An Efficient and Compact Reformulation of NIST Collision Estimate Test

P. R. Mishra* Bhartendu Nandan† Navneet Gaba‡

Abstract

In this paper we give an efficient and compact reformulation of NIST collision estimate test given in SP-800 90B. We correct an error in the formulation of the test and show that the test statistic can be computed in a much easier way. We also propose a revised algorithm for the test based on our findings.

1 Introduction

NIST's collision estimate test is one of the ten tests given in NIST special publication SP-800 90B [1] for entropy estimation for non-IID data. This test is based on the collision estimate proposed by Hagerty and Draper [2]. It calculates the probability of the most-likely output value, based on the collision of bits in a binary data.

This test mainly comprises three steps viz., counting and storing the distance between consecutive indices where collisions are found, computing a modified value of mean with help of mean and standard deviation and carrying out binary search to find the probability. We propose an alternate formulation of the first and the third steps. We indicate an error in step three and give the correct formulation. We also show that with our formulation, the costly binary search can be replaced with a single square root computation. The paper is structured in the following manner.

In the next section we provide description of Collision Estimate Test given in NIST special publication SP-800 90B [1]. In the third section, we provide an alternate formulation of step 1. In the fourth section we discuss an error in NIST's description of the test and give our corrected formulation. The next

*prasanna.r.mishra@gmail.com
†bhartendun@gmail.com
‡navneetgaba2000@gmail.com

i.e., the fifth section contains an algorithm for the test based on our findings and the computation of its complexity. We take NIST's implementation of the test as a benchmark and compare the timings for different data sets.

2 NIST's Description of Collision Estimate Test[1]

Given the input $S = (s_1, \ldots, s_L)$, where $s_i \in A = \{0, 1\}$,

1. Set $v = 0, index = 1$.
2. Beginning with s_{index}, step through the input until any observed value is repeated; i.e., find the smallest j such that $s_i = s_j$, for some i with $index \leq i < j$.
3. Set $v = v + 1$, $t_v = j - index + 1$ and $index = j + 1$.
4. Repeat steps 2-3 until the end of the dataset is reached.
5. Calculate the sample mean $\bar{X}$, and the sample standard deviation ($\hat{\sigma}$), of t_i as
$$\bar{X} = \frac{1}{v}\sum_{i=1}^{v} t_i, \hat{\sigma} = \sqrt{\frac{1}{v-1}\sum_{i=1}^{v}(t_i - \bar{X})^2}$$
6. Compute the lower-bound of the confidence interval for the mean, based on a normal distribution [3] with a confidence level of 99 %,
$$\bar{X} = X - 2.576\frac{\hat{\sigma}}{\sqrt{v}}$$
7. Using a binary search (bisection method [4]), solve for the parameter p, such that
$$\bar{X}' = pq^{-2}\left(1 + \frac{1}{2}(p^{-1} - q^{-1})\right)F(q) - pq^{-1}\frac{1}{2}(p^{-1} - q^{-1}) \quad (1)$$
where
$$q = 1 - p, \quad (2)$$
$$p \geq q,$$
$$F(1/z) = \Gamma(3, z)z^{-3}e^z,$$
and Γ is the incomplete Gamma function [6]. The bounds of the binary search should be 1/2 and 1.

8. If the binary search yields a solution, then the *min-entropy* estimation is the negative logarithm of the parameter, p:

$$min\text{-}entropy = -\log_2(p).$$

If the search does not yield a solution, then the *min-entropy* estimation is:

$$min\text{-}entropy = \log_2(2) = 1.$$

3 Counting and Storing Collision Intervals

Since the binary sequences contain only 0 and 1, any 3-bit pattern must contain a collision and the collision length is either 2 or 3. Let (b_1, b_2, b_3) be a 3-bit pattern. The collision length corresponding to different values of b_1, b_2 and b_3 are given in the following table. It may be verified that the collision

b_1	b_2	b_3	Collision length (t)
0	0	0	2
0	0	1	2
0	1	0	3
0	1	1	3
1	0	0	3
1	0	1	3
1	1	0	2
1	1	1	2

length for 3-bit patterns follows the following relation.

$$t = (b_1 \oplus b_2) + 2$$

The only cases left are the left over 1-bit or 2-bit pattern at the end of the sequence which can be catered separately. If there is single at the end, just discard it. If there are two identical bits, the collision length will be two else discard the case.

4 Computation of Probability for Min-Entropy

4.1 Error in Description of Test

Refer to section 6.3.2 of [1]. For the Collision Estimate test, a function $F(1/z)$ has to be calculated. It is defined as

$$F(1/z) = \Gamma(3, z)z^{-3}e^z.$$

For the efficient implementation of this function, a continued fraction representation of $F(1/z)$ is given in Appendix-G.1.1 on page 74 of [1]. It is written that, *The function F(1/z), used by the collision estimate (Section 6.3.2), can be approximated by the following continued fraction:*

$$\cfrac{1}{z+\cfrac{-k}{1+\cfrac{1}{1+\cfrac{1-k}{1+\cfrac{2}{2-k\ldots}}}}} \tag{3}$$

We observed that an extra parameter k appears in the above continued fraction. It was found that this continued fraction occurs in representation of $\Gamma(k,z)$ [5]. For a positive integer k and a non-zero complex number z, we have,

$$\Gamma(k,z) = \cfrac{z^k e^{-z}}{z+\cfrac{-k}{1+\cfrac{1}{1+\cfrac{1-k}{1+\cfrac{2}{2-k\ldots}}}}} \tag{4}$$

It is clear from (4) that (3) can be equal to $F(1/z)$ only when $k=3$, and not for an arbitrary value of k as stated in the document.

4.2 Polynomial Expression for $F(z)$

Based on the observation of previous subsection, we derive a polynomial expression for F which is given in the following proposition.

Proposition 1. *The function F used in Collision test can be written as*

$$F(z) = 2z^3 + 2z^2 + z$$

Proof. For $s \in \mathbb{N}$ and $z \in \mathbb{R}_{\geq 0}$,

$$\Gamma(s,z) = (s-1)!e^{-z}\sum_{k=0}^{s-1}\frac{z^k}{k!} \tag{5}$$

Putting $s=3$ in (5), we have,

$$\begin{aligned}\Gamma(3,z) &= 2!e^{-z}\sum_{k=0}^{2}\frac{z^k}{k!} \\ &= 2e^{-z}(1+z+\frac{z^2}{2})\end{aligned}$$

$$\begin{aligned}F(1/z) &= \Gamma(3,z)z^{-3}e^z = 2e^{-z}(1+z+\frac{z^2}{2})z^{-3}e^z \\ &= 2z^{-3}+2z^{-2}+z^{-1}\end{aligned}$$

Therefore,

$$F(z) = 2z^3 + 2z^2 + z$$

□

Remark. *For implementation purpose, we can write $F(z)$ as*

$$F(z) = z(z(2z + 2) + 1) \tag{6}$$

This takes three floating point multiplications and two floating point additions or two floating point multiplications and three floating point additions when $2z$ is written as $z + z$. Thus this implementation is quite simpler and more efficient than the implementation of the continued fraction given in the document.

4.3 Avoiding the Binary Search

Binary search is generally used for approximating roots of an equation in cases where it is not possible to solve the equation explicitly. In this section we show that (1) can be explicitly solved in terms of p, and thus the binary search can be avoided. We first state the following theorem:

Theorem 1. *(1) can be written as*

$$\bar{X}' = -2p^2 + 2p + 2. \tag{7}$$

Proof. We have from (1)

$$\bar{X}' = pq^{-2}\left(1 + \frac{1}{2}(p^{-1} - q^{-1})\right) F(q) - pq^{-1}\frac{1}{2}(p^{-1} - q^{-1}) \tag{8}$$

Using proposition (1),

$$\begin{aligned}
\bar{X}' &= pq^{-2}\left(1 + \frac{1}{2}(p^{-1} - q^{-1})\right)(2q^3 + 2q^2 + q) - pq^{-1}\frac{1}{2}(p^{-1} - q^{-1}) \\
&= pq^{-1}\left(2q^2 + 2q + 1 + \frac{1}{2}(p^{-1} - q^{-1})(2q^2 + 2q + 1) - \frac{1}{2}(p^{-1} - q^{-1})\right) \\
&= p\left(2q + 2 + q^{-1} + p^{-1}q - 1 + p^{-1} - q^{-1}\right) \\
&= 2pq + q + p + 1
\end{aligned} \tag{9}$$

Using (2) in (9) we get the required result.

□

Further, (7) can be written as

$$p^2 - p + \left(\frac{\bar{X}'}{2} - 1\right) = 0 \tag{10}$$

Solving (10) gives

$$\begin{aligned} p &= \frac{1 \pm \sqrt{1 - 4\left(\frac{\bar{X}'}{2} - 1\right)}}{2} \\ &= \frac{1 \pm \sqrt{5 - 2\bar{X}'}}{2} \end{aligned} \tag{11}$$

Clearly, (10) will have real roots if and only if $\bar{X}' \leq 2.5$.
Since $p \in [0.5, 1]$, we can discard -ve sign and thus we have

$$\begin{aligned} & \frac{1}{2} \leq \frac{1 + \sqrt{5 - 2\bar{X}'}}{2} \leq 1 \\ \implies \quad & 0 \leq \sqrt{5 - 2\bar{X}'} \leq 1 \\ \implies \quad & 0 \leq 5 - 2\bar{X}' \leq 1 \\ \implies \quad & 2 \leq \bar{X}' \leq 2.5 \end{aligned}$$

This means (8) is solvable for $p \in [0.5, 1]$ if and only if $2 \leq \bar{X}' \leq 2.5$. Once the solvability is ensured, p can be computed using (11) taking +ve sign.

5 Our algorithm and experimental results

Based on our observations, we propose the following algorithm for computation of min-entropy.

1: **function** COLLISION_ESTIMATE($S = \{s_i \mid i = 0, 1, \ldots, L-1\}$)
2: $v \leftarrow 0, index \leftarrow 0, sum \leftarrow 0, \sigma' \leftarrow 0.$
3: $lim \leftarrow L - 2$
4: **while** $(index < lim)$ **do**
5: $t_v \leftarrow (s_{index} \oplus s_{index+1}) + 2$
6: $index \leftarrow index + t_v$
7: $sum \leftarrow sum + t_v$
8: $v \leftarrow v + 1$
9: **end while**
10: $t_v \leftarrow (s_{index} \oplus s_{index+1}) + 2$

```
    if index = lim and t_v = 2 then
        sum ← sum + t_v
        v ← v + 1
    end if
    X̄ ← sum/v
    for i ← 0 to v − 1 do
        σ′ ← σ′ + (X̄ − t_i)^2
    end for
    σ′ ← sqrt(σ′/(v(v−1)))
    X̄′ ← X̄ − 2.576σ′
    if X̄′ ∈ [2, 2.5] then
        p ← (1+sqrt(5−2X̄′))/2
        min_entropy ← − log_2 p
    else
        min_entropy ← 1
    end if
    return min_entropy
end function
```

The correctness of the algorithm is evident in view of Sections 3 and 4.

Remark. *In our algorithm, the starting index of the sequence is taken to be zero.*

5.1 Comparison of Timings

To compare efficiency of our algorithm, four sets of 1000 binary sequences each, of lengths 1000, 10000, 100000 and 1000000 bits were taken. On each of the four sets NIST algorithm and our algorithm were run. The experiments were performed on an i-7 machine with 4GB of RAM. The timings are compared in table 1.

S.No.	Length of the sequence in bits	Time taken in secs	
		NIST Algorithm	Our Algorithm
1	1000	0.19	0.17
2	10000	0.37	0.31
3	100000	2.34	1.65
4	1000000	21.17	14.99

Table 1: Comparison of timings of NIST's algorithm and our algorihtm

References

[1] Meltem Sonmez 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, January, 2018.

[2] P. Hagerty and T. Draper; Entropy Bounds and Statistical Tests, NIST Random Bit Generation Workshop, December 2012, https://csrc.nist.gov/csrc/media/events/random-bit-generation-workshop-2012/documents/hagerty_entropy_paper.pdf.

[3] K. Krishnamoorthy, Handbook of Statistical Distribution with Applications,Chapman and Hall, 2006.

[4] Richard L. Burden, J. Douglas Flairs, Numerical Analysis, 8th Edition, Brooks/Cole Cengage Learning. .

[5] Annie A. M. Cuyt, Vigdis Peterson, Briette Verdonk, H. Waadeland, W. B. Jones, Handbook of Continued Fraction for Special Functions, Springer, 2008.

[6] G. J. O. Jameson, Notes on Incomplete Gamma Function, www.maths.lancs.ac.uk/jameson/gammainc.pdf