Computer Codes for Special Case of Counting: Exact Decision Levels, Errors of the First Kind, and Probability Density Function

Bill Potter and Jodi Strzelczyk

ABSTRACT

In the past exact computations utilizing modified Bessel functions of integral order have been discussed for decision levels in paired counting. This paper transforms the net count to an integer and assumes that the blank count is Poisson distributed with known expected value. Utilizing the Poisson probability density function, a function in C++ is written to compute the exact probability density function for the transformed net count when it is less than zero, equal to zero and greater than zero. The validity of the computation is then checked by summing probabilities over a wide range of transformed net counts and comparing to 1.0 and by computing the expected transformed net count and comparing to 0.0. A decision level is determined by summing the right tail of the probability density function and inverting from a transformed net count to a net count. Codes were written in both double precision arithmetic and long double precision arithmetic. The double precision code is found to be adequate for most applications.

Trademarks:

Borland C++ Builder 5, Borland Software Corporation, 20450 Stevens Creek Blvd., Suite 800, Cupertino, CA 95014.

Microsoft Visual C++ 6.0(Standard Edition), Microsoft Corporation, One Microsoft Way, Redmond, WA 98052-6399.

INTRODUCTION

Two C++ computer codes were written to compute decision levels, errors of the first kind, and the probability density function for the net count when there is no activity in the sample. The first code uses double precision arithmetic and the second one uses long double precision arithmetic. Assume the blank is counted N times longer than the sample with N being an integer in the interval [1,20]. Also assume the blank count in the sample counting time, which is the same as the contribution of the blank count to the gross sample count over the sample counting time, is Poisson distributed with a known expected value B. Then the net count S is given by

$$S = \{\text{gross count} - (\text{blank count})/N\}.$$

In the previous equation S, gross count, and blank count are random variables. For decision levels there is no activity is the sample; so S is given by the following equation relating the random variables S, blank count in sample counting time, and the blank count:

$$S = \{\text{blank count in sample counting time} - (\text{blank count})/\text{N}\}.$$

Because the sum of N identical Poisson distributed random variables also follows the Poisson distribution, the expected value of the blank count in the sample counting time is (1/N)th of the expected value for the blank count..[1] Take s to be an arbitrary value assumed by S. Then from the previous equation:

Probability$\{S=s\}$

Probability {blank count in sample counting time - (blank count)/$N = s$} _

Probability {N (blank count in sample counting time) - (blank count)= Ns}.

The quantity Ns is an integer. After transforming the net count to an integer it is reasonably straightforward to derive the probability density function for the transformed net count for the cases when the transformed net count is greater than zero, zero, and less than zero. The validity of the computations can be checked by verifying that the sum of probabilities, over all integers, Ns, is close to 1.0. Another check on the computation is to verify that the expected value of the transformed net count is close to zero.

DISCUSSION

The net count S is of the form

$S = G$ --- (blank count)/N (Eq. 1)

with N a positive integer and G the gross count. The quantity B is the *expected* blank count in the sample count time. To determine the decision level, it is necessary to know B or have a good estimate for B. Throughout this paper, the convention was utilized:

activity is said to have been detected if the value of the net count is greater than the decision level.

Two applications of the previous formula are as follows:

1. Count the blank N times longer than the sample and use this value to estimate B. Thus B is estimated to be (blank count)/N.) Then for a group of samples, only gross counts are made with a counting time of (1/N)th the time of the total blank count. There are problems if the background changes or the instrument is unstable.

2. Count the sample for a time Ts and the blank for a time Tb with $Tb/Ts = N$.

Use past, acquired experience to estimate B.

For the purpose of determining decision levels it is assumed that there is no activity in the sample. Then Equation 1 becomes:

NS = N (blank count in sample counting time) – (blank count) (Eq. 2)

Take the 3 quantities in Equation 2 to be random variables (not N); then (blank count in sample counting time) is Poisson distributed with expected value B and the (blank count) is Poisson distributed with expected value (NB). It is important to note that the (blank count in sample counting time) and the (total blank count) are independent random variables. Decision levels are determined for arbitrary errors of the first kind in the interval [0.001, 0.5] by summing the right tail of the probability density function and inverting from a transformed net count to a net count.

For discrete random variables it is generally not possible to determine decision levels yielding errors of the first kind equal to a desired value.[2,3] Currie[3] chose the error of the first kind to be equal to or less than the desired value. In many situations it is possible to arrive at an average error of the first kind equal to the desired error of the first kind by averaging an error of the first kind greater than the desired error of the first kind and an error of the first kind less than the desired error of the first kind. A random number would be drawn to arrive at a decision level.

Because the codes are capable of computing errors of the first kind when the expected blank count in the sample count time is known, the codes are capable of comparing approximate formulas for decision levels. From Equation 2:

S = (blank count in sample counting time) - (total blank count)/N. (3)

Then the (blank count in sample counting time) is Poisson distributed with expected value B and the (total blank count) is Poisson distributed with expected value (NB). Taking the variance of both sides of Equation 3 yields:

Variance(S) = $B + (1/N)B$. *(4)*

Equation 4 suggests approximate decision levels $CLapx$ of the form

$$CLapx = correction\ \ factor + zalph\sqrt{B(1+1/N)}$$

If (1-d) is the desired error of the first kind then $zalph$ is the confidence coefficient corresponding to a (100·d) percent one sided confidence interval for a Gaussian probability distribution. In particular, if the desired error of the first kind is 0.05, then $zalph$ = 1.645. So that the code for the exact decision levels can be utilized to determine the approximate errors of the first kind, *CLapx* is transformed into a multiple of *(1/N)* as follows:

$$CLapx = (1/N)\ \mathrm{floor}(N\ CLapx). \qquad (5)$$

Paired counting has been discussed in the past; in this case $N = 1$. Modified Bessel functions of integral order were used.[41] However, when *N* does not equal 1, Bessel functions are not useful. Advantages of the Bessel function approach are simplicity in writing numerical codes and the speed of execution of the codes. Both codes developed here agree with results from the Bessel function based codes, which are known to be very accurate.

Long double precision arithmetic is necessary for larger *B* and larger *N* to avoid numerical overflow/underflow. Borland C++ Builder 5® supports both double precision arithmetic and long double precision arithmetic. Microsoft Visual C++ 6.0® readily supports only double precision arithmetic; it was used for the double precision code. The double precision code is adequate for most applications. However both codes can be utilized to check on the validity of the codes and the accuracy of the double precision code. Both codes sum the probabilities over all transformed net counts for the maximum value of *B*, *BM*, to be considered when there is no activity present; this is a check on the codes and the value should be close to 1.0. The double precision code also computes the expected value of the transformed net count when there is no activity present; this should be close to zero; this option was removed from the long double precision code because of the length of time needed to execute this operation.

EXAMPLES

Below, three examples are presented; two use the double precision code and the third one uses the long double precision code.

Example 1: Double precision code .

Ratio of blank count time to sample count time = $N = 10$ *alpha* = 0.05

correction factor = 0.3

BM (max *B*) = 10, Sum of probabilities at *B = BM = 1*

Expected value of distribution at *B* = *BM* = 3.396278546e-015

B = Expected blank count in sample count time

CL = exact decision level

Err1 = exact error of the first kind

CLapx = approximate decision level

Err1apx = exact error of first kind for approximate decision level

Execution time: about 6 minutes

Table 1. *Example 1 Results*

B	*CL*	*Err1*	*CLapx*	*Err1apx*
0	0	0	0.3	0
1	2	0.04701236146	2	0.04701236146
2	2.7	0.04603303883	2.7	0.04603303883
3	3.2	0.04908622374	3.2	0.04908622374
4	3.7	0.04770315329	3.7	0.04770315329
5	4.1	0.04821942935	4.1	0.04821942935
6	4.5	0.04734201909	4.5	0.04734201909
7	4.8	0.04871667852	4.8	0.04871667852
8	5.1	0.0492527525	5.1	0.0492527525
9	5.4	0.04919211236	5.4	0.04919211236
10	5.7	0.04870496498	5.7	0.04870496498

Example 2: Double precision code

Ratio of background count time to sample count time = *N* = 20 *alpha* = 0.05

correction factor = 0.3

BM (max *B)* = *1*, Sum of probabilities at *B* = *BM* = *1*

Expected value of distribution at *B* = *BM*= -3.916774868e-017

B = Expected blank count in sample count time

CL = exact decision level

Err1 = exact error of the first kind

CLapx - approximate decision level

Err1apx = exact error of first kind for approximate decision level

Execution time: about 12 minutes

Example 3: Long double precision code

Ratio of background count time to sample count time =N= 5

alpha= 0.05

correction factor = 0.35

*BM (*max *B) = 500,* Sum of probabilities at *B = BM= 1*

B = Expected blank count in sample count time

CL = exact decision level

Err1 = exact error of the first kind

CLapx = approximate decision level

Err1 apx = exact error of first kind for approximate decision level

Execution time: about *95* minutes

Table 2. *Example 2 Results*

B	*CL*	*Err1*	*CLapx*	*Err1apx*
0.0	0.0	0	0.3	0.0
0.1	0.9	0.04141576864	0.8	0.08223457805
0.2	0.9	0.03251867312	1.05	0.01752309546
0.3	0.9	0.04079254012	1.2	0.0369193414
0.4	1.5	0.04635574034	1.35	0.05813077236
0.5	1.5	0.04910622627	1.45	0.0585916125
0.6	1.5	0.04706008749	1.6	0.03195598426
0.7	1.5	0.04744762835	1.7	0.03474060761
0.8	1.6	0.04870301794	1.8	0.04602935984
0.9	1.95	0.04955496793	1.85	0.05569909203
1.0	2.00	0.04782033813	1.95	0.05326754542

Table 3. *Example 3 Results*

B	*CL*	*Err1*	*CLapx*	*Err1apx*
0	*0.0*	*0*	*0.2*	*0*
50	*13.0*	*0.0483161358*	*13.0*	*0.0483161358*
100	*18.2*	*0.04948705776*	*18.2*	*0.04948705776*
150	*22.2*	*0.04994907701*	*22.4*	*0.04847912886*
200	*25.8*	*0.04876537932*	*25.8*	*0.04876537932*
250	*28.8*	*0.04893667391*	*28.8*	*0.04893667391*
300	*31.4*	*0.04965391976*	*31.4*	*0.04965391976*
350	*34*	*0.04919406674*	*34*	*0.04919406674*
400	*36.2*	*0.04982914307*	*36.2*	*0.04982914307*
450	*38.4*	*0.0497782957*	*38.4*	*0.0497782957*
500	*40.6*	*0.04924620081*	*40.6*	*0.04924620081*

CONCLUSION

The long double precision code takes much longer to execute than the double precision code. When N = 1 the double precision code presented in this paper takes longer to execute than the old code for paired counting that uses modified Bessel functions of integral order. From numerical experimentation, some limits for applicability of the codes are as follows:

- When $N < 6$ the double precision code will yield results for $B \leq 100.0$;
- When $N = 10$ the double precision code will yield results for $B \leq 66.0$;
- When $N = 20$ the double precision code will yield results for $B < 22.0$;
- When $N < 10$ the long double precision code will yield results for $B < 1000.0$;
- When $N = 20$ the long double precision code will yield results for $B < 250.0$.

Using the codes, one can determine simple, approximate formulae for decision levels that are suitable for a limited range for *B*.

Throughout this paper it has been assumed that the expected blank count is known or that a good estimate is available. The quality of the estimate can be evaluated by looking at a confidence interval for the expected blank count.

The double precision code is appropriate for many applications. The long double precision code is necessary to prevent numerical overflow/underflow that may arise from larger expected blank counts and larger values for N. The long double precision code can be used to validate the accuracy the double precision code.

Acknowledgments

Interactions with Ken Skrable and Clayton French are acknowledged and appreciated.

REFERENCES

1. Lindgren, B.W., *Statistical theory, The* Macmillan Company, New York, 1962, p. 313.
2. Brodsky, A., *Health Phys.* 1992, **63**, 198.
3. Currie, L.A., *Lower Limit of Detection: Definition and Elaboration of a Proposed Position for Radiological Effluent and Environmental Measurements,* Report No. NUREGICR-4007, Nuclear Regulatory Commission, Washington, DC, 1984.
4. Potter, W. E., (2000), *Radiat. Protect. Manag.* **17**, 22.

The Authors

Bill Potter has a Bachelor's degree in Engineering Science, a Master's degree in Engineering Science-Nuclear Engineering, and a Master's degree in Mathematics from the University of California at Berkeley. He has additional graduate work in nuclear engineering, statistics, medical physics, industrial engineering and operations research, and mechanical engineering together with a wide range of interests. He has been a guest with the Health Physics Department at Lawrence Berkeley National Laboratory, and he holds membership in both the Health Physics Society and the Central Rocky Mountain Chapter of the Health Physics Society.

1008 Tenth Street
PMB # 492
Sacramento, CA 95814-3502
E-Mail: PsPr189729@aol.com

Jadwiga (Jodi) Strzelczyk is Associate Professor of Radiology, Radiological Sciences Div. at the University of Colorado (CU) at Denver, Health Sciences Center and serves as the Radiation Safety Officer for the CU Hospital since 1995. She also serves on the Colorado Radiation Advisory Committee since 2000. Her occupational experience includes ten-year tenure as the University Hospital RSO at State University of New York (SUNY), Stony Brook and fifteen-year employment at two industrial firms on Long Island, NY in the capacity of program manager and the RSO. Jodi has a M.Sc. degree in High Energies Nuclear Physics from Poland, a M.S. in Mathematics and Statistics (Industrial Management Track) from SUNY, Stony Brook, NY and a recent (2001) Ph.D. in Health and Behavioral Sciences from the CU.

University of Colorado Health Science Center
School of Medicine
Denver, CO 80262,
E-Mail: Jodi.Strzelczyk@uchsc.edu