Elaine Barker, John Kelsey, Kerry A. McKay, Mary L. Baish, Mike Boyle *Recommendation for the Entropy Sources Used for Random Bit Generation*, NIST Special Publication 800-90B, Jan. 2018
- [2] M. Peter and W. Schindler, *A Proposal for Functionality Classes for Random Number Generators* Version 2.35 - DRAFT, September 2, 2022. [https://www.bsi.bund.de/SharedDocs/Downloads/EN/BSI/Certification/Interpretations/AIS\\_31\\_Functionality\\_classes\\_for\\_random\\_number\\_generators\\_e.pdf?\\_\\_blob=publicationFile&v=7](https://www.bsi.bund.de/SharedDocs/Downloads/EN/BSI/Certification/Interpretations/AIS_31_Functionality_classes_for_random_number_generators_e.pdf?__blob=publicationFile&v=7)
- [3] Franck W. J. Oliver, Daniel W. Lozier, Ronald F. Boisvert, Charles W. Clark, *NIST Handbook of Mathematical Functions*, National Institute of Standards and Technology, 2010
- [4] G. Sakurai, *Proposed list of corrections for NIST SP 800-90B 6.3 Estimators*, Dec. 2022 [https://github.com/g-g-sakura/AnotherEntropyEstimationTool/blob/main/documentation/ProposedListOfCorrections\\_SP800-90B.pdf](https://github.com/g-g-sakura/AnotherEntropyEstimationTool/blob/main/documentation/ProposedListOfCorrections_SP800-90B.pdf)
- [5] Joshua E. Hill, *Algorithms for  $t$ -tuple and LRS Estimates*, June 11, 2018 <https://www.untruth.org/~josh/sp80090b/NIST.SP.800-90B%20implementation%20comments%2020191219.pdf>
- [6] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest and Clifford Stein, *Introduction to Algorithms (fourth edition)*, The MIT Press. <https://mitpress.mit.edu/9780262046305/introduction-to-algorithms/>
- [7] Aaron Kaufer, *Further Improvements for SP 800-90B Tuple Counts*, April 10, 2019 <https://www.untruth.org/~josh/sp80090b/Kaufer%20Further%20Improvements%20for%20SP%20800-90B%20Tuple%20Counts.pdf>