SAS® Technical Report R-105

Computing MIVQUE0 Estimates of Variance Components

5984

SAS® SAS Institute Inc.

Computing MIVQUE0 Estimates of Variance Components

by

J.H. Goodnight

SAS® Technical Report R-105

ABSTRACT

This paper suggests computing methods for variance component estimation which are designed to handle large sample survey or animal breeding experiments where the random effects may have thousands of levels. The underlying theoretical technique is MIVQUE (with V=I) which has been shown to yield unbiased, locally best, admissible, and asymptotically consistent estimators.

SAS Institute Inc.
SAS Circle, Box 8000
Cary, NC 27512-8000

The correct bibliographic citation for this technical report is as follows: SAS Institute Inc., SAS® Technical Report R-105, Computing MIVQUEO Estimates of Variance Components, Cary, NC: SAS Institute Inc., 1978.

SAS® Technical Report R-105, Computing MIVQUEO Estimates of Variance Components

1. INTRODUCTION

In a recent paper, Hartley et al. (1978) suggest the use of Rao's (1971) MINQUE (with V=I) as a method of estimating variance components in large experimental or sample survey designs. The method is intuitively appealing because of its inherent optimality, consistency, and computability properties. In the main, their paper focuses on a computational technique for computing MINQUE (with V=I) estimates. Alternative computing techniques are suggested here which have been implemented in the VARCOMP procedure of the Statistical Analysis System (SAS), Barr et al. (1979).

Following Rao (1972) the technique described by Hartley et al. with their implied choice of a MINQUE norm can be better labeled as MIVQUE (with V=I) or MIVQUE (with 0 priors) or simply MIVQUE0. In what is to follow, the necessary computing formulas for MIVQUE0 will be derived from Rao's original work, and several computing methods will be described which use these basic formulas. There should be little difficulty in ascertaining that the resultant estimates from these methods are equivalent to Hartley et al's(1978) method.

2. THE MODEL

Following Searle (1971) the mixed model is represented as:

$$Y = X_o\beta_o + \sum_{i=1}^{k} X_i\beta_i + e \qquad (1)$$

where,

(1) Y is an n-vector of observations

(2) $X_o, X_1, \ldots, X_k$ are nxm_i known matrices

(3) β_o is a vector of fixed effects

(4) The vectors $\beta_i (i=1,\ldots,k)$ are assumed independent of each other and e and are distributed $N(0, \sigma_i^2 I_{m_i})$

(5) e is an n-vector assumed $N(0, \sigma_e^2 I)$

On making the above normality assumptions,

$$E(Y) = X_o \beta_o$$

$$\text{Var}(Y) = \sum_{i=1}^{k} X_i X_i' \sigma_i^2 + I \sigma_e^2$$

3. MIVQUEO EQUATIONS

$$\text{Letting } H = \sum_{i=1}^{k} X_i X_i' \gamma_i + I \tag{2}$$

where γ_i are apriori estimates of σ_i^2/σ_e^2, the locally best (at γ) MIVQUE estimates of the variance components can be found by solving the following system:

$$\begin{bmatrix} S_{11} & S_{12} & \cdots & S_{1\,k+1} \\ S_{21} & S_{22} & \cdots & S_{2\,k+1} \\ \vdots & & & \\ S_{k+1\,1} & S_{k+1\,2} & \cdots & S_{k+1\,k+1} \end{bmatrix} \begin{bmatrix} \hat{\sigma}_1^2 \\ \hat{\sigma}_2^2 \\ \vdots \\ \hat{\sigma}_e^2 \end{bmatrix} = \begin{bmatrix} T_1 \\ T_2 \\ \vdots \\ T_{k+1} \end{bmatrix} \tag{3}$$

For ease of notation define X_{k+1} to be an nxn identity matrix and define the matrix operator SSQ(A) to equal the sum of squares of the elements of A. Using this notation, the elements of (3) are:

$$S_{ij} = \text{SSQ}(X_i' R X_j)$$

and $T_i = \text{SSQ}(X_i' R Y)$

where $R = H^{-1} - H^{-1} X_o (X_o' H^{-1} X_o)^{-} X_o' H^{-1}$ (4)

When the elements of γ ($\gamma_1, \gamma_2, \ldots, \gamma_k$) are all zero, then (4) reduces to

$$R_o = I - X_o(X_o' X_o)^- X_o' \tag{5}$$

and the computational burden is reduced. In addition computing the elements in the (k+1)st row and column of the S matrix and the T_{k+1} element of the T vector in (3) is simplified since R_o is idempotent. These elements are:

$$S_{i\,k+1} = SSQ(X_i' R_o) = Tr(X_i' R_o R_o X_i) = Tr(X_i' R_o X_i) \quad \text{for} \quad i < k+1 \tag{6}$$

$$S_{k+1,k+1} = SSQ(R_o) = Tr(R_o R_o) = Tr(R_o) = n - Rank(X_o) \tag{7}$$

$$T_{k+1} = SSQ(R_o Y) = Tr(Y' R_o R_o Y) = Tr(Y' R_o Y) = Y' R_o Y \tag{8}$$

4. BASIC COMPUTING FORMULA

To summarize the above results and provide further computational results, MIVQUEO estimates may be computed as follows:

Step 1: Form the symmetric matrix:

$$\left[\begin{array}{cccc|c} X_o'X_o & X_o'X_1 & \cdots & X_o'X_k & X_o' Y \\ & X_1'X_1 & \cdots & X_1'X_k & X_1' Y \\ & & \ddots & \vdots & \\ & & & X_k'X_k & X_k' Y \\ & & & & Y'Y \end{array}\right] \tag{9}$$

Step 2: Apply any of the elimination methods...Gauss, Gauss-Jordan, Doolittle, or Cholesky (see for example Goodnight (1978)) -- to the entire matrix (9) by pivoting on each diagonal of $X_o'X_o$. This reduces the

elements below and to the right of $X_o'X_o$ to the following:

$$\left[\begin{array}{cccc|c} X_1'R_oX_1 & X_1'R_oX_2 & \cdots & X_1'R_oX_k & X_1'R_oY \\ & X_2'R_oX_2 & \cdots & X_2'R_oX_k & X_2'R_oY \\ & & \ddots & & \\ & & & X_k'R_oX_k & X_k'R_oY \\ & & & & Y'R_oY \end{array}\right] \tag{10}$$

In the process of mapping (9) to (10), the rank of X_o may be computed, (see Goodnight (1978)).

Step 3. Form the symmetric SSQ matrix from (10)

$$\left[\begin{array}{ccccc|c} SSQ(X_1'R_oX_1) & \cdots & SSQ(X_1'R_oX_k) & Tr(X_1'R_oX_1) & & SSQ(X_1'R_oY) \\ & \cdots & SSQ(X_2'R_oX_k) & Tr(X_2'R_oX_2) & & SSQ(X_2'R_oY) \\ & & \ddots & \vdots & & \vdots \\ & & & Tr(X_k'R_oX_k) & & SSQ(X_k'R_oY) \\ & & & & & Y'R_oY \end{array}\right] \tag{11}$$

Step 4. Reduce the left hand side of (11) to an identity using pivoting operations, and providing no linear dependencies are found, the MIVQUE0 estimates are obtained.

5. LARGE DESIGNS

When computer storage is not sufficient to hold the entire matrix given by (9), then a multi-pass technique can be used to compute (11). When multi-passing is needed, then two methods are considered. In the discussion of both methods the terminology "fixed rows" refers to the first

m_o rows of (9). The remaining rows are referred to as "random rows." The last row containing only Y'Y is computed and adjusted separately.

Method 1. When all of the fixed rows of (9) can be stored with enough storage left to store one or more of the random rows then the following steps may be taken to compute (11).

Step 1. Compute and store all fixed rows and as many random rows as possible. Also compute Y'Y.

Step 2. Adjust each fixed row for the fixed rows preceeding it using the Cholesky operator. Also adjust Y'Y for the fixed effects and compute rank(X_o) in this step.

Step 3. Adjust each random row in core for the fixed effects. Then square each element of the adjusted random rows and add it to the appropriate element of (11). If an element of a random row is a diagonal element of (9), add it to the appropriate element in the k+1 column of (9).

Step 4. If all random rows have been computed, go to Step 5. Otherwise, re-initiatize the random rows storage locations, assign the next set of random rows to be computed in this storage area, and compute and store this next set of random rows. Go to Step 3.

Step 5. Store n-rank(X_o) and $Y'R_oY$ in (11) and solve the system for the MIVQUE0 estimates.

The virtue of implementing Method 1 is two fold. First, for a large class of designs the only fixed effect is the intercept; thus only one fixed row is held in core. Secondly, no auxiliary storage is used since once a set of random rows is computed, adjusted, and the elements squared and stored they are no longer needed.

Method 2. When all of the fixed rows of (9) cannot be stored, then auxillary storage must be used to compute and adjust (9) and the following steps may be taken.

Step 1. Multipass the original data set, computing at each pass as many rows of (9) as will fit in storage. After each pass, store the rows computed in auxillary storage. Also compute Y'Y during the first data pass. Once all rows have been computed and written to auxillary storage, set the current row designator M equal to 1.

Step 2. Leaving space for one additional row of (9) read and store (beginning with row M) as many fixed rows as is possible from auxiliary storage. Adjust these rows for each preceeding row using a Cholesky algorithim. Also adjust Y'Y. If all of the fixed rows have been read then go to Step 4. Otherwise:

Step 3. Denote by M the number of the next row on auxiliary storage and read each remaining row of (9) from auxiliary storage one at a time. Adjust each of these rows for the fixed rows in storage and rewrite the row to auxiliary storage. Go to Step 2.

Step 4. Read each remaining row of (9) one at a time from auxiliary storage, adjust it for the fixed rows in storage, square each element

and add to the appropriate element of (11). Add any adjusted diagonal elements of (9) to the appropriate element in the(k+1)st column of (9).

6. COMPUTING SPEED

For most models as specified by (1), the random incidence matrices, X_i will consist of rows whose elements are all zeros except for a single element which has the value one. Thus the computation of the random by random portion of (9) can be accomplished by k(k+1)/2 additions per observation. The computing time needed for solving (11) to get the MIVQUE0 estimates is trivial compared to the time needed to compute (11) regardless of the number of random effects. The number of rows in (9), and the number of observations are the critical factors determining computing speed.

To demonstrate the computing speed for large designs several two way crossed models of the form:

$$y = \mu + \alpha_i + \beta_j + (\alpha\beta)_{ij} + \varepsilon \qquad (12)$$

were run. In (12) μ is fixed, and α, β, $(\alpha\beta)$, and ε are random. This model (12) was chosen because of the ease in which large numbers of levels can be generated for $(\alpha\beta)$, a critical factor in computing speed. For each set of data generated,there was only one observation per cell, except for those cells involving α_1 which contained two observations. The problem size below, represents the number of levels of α and β. For a 10x10 there are 10 + 10 + 100 randon rows in (9). Timings for several different region sizes are shown.

Computing Speeds on an IBM 370/168
with Virtual Storage

Problem size	Number of random rows	Number of observations	CPU Time in Seconds			
			200K region	300K region	400K region	500K region
10 x10	120	110	.44			
20 x 20	440	420	1.94			
30 x 30	960	930	11.01	7.66	7.04	
40 x 40	1680	1640		28.75	23.09	20.18
50 x 50	2600	2550			65.60	53.96

The largest problem tested to date involved 23 random effects with 30,491 random levels. It ran in 221 CPU seconds in a region of 900 K.

7. CONCLUSION

For large scale experiments, MIVQUE0 affords the unbeatable combination of unbiasedness, admissibility, consistency and computational efficiency. None of the other methods currently competing in this arena (see Searle (1978)) can approach the computational efficiency achievable by MIVQUE0.

REFERENCES

Barr, A. J., Goodnight, J. H., Sall, J. P., and Helwig, J. T. (1979). SAS User's Guide, SAS Institute, P. O. Box 10066, Raleigh, N. C. 27605.

Goodnight, J. H. (1978). The Sweep Operator: Its Importance in Statistical Computing. Proceedings of the Computer Science and Statistics Eleventh Annual Symposium on the Interface, 218-229, Institute of Statistics, N. C. State University, Raleigh, N. C.

Hartley, H. O., Rao, J. N. K., and LaMotte, L. R. (1978). A Simple Synthesis-Based Method of Variance Component Estimation. Biometrics 34, 233-243.

Rao, C. R. (1971). Estimation of Variance and Covariance Components-Minque Theory. Journal of Multivariate Analysis 1, 257-275.

Rac, C. R. (1972). Estimation of Variance and Covariance Components in Linear Models. JASA 67, 112-115.

Searle, S. R. (1971). Topics in Variance Component Estimation. Biometrics 27, 1-72.

Searle, S. R. (1978). A Summary of Recently Developed Methods of Estimating Variance Components. Proceedings of the Computer Science and Statistics Eleventh Annual Symposium on the Interface, 64-69, Instatute of Statistics, N. C. State University, Raleigh, N. C.

Computing MIVQUE0 Estimates of Variance Components

J. H. Goodnight
SAS Institute, P. O. Box 10066
Raleigh, North Carolina 27605

SUMMARY

This paper suggests computing methods for variance component estimation which are designed to handle large sample survey or animal breeding experiments where the random effects may have thousands of levels. The underlying theoretical technique is MIVQUE (with V=I) which has been shown to yield unbiased, locally best, admissible, and asymptotically consistent estimators.

Key Words:

Variance Component Estimation; MIVQUE; MINQUE; Computational efficiency.

Index