Numerical computation of false alarm probability for multiple trials

Masaharu Hirayama

January 9, 2007

1 Problem

Consider a statistical trial that can result in various events, each of which occurs at a fixed probability. When a certain event occurs at probability of p ($0 \leq p \leq 1$) for a single trial, a probability Q that the event occurs at least once for N trials ($N > 0$) can be written as

$$Q = 1 - (1 - p)^N. \tag{1}$$

Note that, however, when X is a value in range $0 \leq X \leq 1$, term $(1 - X)$ becomes computationally challenging in two cases: either X is very small, or very close to 1. Let ϵ be the difference between 1.0 and the smallest representable value greater than 1.0. In the former case, a regular computation of $(1 - X)$ results in 1 for $X < \epsilon$, In the latter case, $(1 - X)$ gives zero for $X > 1 - \epsilon$. In either case, a result from a regular computation may not be a desired one.

Terms in a form of $(1 - X)$ appear in Eq. 1 in two places: one with $X = p$ and the other with $X = (1-p)^N$. Special care must be taken in those places to obtain a desired result. Illustrated below is a procedure to compute Q for an arbitrary combination of p and N, minimizing loss of precision during the computation.

2 Computational difficulties

Computation of Q can be difficult for certain values of p and N, as mentioned in Section 1. This section summarizes such cases and discusses limitations

of computation of Q. There are four cases that may be problematic in computing Q, and those cases are discussed in detail below.

Case 1: $p < \epsilon$

In this case, Q can be any number in range $0 \leq Q \leq 1$. Even though $(1-p)$ is very close to 1, $(1-p)^N$ can be very small for a large N, for which Q becomes close to 1. For a small N, on the other hand, $(1-p)^N$ may stay close to 1, in which case Q becomes very small. Hence, Q can take any number between 0 and 1 if N is properly chosen between those extreme cases. This case needs a special care in computing Q.

Case 2: $p > 1 - \epsilon$

In this case, Q is even closer to 1. From Eq. 1, it is trivial that $1-Q \leq 1-p$, hence $Q \geq p$. As a result, when $p > 1-\epsilon$, it is guaranteed that $Q > 1-\epsilon$. In that case, the only way to express Q on a computer system is to make it equal to 1. For that reason, there is no problem in computing Q in this case, even though a number of significant digits may be lost in computation of $(1-p)$ by computational rounding. Note that, however, setting $Q = 1$ for $p > 1-\epsilon$ might cause a problem when p is very close to the boundary of the inequality, especially for $N = 1$.

Case 3: $(1-p)^N < \epsilon$

In this case, Q is very close to 1. This case makes $Q > 1-\epsilon$, hence the discussion for the case for $p > 1-\epsilon$ applies to this case.

Case 4: $(1-p)^N > 1 - \epsilon$

In this case, Q is very small, as $Q < \epsilon$. A regular computation of $1-(1-p)^N$ would result in zero, because $(1-p)^N$ would be expressed as exactly 1 in a computer system. This case needs a special care in computing Q.

3 Computation strategy

This section describes in detail how to numerically compute Eq. 1, overcoming the computation difficulties mentioned in Section 2. With a simple manipulation of Eq. 1, it is trivial that Q can be written as

$$Q = 1 - e^{-x} \tag{2}$$

where

$$x \equiv -N \ln(1-p). \tag{3}$$

The strategy here is to compute x from p first, then compute Q from the resultant x. Each of the computationally difficult cases in Section 2 now belongs to one of the two steps: cases 1 and 2 should be taken care of in the first step (i.e., computation of x from p) and cases 3 and 4 the second (i.e., computation of Q from x).

The strategy does not work efficiently for a few cases. Below those special cases are described first, then the detailed descriptions of the computing strategy introduced above follow.

3.1 Special cases

For $N = 1$, $Q = p$. For $N = 0$, $Q = 0$. These cases can be handled as described in Sections 3.2 and 3.3. However, compared with simply letting $Q = p$ and $Q = 0$, respectively, it is clearly inefficient and less precise to compute Q through Eqs. 2 and 3. So, let $Q = p$ for $N = 1$ and let $Q = 0$ for $N = 0$.

For $p = 1$, $Q = 1$ if $N \geq 1$. This case requires a computation of $\ln(0)$ in Eq. 3, hence the strategy in Section 3 does not work. Instead, Q should be set to 1 for this case. Considering the discussion for Case 2 in Section 3, other cases also need the same handling if computation of $(1-p)$ gives zero due to rounding errors. Precisely speaking, let $Q = 1$ if boolean expression "$1 - p = 0$" gives true in a computer system.

For $p = 0$, $Q = 0$ if $N \geq 1$. This case causes zero-division in evaluation of a termination condition for summation of a power series about p, which is introduced later (Section 3.2). So, let $Q = 0$ if boolean expression "$p = 0$" gives true in a computer system. Note that the first term of the power series (Eq. 4) is p itself, so it is sufficient to check whether p is computationally equivalent to zero, in order to avoid a zero-division error.

3.2 Computation of x from p

For any p in range $0 < p < 1$, Eq. 3 can be expanded into a power series as

$$x = N \sum_{n=1}^{\infty} q_n(p) \tag{4}$$

where

$$q_n(p) \equiv \frac{p^n}{n} \tag{5}$$

It is trivial that $q_n(p) > 0$ for any $n \geq 1$ and $q_n(p)$ decreases as n decreases. Consider performing the above summation literally, and terminating the computation when the next term to add becomes too small to add. In other words, let's re-write x as

$$x = N \cdot S_m(p) + N \cdot R_{m+1}(p) \tag{6}$$

where

$$S_m(p) \equiv \sum_{n=1}^{m} q_n(p) \tag{7}$$

$$R_m(p) \equiv \sum_{n=m}^{\infty} q_n(p) \tag{8}$$

and compute $S_m(p)$ for the smallest m that satisfies

$$\frac{q_{m+1}(p)}{S_m(p)} < \epsilon \tag{9}$$

For such m, the residual $R_{m+1}(p)$ is evaluated as

$$R_{m+1}(p) < \sum_{n=m+1}^{\infty} \frac{p^n}{m+1} \tag{10}$$

$$= \frac{p^{m+1}}{m+1} \sum_{n=0}^{\infty} p^n \tag{11}$$

$$= \frac{q_{m+1}(p)}{1-p} \tag{12}$$

because

$$q_n(p) < \frac{p^n}{m+1} \tag{13}$$

for $n > m+1$. Combining Eqs. 9 and 12 gives

$$\frac{R_{m+1}(p)}{S_m(p)} < \frac{\epsilon}{1-p} \tag{14}$$

and the fractional residual is then evaluated as

$$\frac{N \cdot R_{m+1}(p)}{x} = \frac{R_{m+1}(p)}{S_m(p) + R_{m+1}(p)} \tag{15}$$

$$< \frac{\epsilon}{1 - p + \epsilon} \tag{16}$$

Assuming $\epsilon \ll 1$, the estimator of the fractional residual in Eq. 16 can be approximately computed as follows.

$$\frac{\epsilon}{1 - p + \epsilon} \approx \begin{cases} 2\epsilon & \text{for} \quad p = 0.5 \\ 1.1\epsilon & \text{for} \quad p = 0.1 \\ 1.01\epsilon & \text{for} \quad p = 0.01 \end{cases} \tag{17}$$

Now it is trivial that the fractional residual becomes more significant for a larger p. Therefore, direct computation of x as in Eq. 3 works better for such cases. On the other hand, loss of significant digits in direct computation of $(1 - p)$ becomes more serious for a smaller p, starting at $p \approx 0.1$. In other words, The above estimate shows that the fractional residual dominates loss of precision in computation of x for a large p, that direct computation of $(1 - p)$ dominates it for a small p, and that the boundary between those two scenarios lies around $p = 0.1$.

3.3 Computation of Q from x

For any $x \geq 0$, Eq. 2 can be expanded into a power series as

$$Q = \sum_{n=0}^{\infty} q_n(x) \tag{18}$$

where

$$q_n(x) \equiv \frac{x^{2n+1}}{(2n+1)!} \left(1 - \frac{x}{2n+2}\right) \tag{19}$$

For a certain value of x, however, computing the above summation in an iterative manner does not work. For example, the behavior of $q_n(x)$ for $x \gg 1$ changes in the middle of the sum, where $x \approx 2n + 1$. In fact, for $x \geq 1$ ($Q > 0.63$), direct computation of Eq. 2 gives the best estimate of Q, even though it causes loss of significant digits when e^{-x} is subtracted from 1. After all, a resultant Q cannot be expressed any better than the result

of direct computation of the subtraction. In other words, loss of significant digits in the subtraction is unavoidable in principle.

Another case worth noting is $x = 0$ ($Q = 0$), where all $q_n(x)$ becomes zero. This case may not seem important because x becomes zero only if $p = 0$ or $N = 0$, both of which are handled as a special case (see Section 3.1 for details). However, this case should be checked in an actual computation because x may become computationally equivalent to zero for some other reasons, such as rounding errors and floating-point underflow.

For $0 < x < 1$, consider performing the above summation literally, and terminating the computation when the next term to add becomes too small to add. In other words, let's re-write Q as

$$Q = S_m(x) + R_{m+1}(x) \tag{20}$$

where

$$S_m(x) \equiv \sum_{n=0}^{m} q_n(x) \tag{21}$$

$$R_m(x) \equiv \sum_{n=m}^{\infty} q_n(x) \tag{22}$$

and compute $S_m(x)$ for the smallest m that satisfies

$$\frac{q_{m+1}(x)}{S_m(x)} < \epsilon \tag{23}$$

For such m, the residual $R_{m+1}(x)$ is evaluated as

$$R_{m+1}(x) < \sum_{n=m+1}^{\infty} \frac{x^{2n+1}}{(2m+3)!}\left(1 - \frac{x}{2m+4}\right) \tag{24}$$

$$= \frac{x^{2m+3}}{(2m+3)!}\left(1 - \frac{x}{2m+4}\right)\sum_{n=0}^{\infty} x^{2n} \tag{25}$$

$$= \frac{q_{m+1}(x)}{1 - x^2} \tag{26}$$

because

$$q_n(x) < \frac{x^{2n+1}}{(2m+3)!}\left(1 - \frac{x}{2m+4}\right) \tag{27}$$

for $n > m+1$ (see Section 5.1 for a proof of the inequality). Combining Eqs. 23 and 26 gives

$$\frac{R_{m+1}(x)}{S_m(x)} < \frac{\epsilon}{1-x^2} \tag{28}$$

and the fractional residual is then evaluated as

$$\frac{R_{m+1}(x)}{Q} < \frac{\epsilon}{1-x^2+\epsilon} \tag{29}$$

Assuming $\epsilon \ll 1$, the estimator of the fractional residual in Eq. 29 can be approximately computed as follows.

$$\frac{\epsilon}{1-x^2+\epsilon} \approx \begin{cases} 1.3\epsilon & \text{for} \quad x = 0.5 \\ 1.01\epsilon & \text{for} \quad x = 0.1 \\ 1.0001\epsilon & \text{for} \quad x = 0.01 \end{cases} \tag{30}$$

Again, the fractional residual becomes more significant for a larger x. Therefore, direct computation of Q as in Eq. 2 works better for such cases. On the other hand, loss of significant digits in subtraction of e^{-x} from 1 becomes more serious for a smaller x, starting at $x \approx 0.1$ where $e^{-x} \approx 0.9$. In other words, The above estimate shows that the fractional residual dominates loss of precision in computation of Q for a large x, that subtraction of e^{-x} from 1 dominates it for a small x, and that the boundary between those two scenarios lies around $x = 0.1$.

4 Summary

This section summarizes how to compute

$$Q = 1-(1-p)^N \tag{31}$$

$$= 1-e^{-x} \tag{32}$$

where

$$x \equiv -N\ln(1-p) \tag{33}$$

in a computer system in which the difference between 1.0 and the smallest representable value greater than 1.0 is ϵ.

1. Handle special cases, if any one of the conditions below meets. Otherwise, go to the next step. Note that cases for $N = 1$ can be handled for any p by simply letting $Q = p$, cases for $p = 0$ or $p = 1$ need to be handled only for $N \geq 2$.

(a) Set $Q = 0$ if $N = 0$.

(b) Set $Q = p$ if $N = 1$.

(c) Set $Q = 1$ if $N \geq 2$ and $p = 1$.

(d) Set $Q = 0$ if $N \geq 2$ and $p = 0$.

2. Compute x.

   (a) If $p \geq 0.1$, compute $-N \ln(1-p)$ in a regular manner and set the result to x.

   (b) Otherwise, compute $S_m(p)$ iteratively until $r_{m+1}(p) < \epsilon$, where

$$S_m(p) \equiv \sum_{n=1}^{m} q_n(p) \tag{34}$$

$$q_n(p) \equiv \frac{p^n}{n} \tag{35}$$

$$r_m(p) \equiv \frac{q_m(p)}{S_{m-1}(p)} \tag{36}$$

   and set $x = N \cdot S_m(p)$.

3. Compute Q.

   (a) If $x \geq 0.1$, compute $1 - e^{-x}$ in a regular manner and set the result to Q.

   (b) Otherwise, compute $S_m(p)$ iteratively until $r_{m+1}(p) < \epsilon$, where

$$S_m(x) \equiv \sum_{n=0}^{m} q_n(x) \tag{37}$$

$$q_n(x) \equiv \frac{x^{2n+1}}{(2n+1)!}\left(1 - \frac{x}{2n+2}\right) \tag{38}$$

$$r_m(x) \equiv \frac{q_m(p)}{S_{m-1}(p)} \tag{39}$$

   and set $Q = S_m(x)$.

Note that the computing methods with series expansions above (i.e., computation of $S_m(p)$ for x and that of $S_m(x)$ for Q) give a fractional error (difference between the true value and a computed value divided by the true value) of an order of ϵ.

5 Useful inequalities

This section lists the inequalities used in this note with their complete proofs.

5.1 Inequality A

For $n > m \geq 0$ and any $0 < x < 1$, the following inequality holds.

$$q_n(x) < \frac{x^{2n+1}}{(2m+1)!}\left(1 - \frac{x}{2m+2}\right) \tag{40}$$

where

$$q_n(x) \equiv \frac{x^{2n+1}}{(2n+1)!}\left(1 - \frac{x}{2n+2}\right) \tag{41}$$

Proof

For $m \geq 0$, define functions

$$c_m(x) \equiv \frac{1}{(2m+1)!}\left(1 - \frac{x}{2m+2}\right) \tag{42}$$

$$f_m(x) \equiv \frac{c_{m+1}(x)}{c_m(x)} \tag{43}$$

Then, it is trivial that $c_m(x) > 0$ for any $m \geq 0$ and any x that satisfies $0 < x < 1$. And,

$$f_m(x) = \frac{(2m+1)!}{(2m+3)!} \cdot \frac{1 - \frac{x}{2m+4}}{1 - \frac{x}{2m+2}} \tag{44}$$

$$= \frac{1}{(2m+3)(2m+2)} \cdot \frac{2m+2}{2m+4} \cdot \frac{2m+4-x}{2m+2-x} \tag{45}$$

$$= \frac{1}{(2m+4)(2m+3)} \cdot \frac{2m+4-x}{2m+2-x} \tag{46}$$

$$< \frac{1}{(2m+4)(2m+3)} \cdot \frac{2m+4-0}{2m+2-1} \tag{47}$$

$$= \frac{1}{(2m+2)(2m+3)} \tag{48}$$

$$< 1 \tag{49}$$

Therefore,

$$c_{m+1}(x) < c_m(x) \tag{50}$$

for any $m \geq 0$, hence,

$$c_n(x) < c_m(x) \tag{51}$$

for any $n > m \geq 0$. Since $x^{2n+1} > 0$, it is now trivial that

$$c_n(x)x^{2n+1} < c_m(x)x^{2n+1} \tag{52}$$

which is the inequality to prove.
(End of proof)