ELSEVIER Computational Statistics & Data Analysis 43 (2003) 249–267

COMPUTATIONAL STATISTICS & DATA ANALYSIS

www.elsevier.com/locate/csda

Computing discrete mixtures of continuous distributions: noncentral chisquare, noncentral *t* and the distribution of the square of the sample multiple correlation coefficient

Denise Benton, K. Krishnamoorthy*

Department of Mathematics, University of Louisiana at Lafayette, Lafayette, LA 70504-1010, USA

Received 1 October 2001; received in revised form 1 August 2002

Abstract

In this article, we address the problem of computing the distribution functions that can be expressed as discrete mixtures of continuous distributions. Examples include noncentral chisquare, noncentral beta, noncentral F, noncentral t, and the distribution of squared sample multiple correlation. We illustrate the need for improved algorithms by pointing out situations where existing algorithms fail to compute meaningful values of the cumulative distribution functions (cdf) under study. To address this problem we recommend an approach that can be easily incorporated to improve the existing algorithms. For the distributions of the squared sample multiple correlation coefficient, noncentral t, and noncentral chisquare, we apply the approach and give a detailed explanation of computing the cdf values. We present results of comparison studies carried out to validate the calculated values and computational times of our suggested approach. Finally, we give the algorithms for computing the distributions of the squared sample multiple correlation coefficient, noncentral t, and noncentral chisquare so that they can be coded in any desired computer language.

Keywords: Incomplete beta function; Incomplete gamma function; Negative binomial; Poisson distribution

1. Introduction

This article deals with different approaches of computing the cumulative distribution functions (cdfs) that can be written as discrete mixtures of continuous distributions as

* Corresponding author. Tel.: +1-337-482-5283; fax: +1-337-482-5342.
E-mail address: krishna@louisiana.edu (K. Krishnamoorthy).

PII: S0167-9473(02)00283-9

in the following form:

$$P(X \leqslant x) = \sum_{i=0}^{\infty} P(Y = i|\delta) F_{Z_i}(x;\theta), \tag{1.1}$$

where X is the continuous random variable of interest, Y is a discrete random variable, δ is a parameter associated with Y, F_{Z_i} is the cdf of a continuous random variable Z_i. For example, distributions of the square of the sample multiple correlation (see Section 2), noncentral t (Section 3), noncentral chisquare (Section 4), and noncentral beta can be written in the form of (1.1). The following articles, among others, provide algorithms to compute the cdfs which are mentioned above: Lenth (1987) for the noncentral beta; Ding (1992) for the noncentral chisquare; Lenth (1989) for the noncentral t; and Ding and Bargmann (1991) for the distribution of the square of the sample multiple correlation. These algorithms are based on the method which we briefly outline as follows:

Method 1:

1. Evaluate $P(Y = 0|\delta)$ and $F_{Z_0}(x;\theta)$ in (1.1).
2. Compute $P(Y = i|\delta)$, $i = 1, 2, \ldots$, recursively using the initial value $P(Y = 0|\delta)$; compute $F_{Z_i}(x;\theta)$, $i = 1, 2, \ldots$ recursively using the initial value $F_{Z_0}(x;\theta)$.
3. Terminate the series in (1.1) when the sum of the probabilities $P(Y = i|\delta)$ is near 1 or if $F_{Z_i}(x;\theta)$ is a decreasing function of i, terminate the series when $[1 - \sum_{i=0}^{m} P(Y = i|\delta)]F_{Z_{m+1}}(x;\theta)$ is less than a specified fraction (error tolerance).

Method 1 can be easily adopted, because the recursion relations needed in step 2 are available for many commonly used cdfs; however, the algorithms based on Method 1 pose serious problems in the following situations. (i) When the mean of the discrete random variable Y is large, $P(Y = 0|\delta)$ will be very small and hence these algorithms can suffer from under flow error; that is, $P(Y = 0|\delta)$ will be so small that the computer will treat it as zero (see, for example, Helstrom and Ritcey, 1985; Posten, 1993; Frick, 1990). As a consequence, the programs based on Method 1 may return zero (see step 2 of Method 1) even when the actual value of the cdf is quite large. (ii) Computation time of the algorithms based on Method 1 increases drastically as the mean of Y increases. (iii) These algorithms may be ineffective if they are used as auxiliary algorithms to compute values such as percentiles or confidence intervals. For example, the cdf of the square of the sample multiple correlation coefficient R^2 can be used to find confidence limits for the population multiple correlation square, ρ^2 (see Kramer, 1963). In this problem, an algorithm based on Method 1 along with a root searching method will be ineffective to compute confidence limits for ρ^2 when the sample size and ρ^2 are large (see Eq. (2.5)). Another example involves computation of the noncentrality parameter of a noncentral F distribution for a given power of an F test (Tiwari and Yang, 1997). Again, for similar reasons given above, an algorithm based on Method 1 to compute the cdf of noncentral F could be ineffective in computing the noncentrality parameter.

All the above problems can be easily overcome if the initial computation of (1.1) is started at $k = [\text{mean of } Y]$, where $[x]$ denotes the integer closest to x. This is mainly because, in general, the dominant series in (1.1) is the discrete probability $P(Y = i|\delta)$, which attains its maximum around the mean of Y. This alternate approach has been suggested by many authors, among others, Helstrom and Ritcey (1985) and Posten (1993). However, only recently Frick (1990) and Chattamvelli and Shanmugam (1997) have given algorithms based on the alternate approach to compute the cdf of the noncentral beta. Chattamvelli and Shanmugam's comparison study for the noncentral beta clearly indicates that the algorithm based on the alternate approach is more efficient, in terms of accuracy and computation time, than the one based on Method 1. Therefore, we recommend the following method to compute the cdfs of the form (1.1).

Method 2:

Let k be the integer closest to the mean of the discrete random variable Y in (1.1).

1. Evaluate $P(Y=k|\delta)$ and $F_{Z_k}(x;\theta)$ in (1.1).
2. Compute $P(Y=k-i|\delta)$ and $P(Y=k+i|\delta)$, $i=1,2,\ldots,k$, recursively using the initial value $P(Y=k|\delta)$; compute $F_{Z_{k-i}}(x;\theta)$ and $F_{Z_{k+i}}(x;\theta)$, $i=1,2,\ldots,k$ recursively using the initial value $F_{Z_k}(x;\theta)$.
3. When k is large, the series in (1.1) may converge within k iterations in step 2. In such cases, terminate the series in (1.1) when the sum of the discrete probabilities is close to 1.
4. If convergence did not take place in step 2, compute $P(Y=2k+i|\delta)$ and $F_{Z_{2k+i}}(x;\theta)$ recursively for $i=1,2,\ldots,$ until $[1-\sum_{j=0}^{2k+i} P(Y=j|\delta)]F_{Z_{2k+i+1}}(x;\theta)$ becomes less than a specified error tolerance.

It is clear that Methods 1 and 2 are essentially the same, except for step 1 and the "stopping rule" when the mean of Y is large, and hence Method 2 can also be easily adopted as Method 1 to develop algorithms to evaluate the cdfs of form (1.1). Furthermore, the algorithms based on Method 2 give correct values for a wide range of parameter configurations whereas the ones based on Method 1 have some limitations. It should be noted that interval analysis methods suggested by Wang and Kennedy (1994, 1995) can be used to get highly accurate results. But these methods also have some limitations; as pointed out by Wang and Kennedy, they require more computational time and software for extended precision arithmetic. Interval analysis methods, however, are certainly useful to evaluate the accuracy of a scalar computation algorithm. As shown in the following sections, the results of algorithms presented in this paper compare well with existing algorithms but perform better in situations where these existing algorithms fail. Our numerical comparisons in Section 4, for the case of noncentral chisquare distribution, show that the results based on our algorithm are in good agreement with the ones based on the interval computation method given by Wang and Kennedy (1994) whereas the results of the *Applied Statistics* algorithm AS 275 due to Ding (1992) are not.

Although many authors have suggested Method 2, algorithms based on this method are not available for the distributions considered in this paper. The main purpose of this article is to provide easy reference to computational algorithms for computing the distribution functions of the noncentral t, noncentral chisquare, and square of the sample multiple correlation coefficient. In the following sections we give necessary formulas, recursion relations, and stopping rules to compute the cdfs of the square of the sample multiple correlation coefficient (Section 2), noncentral t (Section 3), and noncentral chisquare (Section 4). In Section 5, we compare the new algorithms and the AS algorithms for speed. In Section 6, we give some concluding remarks and references to the auxiliary algorithms needed by the algorithms presented in this article. The algorithms based on Method 2 for computing these cdfs are given in Section 7; they are presented in such a way that they can be coded in any computer language such as Fortran or C.

2. Distribution of the square of the sample multiple correlation coefficient

For a set of variables from a multivariate normal population, the multiple correlation coefficient can be defined as the maximum correlation that is attainable between one variate and a linear combination of the remaining variables in the set. The distribution of the square of the sample multiple correlation coefficient R^2 is useful in testing hypotheses about the population multiple correlation coefficient ρ^2 when the null value is nonzero. For example, in selecting explanatory variables to be included in a multiple regression model, one may test whether the observed value of R^2 for a model with explanatory variables is significantly greater than the observed R^2 for a model with fewer explanatory variables. Also, this distribution can be used to compute the power of the test $\mathrm{H}_0 : \rho^2 = 0$, when the sample is taken from a multivariate normal distribution (Muirhead, 1982, p. 171).

Let $X_1, \ldots, X_N$ be a sample of independent vector observations from a p-variate normal population with mean μ and covariance matrix Σ. Define

$$\bar{X} = \frac{1}{N}\sum_{i=1}^{N} X_i \quad \text{and} \quad \mathbf{A} = \sum_{i=1}^{N} (X_i - \bar{X})(X_i - \bar{X})'. \tag{2.1}$$

Partition A as

$$\mathbf{A} = \begin{pmatrix} a_{11} & \mathbf{a}_{12} \\ \mathbf{a}_{21} & \mathbf{A}_{22} \end{pmatrix},$$

so that a_{11} is 1×1, $\mathbf{a}_{12}$ is $1 \times (p-1)$, and $\mathbf{A}_{22}$ is $(p-1) \times (p-1)$ matrix. In terms of these submatrices, the squared sample multiple correlation coefficient is given by

$$R^2 = \frac{\mathbf{a}_{12}\mathbf{A}_{22}^{-1}\mathbf{a}_{21}}{a_{11}}. \tag{2.2}$$

The sample multiple correlation coefficient is the positive square root of R^2. The population multiple correlation square, ρ^2, is defined similarly in terms of the submatrices of Σ.

The distribution function of R^2 can be written as

$$P(R^2 \leqslant x) = \sum_{i=0}^{\infty} P(Y = i) I_x\left(\frac{p-1}{2} + i, \frac{n-p+1}{2}\right), \tag{2.3}$$

where $n = N - 1$,

$$I_x(a,b) = \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)} \int_0^x t^{a-1}(1-t)^{b-1}\,\mathrm{d}t \tag{2.4}$$

is the incomplete beta function, and

$$P(Y=i) = \frac{\Gamma(n/2+i)}{\Gamma(i+1)\Gamma(n/2)} (\rho^2)^i (1-\rho^2)^{n/2}. \tag{2.5}$$

The expression in (2.5), whether or not $n/2$ is an integer, can be regarded as the negative binomial probability mass function with success probability $(1-\rho^2)$, the number of successes $n/2$, and the number of failures i (see Muirhead, 1982, p. 175). Therefore, $P(Y=i)$ attains its maximum around the mean $n\rho^2/(2(1-\rho^2))$. Let

$$k = [n\rho^2/(2(1-\rho^2))],$$

where $[x]$ denotes the integer which is closest to x. To evaluate the distribution function of R^2, we first compute the kth term in (2.3) and then we compute other terms using forward and backward recursive relations. The following recursive relations for $P(Y=i)$ in (2.5) can be easily obtained:

$$P(Y=i+1) = \frac{n/2+i}{i+1} \rho^2 P(Y=i), \quad i = 0,1,2\ldots$$

and

$$P(Y=i-1) = \frac{i}{n/2+i-1} \rho^{-2} P(Y=i), \quad i = 1,2,\ldots\ .$$

Further, we have the following well-known recursion relations (see Abramovitz and Stegun, 1964, 26.5.16) for the incomplete beta function in (2.4):

$$I_x(a+1,b) = I_x(a,b) - \frac{\Gamma(a+b)}{\Gamma(a+1)\Gamma(b)} x^a (1-x)^b \tag{2.6}$$

and

$$I_x(a-1,b) = I_x(a,b) + \frac{\Gamma(a+b-1)}{\Gamma(a)\Gamma(b)} x^{a-1} (1-x)^b. \tag{2.7}$$

To compute the second term on the right-hand side of (2.6) or (2.7), we need to evaluate $g(x;a,b) = \Gamma(a+b-1)/[\Gamma(a)\Gamma(b)]x^{a-1}(1-x)^b$ only once. Other terms can be evaluated using the relation $\Gamma(a+1) = a\Gamma(a)$. For example, the second term on the right-hand side of (2.6) is $x(a+b-1)g(x;a,b)/a$.

Stopping rule: While computing the $k \pm i$ terms using forward and backward recursions, stop when $[1 - \sum_{j=k-i}^{k+i} P(Y=j)]$ is smaller than the error tolerance or when the number of iterations is greater than a specified number; else stop if $[1 - \sum_{j=0}^{2k+i} P(Y=j)] I_x((p-1)/2 + 2k + i + 1, (n-p-1)/2)$ is less than or equal to the error tolerance or the number of iterations is greater than a specified number.

The algorithm to compute the cdf of R^2 is given in Section 7. We evaluated $P(R^2 \leqslant x | n, \rho^2, p)$ using our Algorithm 7.1 and the *Applied Statistics* algorithm

Table 1
Computed values of $P(R^2 \leqslant x|n, \rho^2, p)$ using Algorithm 7.1 and algorithm AS 260

x	ρ^2	p	n	Algorithm 7.1	AS 260
0.8	0.7	3	21	$0.77709111520\underline{7214}$	...7575
0.1	0.3	5	12	$1.25731267973\underline{7902\text{d-2}}$	...7896d-2
0.9	0.9	4	100	$0.43822559805\underline{1816}$	...1845
0.9	0.9	12	1200	0.433940873305539	0
0.8	0.8	6	1000	0.466114882398756	0
0.8	0.8	6	600	$0.4562254141\underline{23004}$	...22462
0.8	0.8	6	900	$0.46427799\underline{3696865}$	...3535148
0.6	0.6	12	1500	$0.4297101475\underline{65932}$	...66410
0.6	0.6	12	1600	$0.431930\underline{627893402}$	...613046914
0.6	0.6	12	1650	0.432964762618524	0

AS 260 (Ding and Bargmann, 1991) for some selected values of ρ^2, n and p. These probabilities are given in Table 1. As we have already mentioned, the results of AS 260 are not accurate when n is large and/or when ρ^2 is large. We give three examples in Table 1 where AS 260 returns 0 for the value of the cdf evaluated at x, but the correct value is nonzero.

3. Noncentral *t* distribution

The need for the noncentral t distribution arises in several well-known problems, most notably to determine the power of any test based on the Student's t statistic. The percentiles of noncentral t distributions are needed to compute the one-sided tolerance limits for a normal population and for random effects model (see Vangel, 1992). Other uses of the noncentral t distribution include computing confidence intervals and hypothesis tests about the independent variable in a multivariate–univariate calibration problem (Benton et al., 2002). The noncentral t distribution with noninteger degrees of freedom is needed for constructing tolerance limits for $X_1 - X_2$ (Hall, 1984) and for making inference about the reliability parameter $P(X_1 > X_2)$, where X_1 and X_2 are independent normal random variables (Reiser and Guttman, 1986).

Let X have a normal distribution with mean δ and variance 1, and let nS^2 have a chisquare distribution with df $=n$. Assume that X and S^2 are independent. The distribution of $t_n(\delta) = X/S$ is called noncentral t distribution with df $=n$ and noncentrality parameter δ. It follows from the definition of $t_n(\delta)$ that

$$P(-\infty \leqslant t_n(\delta) \leqslant 0) = \Phi(-\delta),$$

where $\Phi(.)$ is the standard normal distribution. Thus, for any $t > 0$, using Guenther's (1978) series expansion for $P(0 < t_n(\delta) \leqslant t)$, the distribution function of $t_n(\delta)$ can be expressed as

$$P(t_n(\delta) \leqslant t) = \Phi(-\delta) + P(0 < t_n(\delta) \leqslant t)$$

$$= \Phi(-\delta) + \frac{1}{2}\sum_{i=0}^{\infty}\left[P_i I_x\left(i+\frac{1}{2},\frac{n}{2}\right) + \frac{\delta}{\sqrt{2}} Q_i I_x\left(i+1,\frac{n}{2}\right)\right], \tag{3.1}$$

where $I_x(a,b)$ denotes the incomplete beta function given in (2.4), $x = t^2/(n+t^2)$,

$$P_i = \mathrm{e}^{-\delta^2/2}(\delta^2/2)^i/i!$$

and

$$Q_i = \mathrm{e}^{-\delta^2/2}(\delta^2/2)^i/\Gamma(i+3/2).$$

For $t < 0$, the distribution function can be computed using the relation

$$P(t_n(\delta) \leqslant t) = 1 - P(t_n(-\delta) \leqslant -t). \tag{3.2}$$

It is to be noted that Craig's (1942, Eq. (4)) series expansion for

$$P(0 < t_n(\delta) < t) = \frac{\mathrm{e}^{-\delta^2/2}}{2}\sum_{i=0}^{\infty}\frac{(\delta^2/2)^{i/2}}{\Gamma(i/2+1)} I_x((i+1)/2, n/2) \tag{3.3}$$

is valid only for nonnegative δ whereas Guenther's series given in (3.1) is valid for any δ. This is mainly because Craig used the relation $\delta = \sqrt{(\delta^2)}$ which is valid only for $\delta \geqslant 0$. Although the expression for the cdf of noncentral t given in (3.1) is not exactly of the form (1.1), the computational procedures to evaluate (3.1) are essentially the same as those for computing (1.1) as shown below.

The following recursion relations for P_i and Q_i can be easily obtained:

$$P_{i+1} = \frac{\delta^2/2}{i+1} P_i, \qquad P_{i-1} = \frac{i}{\delta^2/2} P_i \tag{3.4}$$

and

$$Q_{i+1} = \frac{\delta^2/2}{i+3/2} Q_i, \qquad Q_{i-1} = \frac{i+1/2}{\delta^2/2} Q_i. \tag{3.5}$$

To obtain a bound for the truncation error, let E_m denote the remainder of the infinite series (3.1) after the mth term. Using the facts that $P_i \geqslant Q_i$ and $I_x(a,b)$ is a decreasing function of a, we get

$$\begin{aligned}|E_m| &\leqslant \frac{1}{2}(1+|\delta|/2) I_x(m+3/2, n/2)\sum_{i=m+1}^{\infty} P_i \\ &= \frac{1}{2}(1+|\delta|/2) I_x(m+3/2, n/2)\left(1 - \sum_{i=0}^{m} P_i\right).\end{aligned} \tag{3.6}$$

The error bound in (3.6) is different from the one given in Lenth (1989). Lenth's expression for the error bound appears to be inaccurate; he ignores the fact that his q_j involves $\delta/2$ and hence the relation $p_j > q_j$ does not hold for all δ.

Stopping rule: Forward computation of (3.1) may be stopped once the right-hand side of (3.6) is less than or equal to a specified error tolerance or when the number of iterations exceeds a specified number. Furthermore, forward and backward computations, that is, computation of the $k \pm i$th terms can be stopped when $(1 - \sum_{j=k-i}^{k+i} P_i)$

Table 2
Computed values of $P(t_n(\delta) \leqslant x)$ using Algorithm 7.2 and algorithm AS 243

x	n	δ	Algorithm 7.2	AS 243
2.34	3	1	$0.80188899961\underline{3917}$	...3844
−4.33	126	−2	$1.252846196\underline{792878\text{d-}2}$	...6846891d-2
23	20	23	$0.46013440039\underline{1924}$	...1458
34	20	33	$0.53200838637\underline{8725}$	...8130
39	12	38	$0.4958681\underline{84917805}$	...4761038
39	12	39	0.446304024668836	0
39	200	38	$0.6661942\underline{09961795}$	...9750795
40	200	42	0.179292265426085	0

is less than the error tolerance or when the number of iterations exceeds a specified number.

Using the above stopping rule, and recursion relations (3.4) and (3.5) along with those for the incomplete beta functions given in (2.6) and (2.7), we give the algorithm to compute the noncentral t distribution function in Section 7.

For comparison purpose, we evaluated $P(t_n(\delta) \leqslant x)$ using the *Applied Statistics* algorithm AS 243 (Lenth, 1989) and our Algorithm 7.2 given in Section 7. The probabilities are given in Table 2 for some selected values of n, δ and x. From the table values, we see that the results of AS 243 are not correct for larger values of δ. In particular, we note that when $(x, n, \delta) = (39, 12, 38)$ the results are close; whereas when $(x, n, \delta) = (39, 12, 39)$, AS 243 returns 0, which is incorrect.

4. Noncentral chisquare distribution

The noncentral chisquare distribution is necessary for calculating the power of tests involving a chisquare test statistic, such as chisquare tests of independence or Breusch–Pagan tests for constancy of error variance in a linear regression problem (Neter et al., 1996, p. 115). Percentile values from a noncentral chisquare distribution are useful in computing tolerance limits for a normal population and also tolerance regions for a multivariate normal population (see Krishnamoorthy and Mathew, 1999)

Let $X_1, \ldots, X_n$ be independent normal random variables with mean μ_i and common variance 1 for $i = 1, 2, \ldots, n$. Then, the distribution of $\chi_n^2(\lambda) = \sum_{i=1}^n X_i^2$ is called noncentral chisquare distribution with df $= n$ and noncentrality parameter $\lambda = \sum_i^n \mu_i^2$. Alternatively, a random variable is said to have noncentral chisquare distribution if its probability density function is given by

$$\sum_{i=0}^{\infty} \frac{e^{-\lambda/2}(\lambda/2)^i}{i!} \frac{e^{-x/2} x^{n/2+i-1}}{2^{n/2+i}\Gamma(n/2+i)}, \quad x > 0. \tag{4.1}$$

Note that in the former definition the degrees of freedom n should be a positive integer whereas in the alternate definition n could be any positive number. It follows from (4.1) that the cdf of $\chi_n^2(\lambda)$ is

$$P(\chi_n^2(\lambda) \leqslant x) = \sum_{i=0}^{\infty} \frac{\mathrm{e}^{-\lambda/2}(\lambda/2)^i}{i!} P(\chi_{n+2i}^2 \leqslant x)$$

$$= \sum_{i=0}^{\infty} \frac{\mathrm{e}^{-\lambda/2}(\lambda/2)^i}{i!} I_{x/2}(n/2+i), \tag{4.2}$$

where

$$I_y(a) = \frac{1}{\Gamma(a)} \int_0^y \mathrm{e}^{-x} x^{a-1}\,\mathrm{d}x, \qquad a > 0,\ x > 0,$$

is the incomplete gamma function. To get the second step of (4.2), we used the relation that $P(\chi_a^2 \leqslant x) = P(Y \leqslant x/2)$, where Y is a gamma random variable with shape parameter $a/2$.

To compute (4.2), we need the following recursion relations (Abramovitz and Stegun, 1964, 6.5.21) for the incomplete gamma function:

$$I_x(a+1) = I_x(a) - \frac{x^a \mathrm{e}^{-x}}{\Gamma(a+1)} \tag{4.3}$$

and

$$I_x(a-1) = I_x(a) + \frac{x^{a-1}\mathrm{e}^{-x}}{\Gamma(a)}. \tag{4.4}$$

Further, it follows from (4.3) that

$$I_x(a) = \frac{x^a \mathrm{e}^{-x}}{\Gamma(a+1)} \left(1 + \frac{x}{(a+1)} + \frac{x^2}{(a+1)(a+2)} + \cdots \right), \tag{4.5}$$

which can be used to evaluate $I_x(a)$. To compute (4.2), first evaluate the kth term, where k is the integral part of $\lambda/2$, and then compute other terms recursively.

Stopping rule: Let $P(Y=j) = \mathrm{e}^{-\lambda/2}(\lambda/2)^j/j!$. While computing the $k \pm i$ terms stop if $[1 - \sum_{j=k-i}^{k+i} P(Y=j)]$ is less than error tolerance or the number of iterations is greater than a specified integer; else stop if $[1 - \sum_{j=0}^{2k+i} P(Y=j)] I_x(2k+i+1)$ is less than error tolerance or the number of iterations is greater than a specified integer.

Using the above stopping rule, recursion relations (4.3) and (4.4), and the recursion relations for Poisson probabilities given in (3.4), we give the algorithm for computing the cdf of the noncentral chisquare distribution in Section 7.

In order to demonstrate that the results based on our scalar algorithm are comparable with those based on interval analysis methods, we computed $P(\chi_n^2(\delta))$ using our algorithm and the *Applied Statistics* algorithm AS 275 for the parameter configurations considered in Wang and Kennedy (1994). The results of both Algorithms 7.3 in Section 7 and AS 275 are compared with those of interval computation given in Table 6 of Wang and Kennedy (1994). The results are given in Table 3. We see from these table values that the results of our Algorithm 7.3 are closer to those of interval

Table 3
Computed values of $P(\chi^2_n(\delta) \leqslant x)$ using interval computation, Algorithm 7.3 and algorithm AS 275

x	n	δ	Interval Computation	Algorithm 7.3	Algorithm AS 275
0.00393	1.0	6.0	0.2498463724258039d-2	…58047d-2	…47769d-2
9.23636	5.0	1.0	0.8272918751175548d0	…75470d0	…71730d0
24.72497	11.0	21.0	0.2539481822183126d0	…182580d0	…179610d0
44.98534	31.0	6.0	0.8125198785064969d0	…064480d0	…060920d0
38.56038	51.0	1.0	0.8519497361859118d-1	…8584550d-1	…8068580d-1
82.35814	100.0	16.0	0.1184348822747824d-1	…7441880d-1	…6932340d-1
331.78852	300.0	16.0	0.7355956710306709d0	…05250d0	overflow
459.92612	500.0	21.0	0.2797023600800060d-1	…07884640d-1	overflow
0.00016	1.0	1.0	0.6121428929881423d-2	…81473d-2	…81051d-2
0.00393	1	1	0.303381422975380d-1	…75378d-1	…75253d-1

computation than the results of AS 275. Furthermore, AS 275 resulted in an overflow error for two of the 10 cases. These overflow errors tend to occur when the degrees of freedom is large.

5. CPU time comparisons

In order to understand the speed of the new algorithms and the existing AS algorithms, we computed the CPU times for all the algorithms using the function subroutine CPU_TIME() of *Compaq Visual Fortran* 6.5. The calculations were made using a Pentium IV (1.8 GhZ) computer. Because the CPU time for single evaluation is very small (the function routine returns zero), we computed the CPU time for multiple computations. Details are given in the following subsections.

5.1. Speed comparison between Algorithms 7.1 and AS 260 for computing the CDF of the squared sample multiple correlation coefficient

For fixed x, n, and 10,000 randomly generated ρ^2 from Uniform(a, b) distribution, the CPU times required to evaluate $P(R^2 \leqslant x|\rho^2, n, p)$ are computed for Algorithms 7.1 and AS 260. The CPU times are reported in Table 4 for some selected values of a and b. In Table 4, k_1 denotes the number of times the absolute difference between the computed values using Algorithm 7.1 and AS 260 exceeds 10^{-7}; k_2 denotes the number of times AS 260 returns zero while Algorithm 7.1 yielded a value greater than 0.002. It is clear from the table values that AS 260 is as good as Algorithm 7.1 when ρ^2 and the sample size are small. When $0 < \rho^2 < 0.5$ and sample sizes are large, Algorithm 7.1 is faster than AS 260. For $\rho^2 \in (0.5, 0.8)$, Algorithm 7.1 is not only faster than AS 260 but also returns accurate values in all the cases considered. For $\rho^2 \in (0.8, 1)$ and large values of x, AS 260 returns many inaccurate values even though in two situations it is faster than Algorithm 7.1.

Table 4
Time comparison between Algorithms 7.1 and AS 260 for computing $P(R^2 \leqslant x|\rho^2, n, p)$ 10,000 times (time in second)

	x	p	n	Alg. 7.1	AS 260	k_1	k_2
$\rho^2 \sim U(0, 0.5)$	0.5	3	20	0.08	0.08	0	0
	0.5	3	35	0.08	0.09	0	0
	0.5	3	500	0.14	0.22	0	0
	0.5	3	1000	0.18	0.33	0	0
$\rho^2 \sim U(0.5, 0.8)$	0.6	5	30	0.09	0.12	0	0
	0.9	4	500	0.33	0.78	0	0
	0.6	12	700	0.41	0.68	0	0
	0.9	4	1000	0.44	5.35	1010	804
$\rho^2 \sim U(0.8, 1)$	0.8	12	600	6.39	2.56	0	0
	0.8	12	900	6.50	76.7	2441	1039
	0.9	12	600	6.53	2.74	923	85
	0.9	12	900	6.40	9.32	6156	5232

Table 5
Time comparison between Algorithms 7.2 and AS 243 for computing $P(t_n(\delta) \leqslant x|n, \delta)$ 10,000 times (time in second)

	x	n	Alg. 7.2	AS 243	k_1	k_2
$\delta \sim N(3, 1)$	3	2	0.21	0.11	0	0
	3	12	0.20	0.11	0	0
	3	300	0.26	0.35	0	0
	3	1200	0.55	1.10	0	0
$\delta \sim N(15, 3)$	15	3	0.94	0.26	0	0
	15	12	0.29	0.26	0	0
	15	500	0.28	0.20	0	0
	15	1200	0.30	0.20	0	0
$\delta \sim N(32, 1)$	32	12	0.46	0.67	12	0
	32	300	0.49	0.61	127	0
$\delta \sim N(35, 1)$	35	12	0.49	0.77	48	2
$\delta \sim N(39, 1)$	39	12	0.57	3.90	7857	6644
$\delta \sim N(40, 1)$	40	12	0.60	4.67	9640	9238

5.2. *Speed comparison between Algorithms 7.2 and AS 243 for computing the CDF of noncentral t distribution*

For fixed x and n, the CPU times required to evaluate $P(t_n(\delta) \leqslant x)$ for 10,000 δ's generated from a $N(\mu, \sigma^2)$ distribution are computed. The CPU times are presented in Table 5, $k_1 =$ the number of times the absolute difference between the computed values using Algorithm 7.2 and AS 243 exceeds 10^{-7} and $k_2 =$ the number of times AS 260 returns zeroes when Algorithm 7.2 produces a value greater than 0.002. It appears that, for δ around 32 or less, the algorithm AS 243 is in general faster than Algorithm 7.2

Table 6
Time comparison between Algorithms 7.3 and AS 275 for computing $P(\chi_n^2(\delta) \leqslant x|n,\delta)$ 10,000 times (time in second)

	x	n	Alg. 6.3	AS 275	k_1	k_2
$\delta \sim \text{abs}(N(2,1))$	5	5	0.05	0.04	0	0
	12	12	0.06	0.05	0	0
	400	200	0.15	0.34	0	0
	300	290	0.09	37.98	10^4	10^4
$\delta \sim \text{abs}(N(20,1))$	40	20	0.05	0.07	0	0
	60	40	0.06	0.07	0	0
	220	200	0.08	0.09	0	0
	340	280	0.10	37.78	10^4	0
$\delta \sim \text{abs}(N(280,1))$	290	10	0.15	0.21	0	0
	500	220	0.16	0.23	0	0
	800	520	0.17	overflow	—	—
	1500	30	0.70	0.53	10^4	10^4
$\delta \sim \text{abs}(N(1000,1))$	1000	5	0.25	0.97	87	0
	1200	200	0.25	0.83	186	0
	1300	290	0.26	31.72	10^4	0
	1500	30	0.70	0.53	10^4	10^4

and returns accurate values. For $\delta > 35$, AS 243 returns inaccurate values and is much slower than Algorithm 7.2.

5.3. Speed comparison between Algorithms 7.3 and AS 275 for computing the CDF of the noncentral chisquare distribution

To compare the CPU times of Algorithm 7.3 and AS 275, we evaluated $P(\chi_n^2(\delta) \leqslant x|n,\delta)$ for fixed x and n, and using 10,000 absolute values of $N(\mu,\sigma^2)$ random numbers for δ. The total times required by both algorithms are presented in Table 6. In this table, k_1 denotes the number of times the absolute difference between the computed values using Algorithm 7.3 and AS 275 exceed 10^{-10} and k_2 denotes the number of times AS 275 returns zero when the actual value is significantly greater than zero. We observe from the table values that AS 275 is in general slower than Algorithm 7.3. Furthermore, we observed that algorithm AS 275 suffers from overflow errors or returns zeroes in the following situations.

(i) When the noncentrality parameter is 1490 or above (regardless of the values of other parameters).
(ii) The degrees of freedom n is 290 or above and x is close to $n + \delta$.
(iii) The value of x is greater than or equal to 1500 (regardless of the values of other parameters).

In each of the above situations, we computed the probabilities for various values of x including values close to the mean $n + \delta$ of the noncentral chisquare distribution. It is expected that the cumulative probabilities around the mean are close to 0.5 or at least significantly greater than zero. But algorithm AS 275 returned 0 or encountered overflow problem.

6. Concluding remarks

In this article, we provided algorithms to compute the cumulative distribution functions of the squared sample multiple correlation coefficient, noncentral t, and noncentral chisquare. These algorithms are modifications of the existing *Applied Statistics* algorithms. We showed that the AS algorithms have limitations under some parameter configurations. In particular, they return a probability of zero when, in fact, there should be significant probability, and sometimes suffer from overflow errors. The modified algorithms presented in this paper overcome these problems. We have also shown that the results based on Algorithm 7.3 for computing chisquare cdf are in good agreement with those based on interval computation reported in Wang and Kennedy (1994). We were unable to do such comparison studies for the cdfs of noncentral t and R^2 because algorithms or codes needed for interval computation are not available. However, we believe that the algorithms for evaluating the cdfs of noncentral t and R^2 will share the properties of the algorithm for evaluating the noncentral chisquare cdf because they are all based on Method 2 given in the introduction. Thus, the new algorithms, although slower in some situations, are certainly preferable to the existing ones.

The auxiliary algorithms (to compute incomplete beta function and natural logarithm of gamma function) which are needed by Algorithms 7.1–7.3 can be obtained, for example, from the FTP site: lib.stat.cmu.edu. This site has an almost complete collection of algorithms from *Applied Statistics*.

7. Algorithms

Algorithm 7.1. Distribution function of the squared sample multiple correlation.

Input:

ns = sample size, ($ns \geqslant p$)

p = number of variates, ($p \geqslant 2$)

x=the value at which the distribution function is to be computed, ($0 < x < 1$)

r=squared population multiple correlation, ($0 < r < 1$)

errtol=error tolerance (10^{-6} for single precision, and 10^{-12} for double precision)

maxitr=maximum number of iterations

Output:

cdf = $P(R^2 \leqslant x)$

```
Set:
    $n = ns - 1$
    $k =$ the integral part of $nr/(2 * (1 - r))$
    $a = (p - 1)/2 + k$
    $b = (n - p + 1)/2$
Compute the beta distribution of the $k$th term, and assign it to "betaf" and "betab"
so that it can be called later for forward and backward recursion:
    betaf = beta distribution at $(x, a, b)$
    betab = betaf
"xgamf" is an initialization to compute the second term on the right-hand side of
(2.6) recursively:
    xgamf $= \exp((a-1) * \ln(x) + b * \ln(1-x) + \ln\Gamma(a+b-1) - \ln\Gamma(a) - \ln\Gamma(b))$
$\ln\Gamma(b))$
"xgamb" is an initialization to compute the second term on the right-hand side of
(2.7) recursively:
    xgamb = xgamf $* (a + b - 1) * x/a$
Compute the $k$th term of negative binomial and assign it to"pnegbf" and "pnegbb"
so that it can be used for forward and backward recursions:
    pnegbf $= \exp(\ln(n/2+k) - \ln\Gamma(k+1) - \ln\Gamma(n/2) + k * \ln(r) + (n/2) * \ln(1-r))$
    pnegbb = pnegbf
Compute the remainder of the negative binomial probabilities:
    remain = 1 − pnegbf
    cdf = pnegbf * betaf
    $i = 1$
1   xgamf = xgamf $* (a + b + i - 2) * x/(a + i - 1)$
    betaf = betaf − xgamf
    pnegbf = pnegbf $* (n/2 + k + i - 1) * r/(k + i)$
    cdf = cdf + pnegbf * betaf
    error = remain * betaf
    remain = remain − pnegbf
Do forward and backward computations $k$ times or until convergence
    **if i > k then**
    if error $\leqslant$ errtol or i > maxitr **return**
    $i = i + 1$
    goto 1
    **else**
    xgamb = xgamb $* (a - i + 1)/(x * (a + b - i))$
    betab = betab + xgamb
    pnegbb = pnegbb $* (k - i + 1)/(r * (n/2 + k - i))$
    cdf = cdf + pnegbb * betab
    remain = remain − pnegbb
    if remain $\leqslant$ errtol or $i >$ maxitr **return**
    $i = i + 1$
    goto 1
    **end if**
    **end**
```

Algorithm 7.2. Noncentral t distribution function.

```
Input:
        delta = noncentrality parameter $\delta$, $(-\infty < \delta < \infty)$
        df = degrees of freedom $n$, $(n > 0)$
        $t$ = the real number for which $P(t_n(\delta) \leqslant \mathrm{t})$ is to be computed
        errtol = error tolerance
        maxitr = maximum number of iterations
Output:
        cdf = $P(t_n(\delta) \leqslant \mathrm{t})$
If $t < 0$, then the transformation in (3.2) must be used; In this case,
Set: $x = -t$, del = −delta
If $t > 0$, then
Set: $x = t$, del = delta
Compute the normal cdf at (−del):
        pnorm = $\Phi$(−del)
        if $x = 0$, set cdf = pnorm, return
Set:
        $y = x * x/(df + x * x)$
        dels = del * del/2
        $k$ = integral part of (dels)
        $a = k + 1/2$
        $c = k + 1$
        $b = df/2$
Initialization to compute the $P_k$'s:
        pkf = exp(−dels + $k$ * ln(dels) − ln($k$ + 1))
        pkb = pkf
Initialization to compute the $Q_k$'s:
        qkf = exp(−dels + $k$ * ln(dels) − ln($k$ + 3/2))
        qkb = qkf
Compute the incomplete beta function associated with the $P_k$:
        pbetaf = beta distribution at $(y, a, b)$
        pbetab = pbetaf
Compute the incomplete beta function associated with the $Q_k$:
        qbetaf = beta distribution at $(y, c, b)$
        qbetab = qbetaf
Initialization to compute the incomplete beta functions associated with the $P_i$'s
recursively:
        pgamf = $\exp(\ln\Gamma(a+b-1)-\ln\Gamma(a)-\ln\Gamma(b)+(a-1)*\ln(y)+b*\ln(1-y))$
        pgamb = pgamf $* y * (a + b - 1)/a$
Initialization to compute the incomplete beta functions associated with the $Q_i$'s
recursively:
        qgamf=$\exp(\ln\Gamma(c+b-1)-\ln\Gamma(c)-\ln\Gamma(b)+(c-1)*\ln(y)+b*\ln(1-y))$
        qgamb=qgamf $* y * (c+b-1)/c$
```

```
Compute the remainder of the Poisson probabilities:
    rempois = 1 − pkf
    sum = pkf ∗ pbetaf + del ∗ qkf ∗ qbetaf/$\sqrt{2}$
    $i = 1$
1   pgamf = pgamf ∗ $y$ ∗ $(a + b + i - 2)/(a + i - 1)$
    pbetaf = pbetaf − pgamf
    pkf = pkf ∗ dels/$(k + i)$
    ptermf = pkf ∗ pbetaf
    qgamf = qgamf ∗ $y$ ∗ $(c + b + i - 2)/(c + i - 1)$
    qbetaf = qbetaf − qgamf
    qkf = qkf ∗ dels/$(k + i + 1/2)$
    qtermf = qkf ∗ qbetaf
    sum = sum + ptermf + delta ∗ qtermf/$\sqrt{2}$
    error = rempois ∗ (1 + abs(delta)/2)/2 ∗ pbetaf
    rempois = rempois − pkf
Do forward and backward computations $k$ times or until convergence:
    if i > k, then
    if (error ⩽ errtol or $i$ > maxitr) goto 2
    $i = i + 1$
    goto 1
    else
    pgamb = pgamb ∗ $(a - i + 1)/(y * (a + b - i))$
    pbetab = pbetab + pgamb
    pkb = pkb ∗ $(k - i + 1)$/dels
    ptermb = ptermb ∗ pbetab
    qgamb = qgamb ∗ $(c - i + 1)/(y * (c + b - i))$
    qbetab = qbetab + qgamb
    qkb = qkb ∗ $(k - i + 3/2)$/dels
    qtermb = qkb ∗ qbetab
    sum = sum + ptermb + delta ∗ qtermb/$\sqrt{2}$
    rempois = rempois − pkb
    if rempois ⩽ errtol or $i$ ⩾ maxitr goto 2
    $i = i + 1$
    goto 1
2   cdf = sum/2 + pnorm
If $t$ is negative, then set:
    cdf = 1 − cdf
    end
```

Algorithm 7.3. Noncentral chisquare distribution function.

Input:

y=the value at which $p(\chi_n^2(\lambda) \leqslant y)$ is to be computed, $(0 < y < \infty)$

n = degrees of freedom, $(n > 0)$

```
        $\lambda$ = noncentrality parameter, ($\lambda > 0$)
        errtol = error tolerance
        maxitr = maximum number of iterations

Output:
        cdf = $P(\chi_n^2(\lambda) \leqslant y)$
Set:
        $x = y/2$
        del = $\lambda/2$
        $k$ = integral part of ($\lambda/2$)
        $a = n/2 + k$
Compute the gamma distribution function using (4.5) at ($x, a$), and assign it to
"gamkf" and "gamkb" so that they can be called later for forward as well as
backward computations:
        gamkf = gamma distribution function at ($x, a$)
        gamkb = gamkf
        if $\lambda = 0$, set cdf = gamkf, return
Compute the Poisson probability at ($k$, del) and assign it to "poikf" and "poikb"
so that they can be used as initial values for forward and backward recursions:
        poikf = exp(−del + $k$ * ln(del) − ln $\Gamma(k + 1)$)
        poikb = poikf
"xtermf" is an initialization to compute the second term in (4.3) recursively:
        xtermf = exp(($a$ − 1) * ln($x$) − $x$ − ln $\Gamma(a)$)
"xtermb" is an initialization to compute the second term in (4.4) recursively:
        xtermb = xtermf * $x/a$
        sum = poikf * gamkf
        remain = 1 − poikf
        $i = 1$
1       xtermf = xtermf * $x/(a + i - 1)$
        gamkf = gamkf − xtermf
        poikf = poikf * del/($k + i$)
        sum = sum + poikf * gamkf
        error = remain * gamkf
        remain = remain − poikf
Do forward and backward computations k times or until convergence:
        if i > k then
        if error $\leqslant$ errtol or $i$ > maxitr goto 2
        $i = i + 1$
        goto 1
        else
        xtermb = xtermb * ($a - i + 1$)/$x$
        gamkb = gamkb + xtermb
        poikb = poikb * ($k - i + 1$)/del
        sum = sum + gamkb * poikb
        remain = remain − poikb
        if remain $\leqslant$ errtol or $i >$ maxitr goto 2
        $i = i + 1$
        goto 1
        **end if**
2       cdf = sum
        **end**
```

Acknowledgements

The authors are grateful to two reviewers for their comments and suggestions.

References

Abramovitz, M., Stegun, I.A., 1964. Handbook of Mathematical Functions. U.S. Government Printing Office, Washington, DC.

Benton, D., Krishnamoorthy, K., Mathew, T., 2002. Inferences in multivariate-univariate calibration problems. The Statistician (JRSS-D), to appear.

Chattamvelli, R., Shanmugam, R., 1997. Computing the noncentral beta distribution function. Appl. Statist. 46, 146–156.

Craig, C.C., 1942. Note on the distribution of noncentral t with an application. Ann. Math. Statist. 17, 224 –228.

Ding, C.G., 1992. Computing the noncentral χ^2 distribution function. Appl. Statist. 41, 478–482.

Ding, C.G., Bargmann, R.E., 1991. Evaluation of the distribution of the square of the sample multiple-correlation coefficient. Appl. Statist. 40, 195–236.

Frick, H., 1990. A Remark on Algorithm AS-226—computing non-central beta-probabilities. Appl. Statist. 9, 311–312.

Guenther, W.C., 1978. Evaluation of probabilities for the noncentral distributions and the difference of two t variables with a desk calculator. J. Statist. Comput. Simulation 6, 199–206.

Hall, I.J., 1984. Approximate one-sided tolerance limits for the difference or sum of two independent normal variates. J. Quality Tech. 16, 15–19.

Helstrom, C.W., Ritcey, J.A., 1985. Evaluation of the noncentral F distribution by numerical contour integration. SIAM J. Sci. Statist. Comput. 6, 505–514.

Kramer, K.H., 1963. Tables for constructing confidence limits on the multiple correlation coefficient. J. Amer. Statist. Assoc. 58, 1082–1085.

Krishnamoorthy, K., Mathew, T., 1999. Comparison of approximation methods for computing tolerance factors for a multivariate normal population. Technometrics 41, 234–249.

Lenth, R.V., 1987. Computing noncentral beta probabilities. Appl. Statist. 36, 241–244.

Lenth, R.V., 1989. Cumulative distribution function of the noncentral t distribution. Appl. Statist. 38, 185–189.

Muirhead, R.J., 1982. Aspects of multivariate statistical theory. Wiley, New York.

Neter, J., Kutner, M.H., Nachtsheim, C.J., Wasserman, W., 1996. Applied Linear Regression Models. Irwin, Chicago.

Posten, H.O., 1993. An effective algorithm for the noncentral beta distribution function. Amer. Statist. 47, 129–131.

Reiser, B.J., Guttman, I., 1986. Statistical inference for $Pr(Y < X)$: the normal case. Technometrics 28, 253 –257.

Tiwari, R.C., Yang, J., 1997. An efficient recursive algorithm for computing the distribution function and noncentrality parameter of the noncentral F distribution. Appl. Statist. 46, 408–413.

Vangel, M.G., 1992. New methods for one-sided tolerance limits for a one-way balanced random-effects ANOVA model. Technometrics 34, 176–185.

Wang, M., Kennedy, W.J., 1994. Self-validating computations of probabilities for selected central and non-central probability functions. J. Amer. Statist. Assoc. 89, 878–887.

Wang, M., Kennedy, W.J., 1995. A self-validating numerical method for computation of central and non-central F probabilities and percentiles. Statist. Comput. 5, 155–163.