Computational Statistics & Data Analysis 18 (1994) 449–455
North-Holland

On the computation of the noncentral beta distribution

Cherng G. Ding

Institute of Management Science, National Chiao-Tung University, Taipei, Taiwan, Republic of China

Received September 1991
Revised April 1993

Abstract: Two recurrence formulas for computing the noncentral beta distribution function and the density are proposed. The computations can be carried to any specified accuracy using the error bounds obtained. Corresponding algorithms are provided in a step-by-step form. A close relationship between the formulas allows joint evaluation of the distribution function and the density. The noncentral beta quantile can then be efficiently computed using Newton's method. Its algorithm is also given in a step-by-step form.

Keywords: Error bound; Gamma function; Mean-value theorem; Newton's method; Noncentral beta distribution; Quantile; Series representation

1. Introduction

Let $F(x;\ a,\ b,\ \lambda)$ and $f(x;\ a,\ b,\ \lambda)$ denote, respectively, the noncentral beta distribution function and the density with shape parameters a, b, and noncentrality parameter λ, and $F(x;\ a,\ b)$ and $f(x;\ a,\ b)$ the central beta distribution function and the density with shape parameters a and b. It is well known that (see, e.g., Tang, 1938)

$$F(x;\ a,\ b,\ \lambda) = \sum_{i=0}^{\infty} u_i F(x;\ a+i,\ b), \tag{1}$$

and

$$f(x;\ a,\ b,\ \lambda) = \sum_{i=0}^{\infty} u_i f(x;\ a+i,\ b), \tag{2}$$

where $0 \le x \le 1$, $a > 0$, $b > 0$, $\lambda \ge 0$, and $u_i = \mathrm{e}^{-\lambda/2}(\frac{1}{2}\lambda)^i/i!$. The evaluations of $F(x;\ a,\ b,\ \lambda)$, $f(x;\ a,\ b,\ \lambda)$, and the quantile (percentage point) x_p satisfying $F(x_p;\ a,\ b,\ \lambda) = p$ for given p are always of interest. Lenth (1987) developed a

Correspondence to: Cherng G. Ding, Institute of Management Science, National Chiao-Tung University, 4F, 114 Chung-Hsiao W. Road, Section 1, Taipei, Taiwan, Republic of China.

SSDI 0167-9473(93)E0023-W

recursive algorithm for computing $F(x;\ a,\ b,\ \lambda)$, which was a modification of one given in Norton (1983). Given an accurate auxiliary routine for evaluating $F(x;\ a,\ b)$, both of the algorithms summed up the terms in formula (1) until the corresponding error bounds were less than some desired accuracy. Singh and Relyea (1992) developed a computer program for computing $F(x;\ a,\ b,\ \lambda)$ where $2a$ and $2b$ are both integers based on the error bound given in Lenth (1987) and exact expressions for $F(x;\ a,\ b)$. In this paper, an alternative series representation for $F(x;\ a,\ b,\ \lambda)$ in terms of the central beta densities is developed by using the method similar to that given in Ding (1992). It is easy to compute recursively for all real shape parameters without any auxiliary central beta routine. A simple recursive formula for evaluating $f(x;\ a,\ b,\ \lambda)$ is also given. The accuracy can be easily controlled by using the error bounds provided. The quantile x_p is obtained by Newton's method due to the advantage that the proposed computing formulas for $F(x;\ a,\ b,\ \lambda)$ and $f(x;\ a,\ b,\ \lambda)$ are closely related, and hence they can be evaluated jointly, rather than separately.

2. Numerical methods

Let $\Gamma(a)$ $(a > 0)$ denote the Gamma function. It is easy, using integration by parts, to show that (see also Abramowitz and Stegun, 1965, p. 944)

$$\begin{aligned} F(x;\ a,\ b) &= \frac{\Gamma(a+b)}{\Gamma(a+1)\Gamma(b)} x^a (1-x)^b + F(x;\ a+1,\ b) \\ &= f(x;\ a+1,\ b)(1-x)/(a+b) + F(x;\ a+1,\ b). \end{aligned} \tag{3}$$

The above relation implies that $F(x;\ a+i,\ b)$ can be expressed in terms of a series given by

$$F(x;\ a+i,\ b) = \sum_{k=i}^{\infty} \frac{(1-x)}{(a+b+k)} f(x;\ a+k+1,\ b) = \sum_{k=i}^{\infty} t_k, \quad i = 0, 1, \ldots, \tag{4}$$

where the terms t_k can be obtained recursively by

$$\begin{aligned} t_0 &= \frac{\Gamma(a+b)}{\Gamma(a+1)\Gamma(b)} x^a (1-x)^b, \\ t_i &= t_{i-1} x(a+b+i-1)/(a+i), \quad i \geq 1. \end{aligned} \tag{5}$$

Substituting (4) into (1), we have a new series representation for $F(x;\ a,\ b,\ \lambda)$, which involves the central beta densities:

$$\begin{aligned} F(x;\ a,\ b,\ \lambda) &= \sum_{i=0}^{\infty} u_i \left(\sum_{k=i}^{\infty} t_k \right) \\ &= \sum_{i=0}^{\infty} \left(\sum_{k=0}^{i} u_k \right) t_i \\ &= \sum_{i=0}^{\infty} v_i t_i, \end{aligned} \tag{6}$$

where the terms are evaluated by

$$v_0 = u_0 = e^{-\lambda/2},$$
$$v_i = v_{i-1} + u_i, \quad u_i = u_{i-1}\lambda/(2i), \qquad i \geq 1, \tag{7}$$
$$t_i, \quad i \geq 0, \qquad \text{as in (5)}.$$

Considering equation (4) for $i = 0$, we have

$$F(x; a, b) = \sum_{k=0}^{\infty} t_k = \sum_{k=0}^{n-1} t_k + F(x; a+n, b), \tag{8}$$

where $F(x; a+n, b) = \sum_{k=n}^{\infty} t_k$ is the error of truncation after n terms. It can be shown, by the mean-value theorem, that $F(x; a+n, b)$ is less than or equal to $t_{n-1}x(a+b+n-1)/[(a+n)-(a+b+n)x]$ if $a+n > (a+b+n)x$. Let EF_n be the error of truncation after n terms for the series for $F(x; a, b, \lambda)$ in (6). Since $v_i \leq 1$ for all $i \geq 0$, it follows that, if $a+n > (a+b+n)x$,

$$EF_n = \sum_{i=n}^{\infty} v_i t_i \leq t_{n-1}x(a+b+n-1)/[(a+n)-(a+b+n)x]. \tag{9}$$

Note that the error bound given above is a decreasing function of n when $a+n > (a+b+n)x$, and is a common one used to control the accuracy of evaluations of $F(x; a, b, \lambda)$ and $F(x; a, b)$. The terms in (6), computed recursively through (7), can then be summed up until the error bound is not greater than a predetermined small number ϵ. In fact, evaluation of $F(x; a, b, \lambda)$ through formula (6) requires no central beta routine. Computing the error bound is very easy because it relies only upon the factor t_{n-1} of the last term of the truncated series $\sum_{i=0}^{n-1} v_i t_i$.

The noncentral beta density $f(x; a, b, \lambda)$ expressed in (2) is an infinite weighted sum of central beta densities. Its terms can be evaluated recursively, and are related to those of (6) as follows:

$$f(x; a, b, \lambda) = \sum_{i=0}^{\infty} u_i f(x; a+i, b) = \sum_{i=0}^{\infty} u_i s_i, \tag{10}$$

where

$$s_0 = \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)} x^{a-1}(1-x)^{b-1} = at_0/x/(1-x),$$
$$s_i = s_{i-1}x(a+b+i-1)/(a+i-1) = t_{i-1}(a+b+i-1)/(1-x), \quad i \geq 1, \tag{11}$$

u_i and t_i, $i \geq 0$, are those in (7).

It is clear that, when $a+n > (a+b+n)x$, the sequence $\{s_i\}(i \geq n)$ is decreasing. Letting Ef_n be the truncation error after n terms for the series in (10), we have, when $a+n > (a+b+n)x$,

$$Ef_n = \sum_{i=n}^{\infty} u_i s_i < \sum_{i=n}^{\infty} u_i s_n = s_n\left(1 - \sum_{i=0}^{n-1} u_i\right) = s_n(1 - v_{n-1}). \tag{12}$$

Likewise, the above error bound is a decreasing function of n, and is used to control the accuracy of evaluation of $f(x; a, b, \lambda)$. It is also easy to compute since it relies only upon factors that are computed in the course of evaluating (10).

Let $G(x) = F(x; a, b, \lambda) - p$. The quantile x_p is to be obtained by solving the equation $G(x) = 0$. Since computing formulas for $F(x; a, b, \lambda)$ and $G'(x) = f(x; a, b, \lambda)$ are closely related (the terms in (10) involve the factors u_i same as those for evaluating (6) and the factors s_i whose computations can be based on t_0 in (7)), their evaluations can be combined. It is this property that makes Newton's method for computing the root x_p much more efficient because many redundant computations are avoided. The process is to repeat the iteration (see, e.g., Kennedy and Gentle, 1980, pp. 72–73; and Vandergraft, 1983, pp. 262–264)

$$x_{j+1} = x_j - \frac{\left[F(x_j; a, b, \lambda) - p\right]}{f(x_j; a, b, \lambda)}, \qquad j = 0, 1, \ldots, \tag{13}$$

until the relative absolute difference between two successive iterations is not greater than a specified accuracy δ, i.e., $|x_{j+1} - x_j| \le \delta x_{j+1}$. Since $G(0) = -p$, $G(1) = 1 - p$, and G is strictly increasing in $[0, 1]$, the solution x_p of $G(x) = 0$ is unique. It is obvious that $x_p = 0$ for $p = 0$ and $x_p = 1$ for $p = 1$. No computation is needed for these cases. For $0 < p < 1$, perform iterations with the starting value $x_0 = 0.5$. Within each iteration, the condition that $0 < x_{j+1} < 1$ must be satisfied. If not, replace x_{j+1} by $\frac{1}{2}x_j$ if it is nonpositive, and by $\frac{1}{2}(x_j + 1)$ if it is greater than or equal to 1. This modified rule is easy to implement, and the iterations should converge fast for most cases. Note that evaluations of $F(x; a, b, \lambda)$ and $f(x; a, b, \lambda)$ should be precise enough so that the accuracy of Newton's solution can be warranted. Also, the number of iterations needs to be controlled.

3. Algorithms

Given an accurate routine for computing (the logarithm of) the Gamma function (see Pike and Hill, 1966; or Macleod, 1989) the formulas discussed in Section 2 lead to three simple algorithms for computing the noncentral beta distribution function, the density, and the quantile to any desired accuracy:

Algorithm A (*noncentral beta distribution function*). This algorithm computes the noncentral beta distribution function $F(x; a, b, \lambda)$ for given abscissa x, shape parameters a, b, and noncentrality parameter λ.

A1 (specify the accuracy). Set $EPS \leftarrow \epsilon$. (ϵ denotes the desired accuracy, e.g., 10^{-6}.)

A2 (initialize). Set $n \leftarrow 1$, $\mathrm{t} \leftarrow (\Gamma(a+b)/\Gamma(a+1)\Gamma(b))x^a(1-x)^b$, $u \leftarrow \mathrm{e}^{-\lambda/2}$, $v \leftarrow u$, $CDF \leftarrow vt$.

A3 (compare $a+n$, $(a+b+n)x$). If $a+n > (a+b+n)x$, go to step A5.

A4 (update the term and then accumulate). Set $u \leftarrow u\lambda/(2n)$, $v \leftarrow v + u$, $t \leftarrow tx(a+b+n-1)/(a+n)$, $CDF \leftarrow CDF + vt$, $n \leftarrow n+1$, and return to step A3.

A5 (find the error bound and check for convergence). Set $bound \leftarrow tx(a+b+n-1)/((a+n)-(a+b+n)x)$. If $bound \leq EPS$, terminate the algorithm (CDF is the answer.)

A6 (update the term and then accumulate). Set $u \leftarrow u\lambda/(2n)$, $v \leftarrow v + u$, $t \leftarrow tx(a+b+n-1)/(a+n)$, $CDF \leftarrow CDF + vt$, $n \leftarrow n+1$, and return to step A5.

Algorithm B (*noncentral beta density*). This algorithm computes the noncentral beta density $f(x; a, b, \lambda)$ for given abscissa x, shape parameters a, b, and noncentrality parameter λ.

B1 (specify the accuracy). Set $EPS \leftarrow \epsilon$. (ϵ denotes the desired accuracy, e.g., 10^{-6}.)

B2 (initialize). Set $n \leftarrow 1$, $s \leftarrow (\Gamma(a+b)/\Gamma(a)\Gamma(b))x^{a-1}(1-x)^{b-1}$, $u \leftarrow e^{-\lambda/2}$, $v \leftarrow u$, $PDF \leftarrow us$.

B3 (compare $a+n$, $(a+b+n)x$). If $a+n > (a+b+n)x$, go to step B5.

B4 (update the term and then accumulate). Set $u \leftarrow u\lambda/(2n)$, $v \leftarrow v+u$, $s \leftarrow sx(a+b+n-1)/(a+n-1)$, $PDF \leftarrow PDF + us$, $n \leftarrow n+1$, and return to step B3.

B5 (find the error bound and check for convergence). Set $bound \leftarrow sx(a+b+n-1)(1-v)/(a+n-1)$. If $bound \leq EPS$, terminate the algorithm. (PDF is the answer.)

B6 (update the term and then accumulate). Set $u \leftarrow u\lambda/(2n)$, $v \leftarrow v+u$, $s \leftarrow sx(a+b+n-1)/(a+n-1)$, $PDF \leftarrow PDF + us$, $n \leftarrow n+1$, and return to step B5.

Algorithm C (*noncentral beta quantile*). This algorithm computes the noncentral beta quantile x_p such that $F(x_p; a, b, \lambda) = p$.

C1 (specify the accuracy and the maximum number of Newton's iterations). Set $EPS \leftarrow \epsilon$, $DELTA \leftarrow \delta$, $ITRMAX \leftarrow N_{\max}$. ($\epsilon$ denotes the desired accuracy (e.g., 10^{-6}) for computing $F(x; a, b, \lambda)$ and $f(x; a, b, \lambda)$, and δ the desired accuracy (e.g., 10^{-4}) for computing x_p. $N_{\max}$ is an integer (e.g., 10) used to limit the number of Newton's iterations.)

C2 (set the constants). Set $coeff \leftarrow \Gamma(a+b)/\Gamma(a+1)\Gamma(b)$, $u_0 \leftarrow e^{-\lambda/2}$.

C3 (loop on j, Newton's iteration). First initialize x by setting $x \leftarrow 0.5$. Then perform steps C4 through C10 for $j = 1, 2, \ldots,$ $ITRMAX$. (Steps C4 through C10 constitute one iteration.)

C4 (initialize within each iteration). Set $n \leftarrow 1$, $t \leftarrow coeff \cdot x^a(1-x)^b$, $s \leftarrow at/x/(1-x)$, $u \leftarrow u_0$, $v \leftarrow u$, $CDF \leftarrow vt$, $PDF \leftarrow us$.

C5 (compare $a+n$, $(a+b+n)x$). If $a+n > (a+b+n)x$, go to step C7.

C6 (update the terms and then accumulate for both of $F(x; a, b, \lambda)$ and $f(x; a, b, \lambda)$). Set $u \leftarrow u\lambda/(2n)$, $v \leftarrow v+u$, $s \leftarrow t(a+b+n-1)/(1-x)$, $t \leftarrow tx(a+$

$b+n-1)/(a+n)$, $CDF \leftarrow CDF + vt$, $PDF \leftarrow PDF + us$, $n \leftarrow n+1$, and return to step C5.

C7 (find the corresponding error bounds). Set $bound1 \leftarrow tx(a+b+n-1)/((a+n)-(a+b+n)x)$, $bound2 \leftarrow t(a+b+n-1)(1-v)/(1-x)$.

C8 (check for convergence). If $bound1 \leq EPS$ and $bound2 \leq EPS$, go to step C10.

C9 (update the terms and then accumulate for $F(x; a, b, \lambda)$ and/or $f(x; a, b, \lambda)$). Set $u \leftarrow u\lambda/(2n)$, $v \leftarrow v+u$. If $bound1 \leq EPS$, set $s \leftarrow sx(a+b+n-1)/(a+n-1)$, $PDF \leftarrow PDF + us$, $n \leftarrow n+1$, $bound2 \leftarrow sx(a+b+n-1)(1-v)/(a+n-1)$, and return to step C8; otherwise if $bound2 \leq EPS$, set $t \leftarrow tx(a+b+n-1)/(a+n)$, $CDF \leftarrow CDF + vt$, $n \leftarrow n+1$, $bound1 \leftarrow tx(a+b+n-1)/((a+n)-(a+b+n)x)$, and return to step C8; otherwise set $s \leftarrow t(a+b+n-1)/(1-x)$, $t \leftarrow tx(a+b+n-1)/(a+n)$, $CDF \leftarrow CDF + vt$, $PDF \leftarrow PDF + us$, $n \leftarrow n+1$, and return to step C7.

C10 (find new x and check for convergence of Newton's iterations). Set $diff \leftarrow (CDF-p)/PDF$. If $x - diff \leq 0$, set $x \leftarrow \frac{1}{2}x$; otherwise if $x - diff \geq 1$, set $x \leftarrow \frac{1}{2}(x+1)$; otherwise set $x \leftarrow x - diff$. If $|diff|/x \leq DELTA$, terminate the algorithm. (x is the answer.)

C11 (output error message). Terminate the algorithm with the message "No convergence after N_{max} iterations".

FORTRAN codes based on the above algorithms are available upon request.

Acknowledgements

This work was partially supported by the National Science Council, Republic of China. The author is grateful to Professor Chyan Yang for helpful discussions regarding the algorithms and to a referee for helpful comments which improved the presentation of the algorithms.

References

Abramowitz, M. and I.A. Stegun, *Handbook of mathematical functions* (Dover, New York, 1965).

Ding, C.G., Algorithm AS 275: Computing the non-central χ^2 distribution function, *Appl. Statist.*, 41 (1992) 478–482.

Kennedy, W.J. and J.E. Gentle, *Statistical computing* (Marcel Dekker, New York, 1980).

Lenth, R.V., Algorithm AS 226: Computing noncentral beta probabilities, *Appl. Statist.*, 36 (1987) 241–244.

Macleod, A.J., Algorithm AS 245: A robust and reliable algorithm for the logarithm of the gamma function, *Appl. Statist.*, 38 (1989) 397–402.

Norton, V., A simple algorithm for computing the non-central F distribution, *App. Statist.*, 32 (1983) 84–85.

Pike, M.C. and I.D. Hill, Algorithm 291: Logarithm of the gamma function, *Commun. Ass. Comput. Mach.*, 9 (1966) 684.

Singh, K.P. and G.E. Relyea, Computation of noncentral f probabilities: a computer program, *Comp. Stat. Data Anal.*, 13 (1992) 95–102.

Tang, P.C., The power function of the analysis of variance tests with tables and illustrations of their use, *Statist. Res. Mem.*, 2 (1938) 126–150.

Vandergraft, J.S., *Introduction to numerical computations*, 2nd. ed. (Automated Sciences Group, Silver Spring, MD, 1983).