NUMERICAL COMPUTATION OF CRITICAL VALUES FOR MULTIPLE COMPARISON PROBLEMS

Alan Genz, Washington State University, Frank Bretz, University of Hannover
Alan Genz, Mathematics Department, WSU, Pullman, WA 99164-3113

Key Words: Multivariate-t, Critical Value, Multiple Comparison

1 Introduction

We consider a general linear model with fixed effects:

$$Y = X\boldsymbol{\beta} + \boldsymbol{\epsilon}.$$

We assume that we are given an $N \times 1$ data vector Y, $N \times p$ design matrix X, with unknown $p \times 1$ parameter vector $\boldsymbol{\beta}$, and an $N \times 1$ error vector $\boldsymbol{\epsilon}$, with i.i.d. normally distributed components with unknown variance σ^2. The setup for multiple comparison problems (see Hochberg and Tamhane, 1987 and Hsu, 1992, 1996, Somerville, 1997, 1998, Stoline, 1981) provides an $m \times p$ *comparison (or contrast) matrix* C. The covariance matrix for the multiple comparison problem is then $\boldsymbol{\Sigma} = C(X^tX)^{-1}C^t$, an $m \times m$ positive semi-definite matrix. The resulting basic numerical problem that is the focus for this paper is the determination of confidence intervals (CI's) for

$$x_i = \sum_{j=1}^{p} c_{i,j}\beta_j, \quad i = 1, \ldots, m.$$

The distribution for $\mathbf{x}$ is an m–variate Student's t (MVT), with covariance matrix $\boldsymbol{\Sigma}$ and degrees of freedom $\nu = N - rank(X)$. We use the Dunnett (1954) definition of the MVT distribution given by

$$\mathbf{T}(\mathbf{a}, \mathbf{b}, \boldsymbol{\Sigma}, \nu) = \frac{2^{1-\frac{\nu}{2}}}{\Gamma(\frac{\nu}{2})} \int_0^\infty s^{\nu-1} e^{-\frac{s^2}{2}} \boldsymbol{\Phi}(\frac{s\mathbf{a}}{\sqrt{\nu}}, \frac{s\mathbf{b}}{\sqrt{\nu}}, \boldsymbol{\Sigma}) ds,$$

where the multivariate normal distribution function

$$\boldsymbol{\Phi}(\mathbf{a}, \mathbf{b}, \boldsymbol{\Sigma}) = \frac{1}{\sqrt{|\boldsymbol{\Sigma}|(2\pi)^m}} \int_{a_1}^{b_1} \int_{a_2}^{b_2} \cdots \int_{a_m}^{b_m} e^{-\frac{1}{2}\mathbf{x}^t\boldsymbol{\Sigma}^{-1}\mathbf{x}} d\mathbf{x},$$

$\mathbf{x} = (x_1, x_2, ..., x_m)^t$, $-\infty \le a_i < b_i \le \infty$ for all i, and $\boldsymbol{\Sigma}$ is a positive semi-definite symmetric $m \times m$ matrix. Efficient and robust numerical software is now available for MVT distribution computations for $1 \le m \le 20$ (see Genz and Bretz, 1999, 2000, also Genz and Kwong, 1999, for how to handle singular $\boldsymbol{\Sigma}$'s). The final input for the CI determination problem is a confidence level α. The actual numerical problem consists of finding the *critical value* t_α where $P(t_\alpha) = 1 - \alpha$, with

- $P(t) = \mathbf{T}(-\infty, \mathbf{t}, \nu, \boldsymbol{\Sigma})$ for one-sided CI's, **or**
- $P(t) = \mathbf{T}(\ -\mathbf{t}, \mathbf{t}, \nu, \boldsymbol{\Sigma})$ for two-sided CI's.

Here $\mathbf{t} = (t, ..., t)^t$ and $\boldsymbol{\infty} = (\infty, ..., \infty)^t$. The numerical problem is therefore a combined problem of using an appropriate numerical optimization method to determine t_α with an efficient numerical integration method for evaluating $P(t)$.

2 Numerical Optimization

The numerical optimization-integration methods considered in this paper use the function

$$h(t) = P(t) - (1 - \alpha).$$

These methods involve finding t_α, the point where $h(t_\alpha) = 0$, using a numerical optimization method. But $h(t)$ is often expensive to compute using numerical integration, particularly for large m, so a numerical optimization method that requires only a few iterations is needed. This need can be satisfied if we combine a method for getting good starting points for the optimization method, with an optimization method that converges rapidly. An additional complication that arises with the combined numerical optimization-integration method is the presence of numerical integration errors, which must be controlled along with with the numerical optimization errors. These issues are discussed in the next three subsections.

2.1 Starting Interval Selection

Let $A_j(t) = \{x_j : x_j \le t\}$ for the one-sided cases, or $A_j = \{x_j : |x_j| \le t\}$ for the two-sided cases. We let

$$S_1(t) = \sum_{j=1}^{m} Prob(A_j^c(t)),$$

where $A_j^c(t)$ is the compliment of the set $A_j(t)$. The Bonferroni bound for $P(t)$ (see Hsu, 1996) is

$$L'(t) = 1 - S_1(t) \le P(t).$$

A simple upper bound for $P(t)$ is

$$P(t) \le 1 - \min_j Prob(A_j^c(t)) = U'(t).$$

Both of these bounds require only 1-dimensional distribution values. If t'_a and t'_b are determined by solving $U'(t) = 1-\alpha$ and $L'(t) = 1-\alpha$, respectively, then $t_\alpha \in [t'_a, t'_b]$. This bounding interval for t_α can be found directly using the appropriate 1-dimensional inverse distribution function. For example, with the two-sided case,

$$[t'_a, t'_b] = [t_\nu^{-1}(1 - \frac{\alpha}{2}), t_\nu^{-1}(1 - \frac{\alpha}{2m})],$$

where $t_\nu(u) = \frac{\Gamma(\frac{\nu+1}{2})}{\Gamma(\frac{\nu}{2})\sqrt{\nu\pi}} \int_{-\infty}^{u} (1 + \frac{s^2}{\nu})^{-\frac{\nu+1}{2}} ds$. Shorter intervals can be found using bivariate distribution values (Dunnett and Sobel, 1954) if a modified Bonferroni bound (Dawson and Sankoff, 1967) is combined with the Hunter-Worsley bound. These bounds are described in the book by Hsu (1966, Appendix A). If we define $S_2(t)$ by

$$S_2(t) = \sum_{j<i} Prob(A_j^c(t) \cap A_i^c(t)),$$

then the modified Bonferroni bounds and Hunter-Worsley guarantee that

$$\begin{aligned} L(t) &= 1 - S_1(t) + \sum_{(i,j)\in T^*} Prob(A_j^c(t) \cap A_i^c(t)) \\ &\le P(t) \le 1 - 2\frac{S_1(t) - S_2(t)/k}{k+1} = U(t). \end{aligned}$$

where $k = 1 + \lfloor 2S_2(t)/S_1(t) \rfloor$ and T^* is maximal spanning tree for complete graph of order m with edge weights $Prob(A_j^c(t) \cap A_i^c(t))$. If $U(t_a) = 1 - \alpha$ and $L(t_b) = 1-\alpha$ then $t'_a \le t_a \le t_\alpha \le t_b \le t'_b$. Starting with $[t'_a, t'_b]$, we can use numerical optimization, applied to $L(t)$, to determine t_b, then use numerical optimization, applied to $U(t)$ starting with $[t'_a, t_b]$, to determine t_a.

2.2 Choice of Optimization Method

A primary goal for the selection of an optimization method is to find a method that, given good starting points, requires only a few iterations for a large class of problems, so it would be desirable to use a second order method like Newton's method to find t_α. The Newton iteration method for improving an estimate for t_c for t_α, successively replaces t_c by $t_c - h(t_c)/h'(t_c)$. This method requires values for both $h(t)$ and $h'(t)$. If we make a simple change of variable $\mathbf{x} = t\mathbf{y}$ in the detailed expression for $h(t)$ determined from our definition of the MVT distribution function, we have (for the two-sided case)

$$h(t) = \alpha - 1 + t^m \frac{2^{1-\frac{\nu}{2}} |\mathbf{\Sigma}|^{-\frac{1}{2}}}{\Gamma(\frac{\nu}{2})(2\pi)^{\frac{m}{2}}} \int_0^\infty s^{\nu-1} e^{-\frac{s^2}{2}} \int_{-\frac{s}{\sqrt{\nu}}}^{\frac{s}{\sqrt{\nu}}} e^{-\frac{t^2 \mathbf{y}^t \mathbf{\Sigma}^{-1} \mathbf{y}}{2}} d\mathbf{y} ds.$$

Differentiating $h(t)$, and then changing the variables back to $\mathbf{x}$, we find

$$h'(t) = \frac{1}{t}(mP(t) - H(t)),$$

where $H(t)$ is given by

$$\frac{2^{1-\frac{\nu}{2}} |\mathbf{\Sigma}|^{-\frac{1}{2}}}{\Gamma(\frac{\nu}{2})(2\pi)^{\frac{m}{2}}} \int_0^\infty s^{\nu-1} e^{-\frac{s^2}{2}} \int_{-\frac{st}{\sqrt{\nu}}}^{\frac{st}{\sqrt{\nu}}} \mathbf{x}^t \mathbf{\Sigma}^{-1} \mathbf{x}\, e^{-\frac{\mathbf{x}^t \mathbf{\Sigma}^{-1} \mathbf{x}}{2}} d\mathbf{x} ds.$$

For the one-sided case, the lower limits for the inner integral are all $-\infty$. In either case, $H(t)$ may be computed with only a little extra work during the computation of $P(t)$.

Given a starting interval $[t_a, t_b]$, with $t_\alpha \in [t_a, t_b]$. and a required error tolerance τ for our final estimate of t_α, we let $t_c = \frac{t_a + t_b}{2}$ and use a Newton algorithm that repeats:
if $(t_b - t_a) > 2\tau$ and $|h(t_c)| > \tau |h'(t_c)|$, **then**

a) if $h(t_c) < 0$, set $(t_a, h(t_a)) = (t_c, h(t_c))$, otherwise set $(t_b, h(t_b)) = (t_c, h(t_c))$;

b) set $t_c = t_c - h(t_c)/h'(t_c)$;

else stop and output $t_\alpha \approx \frac{t_a + t_b}{2}$ or $t_\alpha \approx t_c$.

We also investigated the use of Secant-like methods for solving $h(t) = 0$. Some preliminary tests showed that the simple Secant method is not suitable for many problems because $|h'(t)|$ is sometimes very small near t_α (particularly when α is small), and this can result in divergence unless a very good starting value is available. Various bisection-Secant hybrid methods were considered and after some experiments, the "Pegasus" method (see Ralston and Rabinowitz, 1978) was selected. This method has asymptotic order of convergence similar to that of the Secant method and, at each iteration it provides a bracketing interval for t_α. The Pegasus method that we finally implemented (starting with τ and $[t_a, t_b]$) initially sets $t_c = \frac{t_a + t_b}{2}$. If $h(t_c) < 0$,

we set $(t_a, h(t_a)) = (t_c, h(t_c))$; otherwise we set $(t_b, h(t_b)) = (t_c, h(t_c))$. Our basic iteration repeats:
if $(t_b - t_a) > 2\tau$ and $|h(t_c)| > \tau|\frac{h(t_b)-h(t_a)}{t_b-t_a}|$, **then**

a) compute $t_c = t_b - h(t_b)(t_b - t_a)/(h(t_b) - h(t_a))$;

b) if $h(t_c) < 0$, set $(t_a, h(t_a)) = (t_c, h(t_c))$, otherwise set $h(t_a) = h(t_a)h(t_b)/(h(t_b) + h(t_c))$ and $(t_b, h(t_b)) = (t_c, h(t_c))$

else stop and output $t_\alpha \approx \frac{t_a+t_b}{2}$ or $t_\alpha \approx t_c$.
The Pegasus method is the same as the linearly convergent False-Position method except for the h_a modification at step b), which improves the approximate order of convergence to 1.64.

2.3 Error Control

When numerical integration is used to evaluate $h(t)$, what is actually computed is $\hat{h}(t) = h(t) + \epsilon_I$, where ϵ_I is the numerical integration error. This error can, in principle, be made arbitrarily small, but at the expense of more work (computer time). Let $\hat{t} = t_\alpha + \epsilon_t$ be an approximation to t_α with error ϵ_t. What is actually computed at each step in a combined numerical optimization-integration algorithm is

$$\hat{h}(\hat{t}) = h(\hat{t}) + \epsilon_I \approx \epsilon_t h'(t_\alpha) + \epsilon_I, \qquad (1)$$

for sufficiently small $|\epsilon_t|$. In order to balance the numerical integration and optimization errors, an estimate for $|h'(t_\alpha)|$ is needed. This quantity does not need to be very accurate, so a simple difference quotient approximation, in the form

$$h'(t_\alpha) \approx \hat{h}' = \frac{\hat{h}(t_b) - \hat{h}(t_a)}{t_b - t_a},$$

can be determined using quantities from the Pegasus method iterations. If the Newton method is being used, then an accurate value of $h'(t_\alpha)$ is already available.

Given $\hat{h}'$ and τ (the desired error tolerance for t_α), the numerical integration error tolerance must be set at a level at least as small as $\hat{h}'\tau$. Otherwise the numerical integration errors may dominate the total error in $\hat{h}(\hat{t})$ and it will not be possible to reliably determine when the optimization method has converged. The strategy used for the example test results in the next section was to set the error tolerance for the numerical integration at $\tau|\hat{h}'|$.

3 Examples

The following examples illustrate the use of the numerical optimization-integration algorithms described in the previous section. Work is measured in terms of the number of density function evaluations needed for the integration to compute $P(t)$. The basic starting interval $[t_a, t_b]$ was computed using the Pegasus method; the cost for this is minimal because only univariate and bivariate $\mathbf{T}$ values are required. The initial bracketing interval for t_α is given in the first row in each table, and the work for that row is the work to compute the initial $h(t_a)$, $h(t_c)$ and $h(t_b)$ values (with only $h(t_c)$ required to start the Newton iteration). Subsequent rows show results of the iterations necessary to determine t_α to the specified accuracy.

Our first example used data for starch thickness taken from Hsu and Nelson (1998). The desired CI's were for two-sided comparisons versus a control, with $P(t) = \mathbf{T}(-\mathbf{t}, \mathbf{t}, \mathbf{\Sigma}, 86)$, and

$$\mathbf{\Sigma} = \begin{bmatrix} 1 & - & - & - & - & - \\ .3958 & 1 & - & - & - & - \\ .5677 & .4936 & 1 & - & - & - \\ .5468 & .4621 & .7598 & 1 & - & - \\ .5140 & .4488 & .7675 & .6930 & 1 & - \\ .5505 & .4922 & .8651 & .7738 & .7915 & 1 \end{bmatrix}.$$

For $\alpha = .1$, we computed $[t'_a, t'_b] = [1.663, 2.442]$ and $[t_a, t_b] = [2.116, 2.324]$. The following numerical optimization-integration results were obtained.

Pegasus Method Results for $\tau = 0.001$

t_a, t_c, t_b	$h(t_a), h(t_c), h(t_b)$	$\hat{h}'$, Work
2.220, 2.264, 2.324	-.01, .0005, .01	.22, 39952
2.220, 2.262, 2.264	-.01, .00001, .0005	.22, 51152

Newton Method Results for $\tau = 0.001$

t_a, t_c, t_b	$h(t_a), h(t_c), h(t_b)$	$\hat{h}'$, Work
2.116, 2.220, 2.324	, -.01,	.26, 11200
2.220, 2.261, 2.324	-.01, -.0002,	.24, 28752

For $\alpha = .05$, we computed $[t'_a, t'_b] = [1.988, 2.701]$, $[t_a, t_b] = [2.429, 2.606]$ and:

Pegasus Method Results for $\tau = 0.001$

t_a, t_c, t_b	$h(t_a), h(t_c), h(t_b)$	$\hat{h}'$, Work
2.518,2.561,2.606	-.005, .0002, .006	.12, 146432
2.518,2.559,2.561	-.005,.00004,.0002	.12, 187984

Newton Method Results for $\tau = 0.001$

t_a, t_c, t_b	$h(t_a), h(t_c), h(t_b)$	$\hat{h}'$, Work
2.429,2.518,2.606	, -.005,	.14, 41552
2.518,2.557,2.606	-.005,-.0002,	.13, 104880
2.557,2.559,2.606	-.0002,.00001,	.12, 168208

Our second example is an all-pairwise comparisons example based on a one-way ANOVA design with sample sizes $n_i = 20, 3, 3, 15$ (see Westfall et.al., 1999). We need two sided CI's with

$P(t) = \mathbf{T}(-\mathbf{t}, \mathbf{t}, \mathbf{\Sigma}, 37)$, given $\mathbf{\Sigma} =$

$$\begin{bmatrix} 1 & - & - & - & - & - \\ .1304 & 1 & - & - & - & - \\ .2364 & .2364 & 1 & - & - & - \\ -.6594 & .6594 & 0 & 1 & - & - \\ -.8513 & 0 & .3086 & .6455 & 1 & - \\ 0 & -.8513 & .3086 & -.6455 & .1667 & 1 \end{bmatrix}.$$

For $\alpha = .1$, we computed $[t'_a, t'_b] = [1.687, 2.508]$, $[t_a, t_b] = [2.265, 2.383]$ and:

Pegasus Method Results for $\tau = 0.001$

t_a, t_c, t_b	$h(t_a), h(t_c), h(t_b)$	$\hat{h}'$, Work
2.324,2.339,2.383	-.003,.0003,.009	.21, 86144
2.324,2.337,2.339	-.003,-.0002,.0003	.21, 127696

Newton Method Results for $\tau = 0.001$

t_a, t_c, t_b	$h(t_a), h(t_c), h(t_b)$	$\hat{h}'$, Work
2.265,2.324,2.383	, -.003,	.25, 41552
2.324,2.338,2.383	-.003,-.00006,	.21, 68592

For $\alpha = .05$, we computed $[t'_a, t'_b] = [2.050, 2.788]$, $[t_a, t_b] = [2.026, 2.684]$ and:

Pegasus Method Results for $\tau = 0.001$

t_a, t_c, t_b	$h(t_a), h(t_c), h(t_b)$	$\hat{h}'$, Work
2.648,2.654,2.684	-.0006,-.00001,.003	.11, 255680

Newton Method Results for $\tau = 0.001$

t_a, t_c, t_b	$h(t_a), h(t_c), h(t_b)$	$\hat{h}'$, Work
2.612,2.648,2.684	, -.0006,	.13, 63328
2.648,2.654,2.684	-.0006,.00003,	.12, 159504

The results from these example tests demonstrate that the algorithms described in this paper provide feasible methods for computing t_α values for confidence intervals. Given good starting intervals determined from bivariate distribution values, the numerical optimzation based on the use of the Newton method is more efficient than optimization based on the Pegasus method. These conclusions are also supported by a variety of other examples that we have considered.

REFERENCES

- Dawson, D. and Sankoff, A. (1967), 'An Inequality for Probabilities', *Proc. AMS* **18**, pp. 504–507.
- Dunnett, C.W. and Sobel, M. (1954), 'A Bivariate Generalization of Student's t-Distribution, with Tables for Certain Special Cases' *Biometrika* **41**, pp. 153–169.
- Genz, A. and Bretz, F. (1999), 'Numerical Computation of the Multivariate t Probabilities with Application to Power Calculation of Multiple Contrasts', *J. Stat. Comput. Simul.***63**, pp. 361–378.
- Genz, A. and Bretz, F. (2000), 'Methods for the Computation of Multivariate t Probabilities', submitted and available from the website: www.sci.wsu.edu/math/faculty/genz/homepage.
- Genz, A. and Kwong, K.S. (1999), 'Numerical Evaluation of Singular Multivariate Normal Distributions', to appear in *Journal of Statistical Computation and Simulation.*
- Hochberg, Y., and Tamhane, A.C. (1987), *Multiple Comparison Procedures*, John Wiley and Sons, New York.
- Hsu, Jason C. (1996), *Multiple Comparisons*, Chapman and Hall, London.
- Hsu, Jason C. (1992), 'Simultaneous Inference in the General Linear Model', *J. Comput. Graph. Stat.* **1**, pp. 151–168.
- Hsu, Jason C. and Nelson, Barry (1998), 'Multiple Comparisons in the General Linear Model', *J. Comput. Graph. Stat.* **7**, pp. 23–41.
- Ralston, A., and Rabinowitz, P. (1978), *A First Course in Numerical Analysis*, McGraw-Hill, New York.
- Somerville, P.N. (1997, 'Multiple Testing and Simultaneous Confidence Intervals: Calculation of Constants' *Comp. Stat. & Data Analysis* **25**, pp. 217–223.
- Somerville, P.N. (1998), 'Numerical Computation of Multivariate Normal and Multivariate-t Probabilities Over Convex Regions', *J. Comput. Graph. Stat.* **7**, pp. 529–545.
- Stoline, M.R. (1981), 'The Status of Multiple Comparisons: Simultaneous Estimation of All Pairwise Comparisons in One-Way ANOVA Design', *The American Statistician* **35**, pp. 134–141.
- Tong, Y.L. (1990), *The Multivariate Normal Distribution*, Springer-Verlag, New York, New York.
- Westfall, P.H., Tobias, R.D., Rom D., and Wolfinger, R.D. (1999), *Multiple Comparisons and Multiple Tests using the SAS System*, SAS Institute Inc, Cary, NC.