Algorithm AS 106: The Distribution of Non-Negative Quadratic Forms in Normal Variables
Author(s): J. Sheil and I. O'Muircheartaigh
Source: *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, Vol. 26, No. 1 (1977), pp. 92-98
Published by: Wiley for the Royal Statistical Society
Stable URL: http://www.jstor.org/stable/2346884
Accessed: 11-03-2015 18:12 UTC

```
    8 IF (J .EQ. NVAR) GOTO 10
      DO 9 K = I2, J1
      DO 9 L = J2, NVAR
      MAT(K, L) = MAT(K, L) +
     *  MIJ * (MAT(I, K) * MAT(J, L) + MAT(K, J) * MAT(I, L))
    9 CONTINUE
C
C        POSITIONS WITH  K AND L GREATER THAN J
C
   10 IF (J .EQ. NVAR) GOTO 12
      DO 11 K = J2, NVAR
      SAVE1 = MAT(I, K)
      SAVE2 = MAT(J, K)
      MAT(I, K) = SAVE1 - MJJ * SAVE2
      MAT(J, K) = SAVE2 - MII * SAVE1
      MAT(K, K) = MAT(K, K) +
     *  MIJ * (MAT(I, K) * SAVE2 + MAT(J, K) * SAVE1)
      IF (K .EQ. NVAR) GOTO 12
      K1 = K + 1
      DO 11 L = K1, NVAR
      MAT(K, L) = MAT(K, L) +
     *  MIJ * (MAT(I, K) * MAT(J, L) + MAT(J, K) * MAT(I, L))
   11 CONTINUE
   12 RETURN
C
   13 IFAULT = 1
      RETURN
      END
```

Algorithm AS 106

The Distribution of Non-negative Quadratic Forms in Normal Variables

By J. Sheil† and I. O'Muircheartaigh

University College, Galway, Ireland

Keywords: QUADRATIC FORM; DISTRIBUTION; NON-NEGATIVE DEFINITE

Language

ISO Fortran

Description and Purpose

Given that the n-dimensional vector $\mathbf{z}$ has a multivariate normal distribution with expected value vector $\boldsymbol{\mu}$ and non-singular covariance matrix $\mathbf{V}$, this algorithm computes the distribution function of the quadratic form $(\mathbf{z}+\mathbf{a})^{\mathrm{T}}\mathbf{C}(\mathbf{z}+\mathbf{a})$ for a fixed vector $\mathbf{a}$ and symmetric positive definite, or positive semi-definite, matrix $\mathbf{C}$. The value of the density is also presented in the output. The quadratic form is expressed as an infinite series in central χ^2 distribution functions: both the distribution functions and the series coefficients are evaluated recursively.

Theory and Numerical Method

$$n = \text{dimensionality of } \mathbf{z}.$$

By making the linear transformations

$$\mathbf{z}-\boldsymbol{\mu} = \mathbf{L}^{\mathrm{T}}\mathbf{R}\mathbf{x}, \quad \mathbf{a}+\boldsymbol{\mu} = \mathbf{L}^{\mathrm{T}}\mathbf{R}\mathbf{b},$$

† Now at University of Newcastle-upon-Tyne.

it has been shown by Ruben (1962) that

$$P\{(\mathbf{z}+\mathbf{a})^{\mathrm{T}}\mathbf{C}(\mathbf{z}+\mathbf{a})\leqslant t\} = P\{(\mathbf{x}+\mathbf{b})^{\mathrm{T}}\mathbf{A}(\mathbf{x}+\mathbf{b})\leqslant t\},$$

where the components of the vector $\mathbf{x}$ are uncorrelated standard normal variables and where $\mathbf{L}$ is the upper triangular matrix defined by $\mathbf{V}=\mathbf{L}^{\mathrm{T}}\mathbf{L}$, $\mathbf{R}$ is the matrix whose columns are the eigenvectors of $\mathbf{LCL}^{\mathrm{T}}$, and $\mathbf{A}=\mathrm{dg}(\alpha_i)$ is a diagonal matrix whose elements α_i are the eigenvalues of $\mathbf{LCL}^{\mathrm{T}}$.

If $f(n',y)$ denotes the probability density of a central χ^2 with n' degrees of freedom and $F(n',y)$ the corresponding distribution function, then combining results from Ruben (1962) and Kotz *et al.* (1967)

$$\begin{aligned} P\{(\mathbf{z}+\mathbf{a})^{\mathrm{T}}\mathbf{C}(\mathbf{z}+\mathbf{a})\leqslant t\} &= \sum_{k=0}^{\infty} c_k F(n'+2k, t/\beta), \quad t>0,\\ &= 0, \qquad\qquad\qquad\qquad t\leqslant 0, \end{aligned}$$

where β is a distribution free constant, and n' is the rank of $\mathbf{C}$. If $0<\beta<2\min_i(\alpha_i)$, the above series is uniformly absolutely convergent for all $t>0$, where $i=1,2,\ldots,n'$.

This algorithm uses $\beta = 0{\cdot}90625\,\alpha_{\min}$ which makes the above series a mixture representation (Robbins and Pitman, 1949). Therefore,

$$c_k \geqslant 0, \quad \sum_{k=0}^{\infty} c_k = 1.$$

If the series is truncated after N terms, then by the monotonic decreasing property of the χ^2 functions

$$\sum_{k=N+1}^{\infty} c_k F(n'+2k, y/\beta) < \left(1-\sum_{k=0}^{N} c_k\right) F(n'+2N, y/\beta).$$

The series coefficients are evaluated as follows:

$$c_0 = A\,e^{-\lambda/2} \quad c_k = k^{-1}\sum_{r=0}^{k-1} g_{k-r}\,c_r,$$

where

$$A=\prod_{j=1}^{n'}\sqrt{(\beta/\alpha_j)}, \quad \lambda=\sum_{j=1}^{n'} b_j^2, \quad g_m=\frac{m}{2}\sum_{j=1}^{n'} b_j^2\gamma_j^{m-1}+\frac{1}{2}\sum_{j=1}^{n'}(1-mb_j^2)\gamma_j^m \quad (m\geqslant 1)$$

and $\gamma_j = 1-\beta/\alpha_j$ $(j=1,2,\ldots,n')$. The χ^2 distribution functions are computed recursively using the relationships

$$F(\nu,a)=F(\nu-2,a)-2f(\nu,a) \quad (\nu\geqslant 3),$$

$$f(\nu,a)=\frac{a}{(\nu-2)}f(\nu-2,a) \quad (\nu\geqslant 3),$$

$$F(2,a)=1-e^{-a/2}, \quad f(2,a)=\tfrac{1}{2}e^{-a/2},$$

$$F(1,a)=2\Phi(\sqrt{a})-1, \quad f(1,a)=e^{-a/2}/\sqrt{(2\pi a)},$$

where $\Phi(.)$ is the standard normal distribution function and is evaluated using Algorithm AS 66 (Hill, 1973).

The method of Cholesky (Ralston, 1965) is used in the triangular decomposition of the matrix $\mathbf{V}$.

Structure

SUBROUTINE QDIST (*N*, *ETA*, *PRECIS*, *V*, *C*, *XA*, *EXPT*, *XLIM*, *XL*, *A*, *VALU*, *GAMA*, *B*, *ANS*, *DNSTY*, *NN*, *IFAULT*)

Formal parameters

N	Integer	input:	dimensionality
ETA	Real	input:	set ETA = smallest positive number representable in the machine
PRECIS	Real	input:	$PRECIS$ = smallest positive number for which $1+PRECIS \neq 1$
V	Real array (N, N)	input:	covariance matrix
C	Real array (N, N)	input:	symmetric matrix of the quadratic form
XA	Real array (N)	input:	fixed vector of quadratic form
EXPT	Real array (N)	input:	expected value vector
XLIM	Real	input:	value of t for which probability is required
XL	Real array (N, N)	work space:	work matrix holding the upper triangular matrix $\mathbf{L}$
A	Real array (N, N)	work space:	work matrix. Contains $\mathbf{LCL}^\mathrm{T}$ in call to AS 60. Holds eigenvectors of $\mathbf{LCL}^\mathrm{T}$ as output from AS 60
VALU	Real array (N)	work space:	work vector holding eigenvalues of $\mathbf{LCL}^\mathrm{T}$
GAMA	Real array (N)	work space:	work vector
B	Real array (N)	work space:	holds elements of non-centrality vector
ANS	Real	output:	resultant probability
DNSTY	Real	output:	density ordinate at t
NN	Integer	output:	numbers of terms used in series
IFAULT	Integer	output:	fault indicator $IFAULT = 0$: no error $IFAULT = 1$: $XLIM$ not positive $IFAULT = 2$: $\mathbf{V}$ is not positive definite $IFAULT = 3$: more than $MITS$ iterations are required in $LRVT$. $MITS$ is set to 30 in a $DATA$ statement, but may be reset by the user $IFAULT = 4$: $\mathbf{C}$ is not positive semi-definite $IFAULT = 5$: More than 100 terms required in the series expansion. The dimensions of $COEFA$ and $COEFB$ can be increased by the user if required

Auxiliary Algorithms

Algorithm AS 60 (Sparks and Todd, 1973a) is used to find eigenvalues and corresponding eigenvectors and the standard normal integral is evaluated by using Algorithm AS 66 (Hill, 1973).

Accuracy

The series is terminated when the maximum possible contribution from the remaining terms is less than 10^{-5}. If greater accuracy is required the assignment to *TEST* 2 in the *DATA* statement should be decreased.

In view of the comments on accuracy in Sparks and Todd (1973b) all tests were carried out using double precision arithmetic.

The algorithm was tested on a UNIVAC 1106 computer and the results compared with those published in Grad and Solomon (1955). A detailed description of results is in O'Muircheartaigh and Sheil's forthcoming paper.

Additional Comment

When **V** and **C** are both unit matrices or when $\mathbf{V} = \mathbf{C}^{-1}$ this algorithm may be used to find the distribution function of both central and non-central χ^2 random variables.

Precision

The version of the algorithm given below is in single precision. To obtain a double precision version it is necessary to make the following alteration to *QDIST*.

1. Change *REAL* to *DOUBLE PRECISION*.
2. Change the real constant assignments in the *DATA* statements to double precision values.
3. Replace *SQRT* by *DSQRT* in the statement before the *DO* 50 loop, before the label 50, in the *DO* 150 loop and in the statement following label 180.
4. Change *EXP* to *DEXP* in the statement immediately following the label 150, in the second statement following label 180 and at label 200.
5. Replace *ALOG* by *DLOG* in the second statement after the label 180.

The value assigned to the local variable *CONST* is $\ln\sqrt{(2\pi)}$. When a double precision version of *QDIST* is being employed, the user may add some more digits (provided the compiler allows it).

To obtain double precision versions of *LRVT*, *TDIAG* and *ALNORM* see Hill (1973) and Sparks and Todd (1973a).

Acknowledgements

The authors wish to thank Dr I. D. Hill and the referee for their helpful comments on an earlier version of the algorithm.

References

Grad, A. and Solomon, H. (1955). Distribution of quadratic forms and some applications. *Ann. Math. Statist.*, **26**, 464–477.

Hill, I. D. (1973). Algorithm AS 66. The normal integral. *Appl. Statist.*, **22**, 424–427.

Kotz, S., Johnson, N. L. and Boyd, D. W. (1976). Series representations of distributions of quadratic forms in normal variables. *Ann. Math. Statist.*, **38**, 838.

O'Muircheartaigh, I. G. and Sheil, J. (1977). Evaluating the distribution of positive definite quadratic forms in normal variables: a correction. (Forthcoming paper.)

Ralston, A. (1965). *A First Course in Numerical Analysis*. New York: McGraw-Hill.

Robbins, H. and Pitman, E. J. G. (1949). Application of the method of mixtures to quadratic forms in normal variables. *Ann. Math. Statist.*, **20**, 552–560.

Ruben, H. (1962). Probability content of regions under spherical normal distributions. IV. The distribution of homogeneous and non-homogeneous quadratic functions of normal variables. *Ann. Math. Statist.*, **33**, 542–570.

Sparks, D. N. and Todd, A. D. (1973a). Algorithm AS 60. Latent roots and vectors of a symmetric matrix. *Appl. Statist.*, **22**, 260–265.

Sparks, D. N. and Todd, A. D. (1973b). A comparison of Fortran subroutines for calculating latent roots and vectors. *Appl. Statist.*, **22**, 220–225.

```
      SUBROUTINE QDIST(N, ETA, PRECIS, V, C, XA, EXPT, XLIM, XL, A,
     *  VALU, GAMA, B, ANS, DNSTY, NN, IFAULT)
C
C        ALGORITHM AS 106  APPL. STATIST. (1977), VOL.26, NO.1
C
C        FINDS THE DISTRIBUTION OF NON-NEGATIVE DEFINITE QUADRATIC FORMS
C        IN NORMAL VARIABLES
C
      REAL
     1  V(N, N), C(N, N), A(N, N), XL(N, N), EXPT(N), XA(N), VALU(N),
     2  GAMA(N), B(N), ETA, PRECIS, XLIM, ANS, DNSTY, TOL, BETA, AP,
     3  XLMDA, XMU, AO, XTOP, PANS, DANS, DN, XN, XNN, BJ2NN, TERM,
     4  ZERO, ONE, TWO, MULT, TEST2, CONST, SUM, XN1, TOTAI, ALNORM,
     5  COEFA(100), COEFB(100)
```

```
      DATA ZERO /0.0/, ONE /1.0/, TWO /2.0/, MULT /0.90625/
      DATA TEST2 /1.0E-5/, CONST /0.9189385332/
      IFAULT = 1
      IF (XLIM .LE. ZERO) RETURN
C
C        TRIANGULAR FACTORISATION OF COVAR. MATRIX - V = ML - L
C        UPPER TRIANGULAR M IS TRANSPOSE OF L
C
      XL(1, 1) = SQRT(V(1, 1))
      DO 50 J = 2, N
      JJ = J - 1
      DO 25 I = 1, JJ
      XL(I, J) = V(I, J) / XL(I, I)
      IF (I .LE. 1) GOTO 25
      KK = I - 1
      DO 20 K = 1, KK
   20 XL(I, J) = XL(I, J) - XL(K, I) * XL(K, J) / XL(I, I)
   25 CONTINUE
      XL(J, J) = V(J, J)
      DO 30 K = 1, JJ
   30 XL(J, J) = XL(J, J) - XL(K, J) ** 2
      IFAULT = 2
      IF (XL(J, J) .LE. ZERO) RETURN
      XL(J, J) = SQRT(XL(J, J))
   50 CONTINUE
C
C        COMPUTE THE SYMMETRIC MATRIX  A = LCM
C
      DO 85 J = 1, N
      DO 85 I = 1, J
      A(I, J) = ZERO
      DO 80 K = I, N
      DO 80 L = J, N
   80 A(I, J) = A(I, J) + XL(I, K) * C(K, L) * XL(J, L)
      A(J, I) = A(I, J)
   85 CONTINUE
C
C        STORE EIGENVALUES OF LCM, IN ORDER OF DESCENDING
C        MAGNITUDE, IN ARRAY-VALU. PUT CORRESPONDING
C        EIGENVECTORS INTO THE ARRAY-A
C
      TOL = ETA / PRECIS
      CALL TDIAG(N, TOL, A, B, GAMA, A)
      CALL LRVT(N, PRECIS, B, GAMA, A, IER)
      IFAULT = 3
      IF (IER .EQ. 1) RETURN
      IFAULT = 4
      N1I = N + 1
      DO 90 I = 1, N
      N1I = N1I - 1
      VALU(I) = B(N1I)
      IF (VALU(I) .LT. ZERO) RETURN
   90 CONTINUE
C
C        INVERT THE UPPER TRIANGULAR MATRIX - L
C
      DO 100 I = 1, N
  100 XL(I, I) = ONE / XL(I, I)
      JEND = N + 1
      DO 120 J = 2, N
      JEND = JEND - 1
      IEND = JEND
      ITOP = JEND - 1
      DO 120 I = 1, ITOP
      KBOT = IEND
      IEND = IEND - 1
      SUM = ZERO
      DO 110 K = KBOT, JEND
  110 SUM = SUM + XL(IEND, K) * XL(K, JEND)
      XL(IEND, JEND) = -SUM * XL(IEND, IEND)
  120 CONTINUE
```

```
C       COMPUTE THE NON-CENTRALITY VECTOR - B
C
      N1I = N + 1
      DO 130 I = 1, N
      N1I = N1I - 1
      B(I) = ZERO
      DO 130 K = 1, N
      DO 130 J = 1, K
      B(I) = B(I) + A(K, N1I) * XL(J, K) * (XA(J) + EXPT(J))
  130 CONTINUE
C
C       NOW COMPUTE THE PROGRAM PARAMETERS USED TO DETERMINE
C       COEFFS IN SERIES
C
      DO 135 I = 1, N
      IF (VALU(I) .EQ. ZERO) GOTO 140
  135 CONTINUE
      NPRIME = N
      GOTO 145
  140 NPRIME = I - 1
  145 BETA = MULT * VALU(NPRIME)
      AP = ONE
      XLMDA = ZERO
      DO 150 I = 1, NPRIME
      XMU = BETA / VALU(I)
      AP = AP * SQRT(XMU)
      XLMDA = XLMDA + B(I) ** 2
      GAMA(I) = ONE - XMU
  150 CONTINUE
      AO = AP * EXP(-XLMDA / TWO)
C
C       DETERMINE THE DENSITY AND C.D.F. OF CENTRAL CHI-SQUARE
C       WITH NPRIME DEGREES OF FREEDOM
C
      XN = NPRIME
      XTOP = XLIM / BETA
      N1 = NPRIME
      NN1 = 0
  160 IF (N1 .LT. 3) GOTO 180
      NN1 = NN1 + 1
      N1 = N1 - 2
      GOTO 160
  180 IF (N1 .EQ. 2) GOTO 200
      PANS = TWO * ALNORM(SQRT(XTOP), .FALSE.) - ONE
      DANS = EXP(-(XTOP + ALOG(XTOP)) / TWO - CONST)
      GOTO 210
  200 PANS = EXP(-XTOP / TWO)
      DANS = PANS / TWO
      PANS = ONE - PANS
  210 IF (NN1 .LE. 0) GOTO 240
      XN1 = N1
      XN1 = XN1 / TWO - ONE
      DO 230 I = 1, NN1
      XN1 = XN1 + ONE
      DANS = DANS * XTOP / (TWO * XN1)
      PANS = PANS - TWO * DANS
  230 CONTINUE
C
C       EVALUATE COEFFICIENTS AND HENCE DETERMINE
C       SUCCESSIVE TERMS OF SERIES
C
  240 XN1 = XN / TWO - ONE
      NN = 0
      ANS = AO * PANS
      DNSTY = AO * DANS / BETA
      DN = DANS
      TOTAI = AO
  270 NN = NN + 1
      COEFB(NN) = ZERO
      XNN = NN
      DO 280 J = 1, NPRIME
      BJ2NN = XNN * B(J) ** 2
```

```
      COEFB(NN) = COEFB(NN) + (BJ2NN + (ONE - BJ2NN) * GAMA(J))
     *  * GAMA(J) ** (NN - 1) / TWO
  280 CONTINUE
      COEFA(NN) = COEFB(NN) * AO / XNN
      IF (NN .LE. 1) GOTO 310
      ITOP = NN - 1
      DO 300 IR = 1, ITOP
      INNR = NN - IR
      COEFA(NN) = COEFA(NN) + COEFB(INNR) * COEFA(IR) / XNN
  300 CONTINUE
  310 XN1 = XN1 + ONE
      DN = DN * XTOP / (TWO * XN1)
      PANS = PANS - TWO * DN
      TERM = PANS * COEFA(NN)
      TOTAI = TOTAI + COEFA(NN)
C
C        COMPUTATION TERMINATED WHEN THE MAXIMUM POSSIBLE
C        CONTRIBUTION FROM THE REMAINING TERMS IN THE SERIES
C        EXPANSION IS LESS THAN TEST2
C
      IF (PANS * (ONE - TOTAI) .LT. TEST2) GOTO 340
      ANS = ANS + TERM
      DNSTY = DNSTY + COEFA(NN) * DN / BETA
      IFAULT = 5
      IF (NN .EQ. 100) RETURN
      GOTO 270
  340 NN = NN - 1
      IFAULT = 0
      RETURN
      END
```

Algorithm AS 107

Operating Characteristics and Average Sampling Number for a General Class of Sequential Sampling Plans

By B. Leventhal

Audits of Great Britain Ltd, Ruislip, Middx.

Keywords: SEQUENTIAL SAMPLING; OPERATING CHARACTERISTICS; AVERAGE SAMPLING NUMBER

Language

ISO Fortran

Purpose

This is an algorithm for calculating the operating characteristic and average sampling number functions for a wide class of closed sequential sampling plans. It is more general than Algorithm AS 67 (McPherson, 1974) at the expense of being slower and of requiring that sampling must terminate.

Applicable Problems

To specify the class of sequential sampling plans to which the algorithm is applicable, we describe a sequential sampling situation and formulate a general class of closed plans that are reasonable to consider.

Suppose that a decision maker has to choose between m "terminal decisions", $d_1, d_2, \dots d_m$, concerning an unknown scalar parameter θ. A common example is where m is 2 and the decisions correspond to the composite hypotheses "$\theta \leqslant \theta^*$" and "$\theta > \theta^*$", where θ^* is a known value. In some situations m is 3, d_1 is "$\theta \leqslant \theta_1^*$", d_2 is "$\theta_1^* < \theta \leqslant \theta_2^*$" and d_3 is "$\theta_2^* < \theta$", for known θ_1^* and θ_2^*.