Behavior Research Methods & Instrumentation
1977, Vol. 9 (6), 533-537

Procedures for computing d′ and β

WILLIAM A. AHROON, JR.
Parmly Hearing Institute, Loyola University of Chicago
6525 North Sheridan Road, Chicago, Illinois 60626

and

RICHARD E. PASTORE
Psychoacoustics Laboratory, State University of New York
Binghamton, New York 13901

Although originally employed in studies of sensory processes, signal detection theory (TSD) has been adapted for a wide range of studies in psychological research (see Pastore & Scheirer, 1974).[1] The general availability of computers (mega, mini, or micro) to psychological research laboratories suggests a need for a general-purpose routine for computing d′ and β, the major statistics of the equal-variance Gaussian model of signal detection theory. This report presents several higher level language routines that may be adapted for use in a number of large or small computer systems.

The Gaussian model assumes that the observers establish a criterion value of the ratios of the likelihood that the given stimulus event arose from a "signal-plus-noise" vs. a "noise-alone" condition (i.e., from two classes of experimenter-defined events). The events arising from the signal-plus-noise and noise-alone conditions are both assumed to be normally distributed with equal variance, but differ in the placement of their means. The d′ statistic represents a measure of the normalized distance between the means of two hypothesized Guassian distributions. It is a measure of the sensitivity of the observer (subject) to differences between the alternative classes

Table 1
FORTRAN Function to Compute Index of Sensitivity d′

```
C                    FUNCTION "DPRIME"
C
C     INPUT PROBABILITIES OF HIT AND FALSE ALARM
C     DPRIME IS RETURNED
      FUNCTION DPRIME (PH, PFA)
      DIMENSION Z(51)
      DATA Z/ 0.00000,0.02507,0.05015,0.07527,0.10043,0.12566,0.15097,
     1 0.17637,0.20189,0.22754,0.25335,0.27932,0.30548,0.33185,0.35846,
     2 0.38532,0.41246,0.43991,0.46770,0.49585,0.52440,0.55338,0.58284,
     3 0.61281,0.64335,0.67449,0.70630,0.73885,0.77219,0.80642,0.84162,
     4 0.87790,0.91537,0.95417,0.99446,1.03643,1.08032,1.12639,1.17499,
     5 1.22653,1.28155,1.34076,1.40507,1.47579,1.55477,1.64485,1.75069,
     6 1.88079,2.05375,2.32635,2.57572/
C
C     ROUND PROBABILITIES TO NEAREST .01%, MULTIPLY BY 100, AND TRUNCATE
C
      IH = (PH + 0.005) * 100.
      IH = IH - 50
      IFA = (PFA + 0.005) * 100.
      IFA = IFA - 50
C
C     COMPUTE HIT Z-SCORE
C
      IF (IH) 10, 20, 30
   10 ZHIT = Z (1-IH) * (-1.0)
      GO TO 40
   20 ZHIT = 0.00
      GO TO 40
   30 ZHIT = Z (1 + IH)
C
C     COMPUTE FALSE ALARM Z-SCORE
C
   40 IF (IFA) 50, 60, 70
   50 ZFALSE = Z (1 - IFA)
      GO TO 80
   60 ZFALSE = 0.00
      GO TO 80
   70 ZFALSE = Z (1 + IFA) * (-1.0)
C
C     COMPUTE DPRIME AND RETURN
C
   80 DPRIME = ZHIT + ZFALSE
      RETURN
      END
```

of signal events and is assumed to be independent of the observer's response bias, as defined by the criterion β. The d′ statistic is computed by subtracting the (signed) z-score of the criterion likelihood ratio associated with the signal-plus-noise distribution from that of the noise-alone distribution. The statistic representing the placement of the criterion likelihood ratio, β, is considered to be representative of an observer's response bias. It is computed as the ratio of the ordinates of the unit-normal curve at the criterion likelihood ratio for the two distributions. The following FORTRAN IV functions are designed to compute values of d′ and β using these definitions. (Care should be taken throughout these programs to account for individual machine characteristics; e.g., will the computer round or will it truncate when converting from floating-point to fixed-point representation or when printing results.)

Function DPRIME. The DPRIME function presented in Table 1 inputs the probability of a hit (correctly detected signal) and the probability of a false alarm (erroneous report of signal detection) and returns the value of d′. The inputs to the function are multiplied by 100, fixed (to integer representation), and used as indices to select two z-scores from the table of z-scores in the DATA statement.[2] If the probability of a hit is less than .50 or the probability of a false alarm is greater than .50, the z-score associated with that probability is made negative. The z-score associated with the probability of a hit is added to the z-score associated with the probability of a false alarm, and a value of d′ is returned. Hit and false alarm probabilities are not altered.

Table 2 presents a listing of a routine (Program TABLE) which will print an entire table of d′ values for P(Hit) and P(False Alarm) from .01 to 1.00. These values correspond to those of Elliott (Note 1; also presented in Swets, 1964, Appendix 1). The input of Program TABLE, which uses one more degree of precision than did Elliot (Note 1), suggests that Elliot truncated the d′ values (at two places to the right of the decimal) instead of rounding to the nearest hundredth.

Function BETA. The value of the statistic β reflects the placement of the observer's response criterion for a single set of trials. The statistic is computed as a ratio of the heights of the unit-normal curves for the signal-plus-noise and noise-alone distributions. Thus, with P(Hit) = .94 and P(False Alarm) = .10, β = .11912/.17550 = .67895. A value of β equal to 1.00 is indicative of no response bias or response preference. Since β is a ratio of two distributed parameters, the statistic "log β" (but not β) has a theoretical distribution which is symmetric around $\beta = 1.00$ ($\log \beta = 0$).

Function BETA, presented in Table 3, like Function DPRIME, inputs P(Hit) and P(False Alarm), converts these probabilities to integers, and indexes the two Y-ordinate values. The ratio between the two ordinates is the value of β returned. The input probabilities remain unchanged.

Subroutine TSDDPB. Egan (1975) has presented a relationship between d′ and P(False Alarm) which can be used to compute β. Hence, the addition of one statement to Function DPRIME and a change in the calling sequence to call a subroutine (vs. a function) allows the computation of both d′ and β in a single subroutine. The calling sequence would then be

```
CALL TSDDPB (PH, PFA, DPRIME, BETA)
```

in which the hit and false alarm probabilities are input and both statistics are returned.

To change Function DPRIME to Subroutine TSDDPB (*T*heory of *S*ignal *D*etection: *D P*rime and *B*eta), change

```
FUNCTION DPRIME (PH, PFA)
```

to

```
SUBROUTINE TSDDPB (PH, PFA, DPRIME, BETA)
```

Add the following statement just prior to the RETURN statement:

```
BETA = EXP((ZFALSE * DPRIME)
       – (DPRIME * DPRIME/2.))
```

Table 2
FORTRAN Program to Compute Table of d′ Values

```
C      PROGRAM TABLE (TEST FOR FUNCTION DPRIME)
C
       DIMENSION D(10), PLAB(10)
       PFA = 0.0
       PH = 0.0
    10 PLAB(1) = PFA + 0.01
       DO 15 K = 2, 10
    15 PLAB(K) = PLAB(K-1) + 0.01
    16 WRITE(7,102) (PLAB(K), K = 1, 10)
       DO 30 J = 1, 50
       PH = PH + 0.01
       PF = PFA
       DO 20 I = 1, 10
       PF = PF + 0.01
    20 D(I) = DPRIME (PH, PF)
    30 WRITE(7,100) PH, (D(I), I = 1, 10)
       IF (PH - 0.90) 16, 16, 17
    17 PH = 0.00
       PFA = PFA + 0.10
       IF (PFA - 0.99) 99, 103, 103
    99 WRITE(7,101)
       GO TO 10
   100 FORMAT (8X, F4.2, 10F7.3)
   101 FORMAT (////////////)
   102 FORMAT (41X, 'PN(A)'/11X, 10F7.2/8X, 'PSN(A)'/)
   103 WRITE(7,101)
       WRITE(7,101)
       CALL EXIT
       END
```

Table 3
FORTRAN Function to Compute Response Criterion

```
C                     FUNCTION "BETA"
C
C     INPUT PROBABILITIES OF HIT AND FALSE ALARM.
C     BETA IS RETURNED.
C
      FUNCTION BETA (PH, PFA)
      DIMENSION Y(51)
C
      DATA Y/ 0.39894,0.39882,0.39844,
     1 0.39781,0.39694,0.39580,0.39442,0.39279,0.39089,0.38875,0.38634,
     2 0.38368,0.38076,0.37757,0.37412,0.37040,0.36641,0.36215,0.35761,
     3 0.35279,0.34769,0.34230,0.33662,0.33065,0.32437,0.31778,0.31087,
     4 0.30365,0.29609,0.28820,0.27996,0.27137,0.26240,0.25305,0.24331,
     5 0.23316,0.22258,0.21155,0.20004,0.18804,0.17550,0.16239,0.14867,
     6 0.13427,0.11912,0.10314,0.08617,0.06804,0.04842,0.02665,0.01447/
C
C     ROUND PROBABILITIES TO NEAREST .01%, MULTIPLY BY 100, AND TRUNCATE
C
      IH = (PH + 0.005) * 100.
      IH = IH - 50
      IFA = (PFA + 0.005) * 100.
      IFA = IFA - 50
C
C     COMPUTE HIT Y-ORDINATE
C
      IF (IH) 10, 20, 30
   10 BETAH = Y(1 - IH)
      GO TO 40
   20 BETAH = 0.39894
      GO TO 40
   30 BETAH = Y(1 + IH)
C
C     COMPUTE FALSE ALARM Y-ORDINATE
C
   40 IF (IFA) 50, 60, 70
   50 BETAFA = Y(1 - IFA)
      GO TO 80
   60 BETAFA = 0.39894
      GO TO 80
   70 BETAFA = Y(1 + IFA)
C
C     COMPUTE BETA AND RETURN
C
   80 BETA = BETAH / BETAFA
      RETURN
      END
```

Subroutine TSDDPB requires less memory than using Function DPRIME and Function BETA but may take more time to run due to use of the exponential function. Since many researchers utilize log β instead of β in descriptive and inferential statistics (see above), the log of β may be computed instead of β by:

```
LBETA = (ZFALSE * DPRIME)
        - (DPRIME * DPRIME) / 2.
BETA = LBETA
```

in place of the statement utilizing the exponential function.

FORTRAN II. Since many FORTRAN compilers do not support DATA statements, the previous functions and subroutine must be modified to run with some machines. If DATA statements are not valid, a vector of z-scores or Y-ordinates must be set up in another sub-

Table 4
Abbreviated Listings for Subroutines ZSCORE and YSCORE*

```
SUBROUTINE ZSCORE (Z)      SUBROUTINE YSCORE (Y)
DIMENSION Z(51)            DIMENSION Y(51)
Z(01) = 0.00000            Y(01) = 0.39894
Z(02) = 0.02507            Y(02) = 0.39882
Z(03) = 0.05015            Y(03) = 0.39884
   .                          .
   .                          .
   .                          .
Z(50) = 2.32635            Y(50) = 0.02665
Z(51) = 2.57572            Y(51) = 0.01447
RETURN                     RETURN
END                        END
```

**Called by CALL ZSCORE (Z) and CALL YSCORE (Y).*

routine. To optimize time, one should call the ZSCORE or YSCORE subroutines early in the analysis program, since it need be called only once. The values in the tables are unaffected by the functions and subroutine. Further, it is necessary to pass the values of Z or Y to the function/subroutine when called. Thus, calling sequences such as

```
D = DPRIME (PH, PFA, Z)
B = BETA (PH, PFA, Y)
CALL TSDDPB (PH, PFA, Z, DPRIME, BETA)
```

should be used. Note that the initial statement in the function or subroutine also should contain the variable Z or Y. For example, the statement

```
SUBROUTINE TSDDPB (PH, PFA, Z, DPRIME, BETA)
```

would be the initial statement in the FORTRAN II subroutine TSDDPB.

Table 4 presents abbreviated routines to input the vectors of z-scores and Y-ordinates. Obtain all values for Z or Y for these subroutines from the DATA statements in Tables 1 and 3.

BASIC. A BASIC program utilizing a routine fashioned after the FORTRAN subroutine TSDDPB is

Table 5
BASIC Program to Compute d′ and Beta

```
10 REM      PROGRAM DPB
20 REM
30 REM      INPUT P(HIT)
40 REM      INPUT P(FALSE ALARM)
50 REM
60 REM      OUTPUT P(H), P(FA), D', BETA
70 REM
80 REM      EXIT PROGRAM WITH P(H) = 0
90 REM
100 GOSUB 1000      \REM READ VECTOR OF Z-SCORES
110 PRINT "P(HIT)              = ";\INPUT H\IF H=0 THEN 9999
120 PRINT "P(FALSE ALARM = ";\INPUT F
130 GOSUB 2000      \REM      COMPUTE D' AND BETA
140 PRINT "P(H)", "P(F.A.)", "DPRIME", "BETA"\PRINT H,F,D,B
150 GOTO 110
1000 REM      SUBROUTINE TSDDPB
1010 REM THEORY OF SIGNAL DETECTION:   D PRIME & BETA
1020 REM -            -       -            - -        -
1030 REM
1040 REM CALCULATES VALUE OF D' AND BETA GIVEN P(H) AND P(FA)
1050 REM
1060 DIM Z(51)
1070 RESTORE
1080 FOR I=1 TO 51\READ Z(I)\NEXT   I\RETURN
2000 H1=INT((H*100)-50+0.0001)\F1=INT((F*100)-50+0.0001)
2010 IF H1>=0 THEN 2030
2020 D1=Z(1-H1)*-1\GOTO 2080
2030 IF H1<>0 THEN 2050
2040 D1=0.00\GOTO 2080
2050 IF H1<=0 THEN 2070
2060 D1=Z(1+H1)\GOTO 2080
2070 PRINT "ERROR - HIT"\GOTO 9999
2080 IF F1>=0 THEN 2100
2090 D2=Z(1-F1)\GOTO 2150
2100 IF F1<>0 THEN 2120
2110 D2=0.00\GOTO 2150
2120 IF F1<=0 THEN 2140
2130 D2=Z(1+F1)*-1\GOTO 2150
2140 PRINT "ERROR - FALSE ALARM"\GOTO 9999
2150 D=D1+D2      \REM      COMPUTE D'
2170 B=EXP((D2*D)-(D*D/2))      \REM      COMPUTE BETA
2180 RETURN
5000 DATA .00000,.02507,.05015,.07527,.10043,.12566,.15097,.17637,.20189,.22754
5010 DATA .25335,.27932,.30548,.33185,.35846,.38532,.41246,.43991,.46770,.49585
5020 DATA .52440,.55338,.58284,.61281,.64335,.67449,.70630,.73885,.77219,.80642
5030 DATA .84162,.87790,.91537,.95417,.99446,1.03643,1.08032,1.12639,1.17499
5040 DATA 1.22653,1.28155,1.34076,1.40507,1.47579,1.55477,1.64485,1.75069
5050 DATA 1.88079,2.05375,2.32635,2.57572
9999 END
```

presented in Table 5 for those laboratories which utilize BASIC compilers/interpreters as the main high-level language.[3] Two subroutine calls are made in this BASIC program. The first (GOSUB 1000) reads the vector of z-scores in the DATA statement and the second (GOSUB 2000) is analogous to the FORTRAN call to the subroutine TSDDPB.

Restrictions. Theoretically, the probability of a hit or false alarm should never be 0.00 or 1.00. (These would correspond to infinite values of d′ and β.) Practically, however, it is possible to obtain these values if highly detectable/discriminable signals are used or if the statistics are derived from a small block of trials (such as might be obtained when examining stimulus/response contingencies within a larger block of trials). Therefore, the z-score value of ±2.57572 and Y-ordinate value of .01447 (corresponding to P = .995 or .005) are selected when the probability of a hit or false alarm is 0.00 or 1.00. The range of d′ and β values derived from these programs is:

$$-5.151 \leqslant d' \leqslant +5.151$$
$$.036 \leqslant \beta \leqslant 27.570.$$

Computer. The functions and subroutines presented in this report were developed initially on an INTERDATA Model 70 minicomputer utilizing the EXTENDED FORTRAN (IV) compiler, run successfully on an IBM 370/158, and revised on a Digital Equipment Corporation PDP-11/10 under the RT-11 operating system. The FORTRAN II and BASIC subroutines were adapted for use on a Digital Equipment Corporation PDP-8/E minicomputer under OS-8.

REFERENCE NOTE

1. Elliot, P. A. *Tables of d′* (Technical Report 97). Ann Arbor: University of Michigan Electronic Defense Group, 1959.

REFERENCES

Egan, J. P. *Signal detection theory and ROC analysis*. New York: Academic Press, 1975.

Egan, J. P., & Clarke, F. R. Psychophysics and signal detection. In J. B. Sidowski (Ed.), *Experimental methods and instrumentation in psychology*. New York: McGraw-Hill, 1966.

Green, D. M., & Swets, J. A. *Signal detection theory and psychophysics*. New York: Wiley, 1966.

Hochhaus, L. A table for the calculation of d′ and β. *Psychological Bulletin*, 1972, **77**, 375-376.

Pastore, R. E., & Scheirer, C. J. Signal detection theory: Considerations for general application. *Psychological Bulletin*, 1974, **81**, 945-958.

Pearson, K. *Tables for staticians and biometricians* (Vol. 2). London: Cambridge University Press, 1931.

Swets, J. A. *Signal detection and recognition by human observers*. New York: Wiley, 1964.

NOTES

1. See Green and Swets (1966) and Egan (1975) for developments of signal detection theory or Egan and Clarke (1966) and Pastore and Scheirer (1974) for brief reviews.

2. The values for the unit-normal curve are obtained from Pearson (1931), as referenced by Hochhaus (1972).

3. The back slash (\) separates multiple statements in a single line when supported by the BASIC interpreter.

(Accepted for publication June 21, 1977.)