Taylor & Francis
Taylor & Francis Group

Communications in Statistics - Simulation and Computation

ISSN: 0361-0918 (Print) 1532-4141 (Online) Journal homepage: http://www.tandfonline.com/loi/lssp20

Sparse matrices, and the estimation of variance components by likelihood methods

William H. Fellner

To cite this article: William H. Fellner (1987) Sparse matrices, and the estimation of variance components by likelihood methods, Communications in Statistics - Simulation and Computation, 16:2, 439-463, DOI: 10.1080/03610918708812599

To link to this article: http://dx.doi.org/10.1080/03610918708812599

Published online: 27 Jun 2007.

COMMUN. STATIST. -SIMULA., 16(2), 439-463 (1987)

SPARSE MATRICES, AND THE ESTIMATION OF VARIANCE COMPONENTS BY LIKELIHOOD METHODS

William H. Fellner
E. I. du Pont de Nemours and Company
Wilmington, Delaware

Key Words and Phrases: restricted maximum likelihood; analysis of variance; mixed models; analysis of covariance; algorithms.

ABSTRACT

It is generally considered that analysis of variance by maximum likelihood or its variants is computationally impractical, despite existing techniques for reducing computational effect per iteration and for reducing the number of iterations to convergence. This paper shows that a major reduction in the overall computational effort can be achieved through the use of sparse-matrix algorithms that take advantage of the factorial designs that characterize most applications of large analysis-of-variance problems. In this paper, an algebraic structure for factorial designs is developed. Through this structure, it is shown that the required computations can be arranged so that sparse-matrix methods result in greatly reduced storage and time requirements.

1. INTRODUCTION

Estimation of variance components by maximum-likelihood or restricted maximum-likelihood (Patterson and Thompson 1971) is generally thought to be attractive from a statistical point of view, but computationally impractical. This is due to the fact that iterative calculations on very large matrices are generally required. Hemmerle and Hartley (1973) have shown how the computational load per iteration could be greatly reduced by saving calculations made on previous iterations. Jennrich and Sampson (1976) have concentrated on reducing the number of iterations necessary for satisfactory convergence. Despite such efforts, maximum-likelihood methods remain sufficiently costly that they are not commonly used.

This paper considers a third aspect of reducing the computational burden. This is the fact that the factorial designs which characterize most applications typically result in very sparse matrices. By designing computer algorithms specifically for such designs, large cost-reductions can be achieved.

Expositions of the relevant sparse-matrix techniques are found in George and Liu (1981, Chapter 5), and Eisenstat et al (1981). Therefore, the major thrust of this paper is to show how these techniques can be effectively applied to the variance-component estimation problem.

In this paper, the methods of maximum-likelihood and restricted maximum-likelihood will be explicitly considered. A review of these and related methods can be found in the paper by Harville (1977).

2. BACKGROUND

Consider the linear model

$$y = X\alpha + Z b + e$$

where y is an n by 1 vector of observations, X is a full-rank n by

p known matrix, α is a p by 1 vector of fixed effects, Z is an n by q known matrix, b is a q by 1 vector of random effects, and e is an n by 1 vector of random errors. Assume that b and e are jointly normally distributed with

$$E(b) = 0,\ E(e) = 0,\ Cov(b,e) = 0,$$

and

$$Var(b) = D = diag\left[\ \sigma_1^2\ I_1,\ \ldots,\ \sigma_c^2\ I_c\ \right],$$

where I_i is the order-q_i identity matrix, with

$$q = q_1 + \ldots + q_c;$$

and

$$Var(e) = R = \sigma_{c+1}^2\ I_{c+1},$$

where I_{c+1} is the order-n identity matrix.

It is convenient to partition Z and b analogously to D:

$$Z = \left[\ Z_1\ \ldots\ Z_c\ \right]$$

$$b' = \left[\ b_1'\ \ldots\ b_c'\ \right]$$

Let β be the realized value of b, partitioned similarly.

Given $\sigma_1^2,\ \ldots,\ \sigma_{c+1}^2$, the best linear unbiased estimates, $\hat{\alpha}$ and $\hat{\beta}$, of α and β are the solution to the system of equations (Henderson 1963):

$$C\begin{bmatrix}\hat{\beta}\\ \hat{\alpha}\end{bmatrix} = \begin{bmatrix} Z & X \\ D^{-1/2}R^{1/2} & 0\end{bmatrix}\begin{bmatrix}\hat{\beta}\\ \hat{\alpha}\end{bmatrix} = \begin{bmatrix} y \\ 0\end{bmatrix} \qquad (1)$$

Denote estimates of $\sigma_1^2,\ \ldots,\ \sigma_{c+1}^2$ by $\hat{\sigma}_1^2,\ \ldots,\ \hat{\sigma}_{c+1}^2$, and assume that R and D in Equation (1) are formed from these estimates. Maximum-likelihood estimates are obtained as follows.

Let T* be the inverse of the matrix formed from the first q rows and columns of C'C, partitioned similarly to D:

$$T^* = \begin{bmatrix} T^*_{11} & \cdots & T^*_{1c} \\ \vdots & & \vdots \\ T^*_{c1} & \cdots & T^*_{cc} \end{bmatrix}.$$

Let

$$v_i^* = \mathrm{tr}(T^*_{ii}) \tag{2}$$

Then the maximum-likelihood estimates of $\sigma_1^2, \ldots, \sigma_{c+1}^2$ satisfy

$$\hat{\sigma}_i^2 = \left|\left| \hat{\beta}_i \right|\right|^2 / (q_i - v_i^*) \qquad (i = 1, \ldots, c) \tag{3}$$

and

$$\hat{\sigma}_{c+1}^2 = \frac{\left|\left| y - X \hat{\alpha} - Z \hat{\beta} \right|\right|^2}{\left[n - \sum(q_i - v_i^*) \right]}. \tag{4}$$

The restricted maximum-likelihood estimates maximize the likelihood of n - p linearly independent error contrasts in y (contrasts with zero expected value). They typically have smaller biases than the ordinary maximum-likelihood estimates, particularly when p is a large fraction of n. Restricted maximum-likelihood estimates of $\sigma_1^2, \ldots, \sigma_{c+1}^2$ are formed as a slight modification of the ordinary maximum-likelihood estimates.

Let T be the the matrix formed from the first q rows and columns of C'C, partitioned similarly to D:

$$T = \begin{bmatrix} T_{11} & \cdots & T_{1c} \\ \vdots & & \vdots \\ T_{c1} & \cdots & T_{cc} \end{bmatrix}.$$

Let

$$v_i = \mathrm{tr}(T_{ii}) \tag{5}$$

Then $\hat{\sigma}_1^2, \ldots, \hat{\sigma}_{c+1}^2$ satisfy

$$\hat{\sigma}_i^2 = \left|\left| \hat{\beta}_i \right|\right|^2 / (q_i - v_i) \quad (i = 1, \ldots, c) \tag{6}$$

and

$$\hat{\sigma}_{c+1}^2 = \frac{\left|\left| y - X \hat{\alpha} - Z \hat{\beta} \right|\right|^2}{\left[(n - p) - \sum(q_i - v_i) \right]} . \tag{7}$$

Fellner (1986) has developed an outlier-resistant variant of the method of restricted maximum likelihood.

Regardless of which of these methods is used, algorithms for obtaining $\hat{\sigma}_1^2, \ldots, \hat{\sigma}_{c+1}^2$ are necessarily iterative. On each iteration, Equation (1) must be solved for $\hat{\alpha}$ and $\hat{\beta}$, and at least the diagonal elements of T* or T must be computed in order to obtain new values of $\hat{\sigma}_1^2, \ldots, \hat{\sigma}_{c+1}^2$. These new values can be obtained by any of several methods (Harville 1977).

In many applications, q, the number of columns of Z, is very large. Then the solution of Equation (1) can be so demanding of computer storage and time that iterative methods become prohibitively expensive. However, in most applications, Z is a factorial design matrix, and this can be exploited to reduce the requirements enormously.

3. FACTORIAL DESIGNS

In most applications of the linear model outlined above, the matrix Z results from expressing factors and their interactions in terms of dummy variables. This imposes a structure on Z that is developed below. The matrix representations are similar to those of Bock (1963).

Let

$$Z_i = \zeta(K W_i) \qquad (i = 1, \ldots, c), \tag{8}$$

where K is an incidence matrix, taken to be independent of i, which retains selected rows of W_i, possibly with multiple replicates; and the operator ζ deletes the zero columns of KW_i.

The matrices W_i are defined as follows. Suppose that the linear model involves r factors, with factor j taking on n_j levels (j = 1, ..., r). Let Θ_i be a subset of the integers 1 through r. Then

$$W_i = J_{i1} \times \ldots \times J_{ir}, \tag{9}$$

where $\times$ denotes the direct (or Kronecker) product, and the matrix J_{ij} is the order-n_j identity matrix if $j \in \Theta_i$; otherwise, J_{ij} is the order-n_j vector $(1\ 1\ \ldots\ 1)'$.

Defined in this way, the matrices Z_i correspond to the usual ideas of main effects and interactions. For example, when $\Theta_i = \{1\}$, then Z_i consists of those columns of Z corresponding to the main effect of factor 1. Similarly, when $\Theta_i = \{1,2\}$, then Z_i consists of those columns representing the interaction of factors 1 and 2.

Definition 1: The partitioned matrix Z is a "factorial design" matrix if and only if its components, Z_i, satisfy Equations (8) and (9).

This definition allows for the possibility that some columns of KW_i are zero; that is, the design may have "missing cells".

The exploitation of the special structure of the matrix Z is based on its adjacency structure. For a fuller discussion of the graph-theoretic description of a matrix, see George and Liu 1981, Chapter 3.

Definition 2: Two columns of a matrix A are "adjacent" if and only if there is a row in which both have nonzero elements.

A pair of columns may be connected by a chain of adjacent columns. This concept is most easily expressed by a recursive definition.

Definition 3: Columns a_i and a_j of a matrix A are "connected" by the columns of A if and only if a_i and a_j are adjacent or a_i is adjacent to a third column, a_m, which is, in turn, "connected" to a_j.

For every set Θ_i, let

$$h_i = \Pi_{j \varepsilon \Theta_i} n_j \ . \tag{10}$$

Note that h_i is the number of columns of W_i.

Let ϕ be a subset of the integers 1 through c. Let W_ϕ be the partitioned matrix

$$W_\phi = \left[\ \ldots \ W_i \ \ldots \ \right] \qquad (\ i \ \varepsilon \ \phi \ ), \tag{11}$$

where W_i is given by Equation (9). Let

$$\Theta_\phi = \cap_{i \varepsilon \phi} \Theta_i \ , \tag{12}$$

and define h_ϕ by

$$h_\phi = \Pi_{j \varepsilon \Theta_\phi} n_j \ . \tag{13}$$

The following lemma quantifies the connectivity of the columns of W_ϕ.

Lemma 1: Let the partitioned matrix W_ϕ be defined by Equation (11), where the matrices W_i are defined by Equation (9). For i = 1, ..., c, let Θ_i be a subset of the integers 1 through k, and let h_i, Θ_ϕ and h_ϕ be defined by Equations (10) (12) and (13). Then, for i and j belonging to ϕ, the number of pairs of columns, one in W_i, the other in W_j, that are connected by columns in W_ϕ is

$$h_i \, h_j \, / \, h_\phi$$

for $i \neq j$, and

$$(1/2) \, h_i \, (\, 1 + h_i \, / \, h_\phi \,)$$

for $i = j$.

Proof: For $i \neq j$, the number of pairs of columns, one from W_i and the other from W_j, is $h_i \, h_j$. By Equations (9) and (11), the columns of the matrix W_ϕ can be permuted so that W_ϕ is block diagonal. The number of blocks is h_ϕ, and by symmetry, each block consists of the same number of columns. A similar argument holds for the case $i = j$. The lemma follows.

Let Z_ϕ be the partitioned matrix

$$Z_\phi = \left[\, \ldots \, Z_i \, \ldots \, \right] \qquad (\, i \, \varepsilon \, \phi \,), \tag{14}$$

where Z_i is given by Equation (8).

The following Lemma bounds the number of pairs of connected columns in Z.

Lemma 2: Let the partitioned matrix Z_ϕ be defined by Equation (14). For $i = 1, \ldots, c$, let Θ_i be a subset of the integers 1 through k, and let h_i, Θ_ϕ and h_ϕ be defined by Equations (10), (12) and (13). Then, for i and j belonging to ϕ, the number of pairs of columns, one in Z_i, the other in Z_j, that are connected by columns in Z_ϕ is bounded from above by

$$\min(\, q_i h_j / h_\phi \, , \, q_j h_i / h_\phi \,) \tag{15}$$

for $i \neq j$, and

$$(1/2) \, q_i (1 + h_i / h_\phi) \tag{16}$$

for $i = j$, where q_i is the number of columns of Z_i.

Proof: From Equations (8), (11) and (14), matrix K deletes, retains, replicates and/or permutes the rows of W_ϕ. Of these operations, only deletion modifies the adjacency structure of W_ϕ by making nonadjacent formerly adjacent pairs of columns. Similarly, the deletion of the columns of zeros in KW_i, performed by the operator ζ, does not alter the connectivity of the remaining columns.

Thus, for $i \neq j$, h_j / h_ϕ is an upper bound on the number of columns of Z_j that are connected to a specified column of Z_i, as is (trivially) q_j. Similarly, h_i / h_ϕ is an upper bound on the number of columns of Z_i that are connected to a specified column of Z_j.

A similar argument applies to the case $i = j$. The lemma follows.

4. SPARSENESS AND STORAGE REQUIREMENT

The solution of Equation (1) requires that the Cholesky factor of C'C be stored. Large quantities of storage can be saved if this factor is sparse; that is, if it has few nonzeros. Storage then need only be allocated to these nonzeros and pointers to them. In this section, the sparseness of the Cholesky factor of C'C is quantified.

Begin by considering the sparseness of the matrix F, where F is upper triangular, and $F'F = Z'Z$. We will ignore the fact that Z'Z is generally singular, since C'C is generally not. The following lemma characterizes the sparseness of F.

Lemma 3 (Parter 1961): Let $F'F = Z'Z = B$, where F is upper triangular. Let b_{ij} be the ij^{th} element of B, and let f_{ij} be the ij^{th} element of F. Then f_{ij} $(i \leq j)$ is nonzero only if b_{ij} is nonzero or there is a $k < i$ such that the product $f_{ki} f_{kj}$ is nonzero.

Corollary 1: Element f_{ij} $(i \leq j)$ of matrix F is nonzero only if columns z_i and z_j of matrix Z are adjacent or connected by columns $z_1, \ldots, z_i$.

Proof: By Definition 2, b_{ij} is nonzero only if columns z_i and z_j are adjacent. The corollary then follows from Definition 3 and Lemma 3.

Theorem 1 quantifies the sparseness of F when Z is a factorial design matrix.

Theorem 1: Let $B_{ij} = Z_i' Z_j$, where Z_i is given by Equation (5). Let $B = F'F$, where F is upper triangular and partitioned analogously to B. In Equation (14), let the set ϕ be given by

$$\phi = \{ 1, 2, \ldots, i \} \cup \{ j \} . \tag{17}$$

Then, for $i \leq j$, the number of nonzero elements in F_{ij} is bounded by Expression (15) for $i \neq j$, and by Expression (16) for $i = j$.

Proof: The theorem follows immediately by applying Lemma 2 and Corollary 1.

Nested designs are an important special case of factorial designs.

Definition 4: The matrix Z is "nested" if and only if $\theta_i \subset \theta_{i-1}$ $(i = 2, \ldots, c)$, where θ_i is the set used in Equation (9).

Corollary 2: If Z is nested, then the number of nonzero elements in F_{ij} $(i \leq j)$ is bounded by q_i, the number of columns of Z_i.

Proof: By Definition 4, and with ϕ given by Equation (17), $\theta_j = \theta_\phi$. The corollary follows.

In one form or another, Corollary 2 is well-known and accounts for the historic development of special-purpose computational algorithms for the analysis of nested designs.

Often a factorial design is large because one factor has a large number of levels. The following corollary then applies.

Corollary 3: Let the submatrices $Z_1, \ldots, Z_c$ be numbered so that $1 \in \Theta_i$ implies that $1 \in \Theta_{i-1}$ $(i = 2, \ldots, c)$. Then the number of nonzeros in the Cholesky factor of $Z'Z$ is $O(n_1)$, where n_1 is the number of levels of factor 1.

Proof: If $1 \in \Theta_i$ and $1 \in \Theta_j$, then h_a includes the multiplicative factor n_1. Otherwise, n_1 is a factor of at most one of the quantities, h_i and h_j. The corollary follows.

Since the total number of elements in the Cholesky factor is $O(n_1^2)$, Corollary 3 indicates that retention of only the nonzero elements can reduce storage requirements considerably.

Now consider the larger design matrix C. Let G be the Cholesky factor of $C'C$; that is, G is upper triangular and $G'G = C'C$. Partition G analogously to the partitioning of C in Equation (1):

$$G = \begin{bmatrix} G_{zz} & G_{zx} \\ 0 & G_{xx} \end{bmatrix}.$$

First note that the diagonal matrices, D and R, have no effect on the sparseness of G. Also, by Lemma 3, the matrix X, since it is to the right of matrix Z, has no effect on the sparseness of G_{zz}. Thus G_{zz} and F are equally sparse.

If the matrix X is full; that is, has no zero elements, then G_{zx} and G_{xx} will also be full. This would be the case if the columns of X were "covariates" or "regressors".

Suppose, instead, that the factorial structure of Z is continued into the matrix X. This would be the case if the matrix X represented factors regarded as "fixed" rather than "random". It might then be expected that Theorem 1 could be extended in a straightforward manner. However, there is a major difference between matrices Z and X. In the case of Z, the diagonal matrix $D^{-1/2}R^{1/2}$ in Equation (1) resolves any rank-deficiencies. No matrix plays a similar role in the case of X. Thus, X is typically rank-deficient.

Rank deficiencies in X are commonly resolved by adding side-conditions, that is, additional linear equations relating the elements of $\hat{\alpha}$. These equations, in effect, serve to make adjacent formerly nonadjacent columns of X. By Corollary 1, this has no effect on the sparseness of G_{zx}; however, in general, additional nonzeros in G_{xx} will be introduced.

Thus, if the factorial structure extends to matrix X, then Theorem 1 extends to G_{zx}. The sparseness of G_{xx} depends on the specific way in which rank-deficiencies in X are resolved. In general, X will have considerably fewer columns than will Z. Thus the potential lack of sparseness in G_{xx} is of little consequence.

5. AN ALGORITHM FOR UPDATING THE VARIANCE COMPONENTS

Up to now, we have considered the benefits of sparseness in terms of savings in storage requirements. However, any method that requires only storage of nonzero elements also provides for skipping the computational steps involving these elements.

In order to quantify the resulting saving in computational effort, it is necessary to specify a particular algorithm. We will consider the method of forming and solving the "normal equations":

$$C'C \begin{bmatrix} \hat{\beta} \\ \hat{\alpha} \end{bmatrix} = \begin{bmatrix} Z'Z+D^{-1}R & Z'X \\ X'Z & X'X \end{bmatrix} \begin{bmatrix} \hat{\beta} \\ \hat{\alpha} \end{bmatrix} = \begin{bmatrix} Z'y \\ X'y \end{bmatrix} \tag{18}$$

This method is appropriate if minimizing computational effort is the prime criterion. This is the algorithm described below. If, on the other hand, numerical stability is the major concern, the formation of the normal equations can be avoided by directly decomposing the matrix C of Equation (1) using Givens rotations (George and Heath 1980).

Since the computations are iterative, only that part of the computational effort that must be repeated for each iteration will be quantified.

Algorithm A: Given estimates of the variance components, the following steps constitute an iteration for the restricted maximum likelihood estimates:

Step 1: Update the diagonal-matrix term $D^{-1}R$ of $C'C$ in Equation (18).

Step 2: Obtain G, the Cholesky factor of $C'C$.

Step 3: Obtain the least-squares solution to Equation (18), as well as the residual sum-of-squares.

Step 4: Obtain G^{-1}.

Step 5: Obtain the first q diagonal elements of $(C'C)^{-1}$.

Step 6: Update the variance component estimates, using Equations (5), (6) and (7).

Before considering this algorithm quantitatively, some comments are in order.

As noted by Hemmerle and Hartley (1973), the normal equations are unchanged from iteration to iteration, apart from the diagonal matrix $D^{-1}R$. Thus they need to be formed only once.

If ordinary maximum likelihood estimates are wanted, only the first q rows and columns of G need be inverted in Step 4. Then obtain the diagonal elements of $(Z'Z+D^{-1}R)^{-1}$ in Step 5. Equations (2), (3) and (4) should be substituted in Step 6.

Sparse-matrix algorithms for implementing Steps 2 and 3 are readily obtained. Both the Yale Sparse Matrix Package (Eisenstat et al 1977) and SPARSPAK (George and Liu 1978) provide the necessary capabilities, including algorithms for determining the nonzero structure of the Cholesky factor of $C'C$.

The matrix G^{-1} will generally contain a larger number of nonzeros than will G itself. However, Steps 4 and 5 can be combined in such a way that the extra storage requirement is

avoided. The appendix gives a sparse-matrix algorithm for this purpose that fits well with both the Yale Sparse Matrix Package and SPARSPAK.

The use of Equations (5), (6) and (7) to update the variance component estimates has been suggested by Harville (1977), who reviews a number of alternative calculations for this step. The use of Newton-Raphson or Fisher-Scoring techniques (Jennrich and Sampson 1976) would reduce the number of iterations required, but would require the calculation of the off-diagonal elements of the full matrix $(C'C)^{-1}$ in Step 5. This would greatly increase the amount of computation per iteration. Least-change secant update methods (Dennis and Schnabel 1979) potentially improve convergence without greatly increasing the computational cost per iteration, but we have not tried them.

6. SPARSENESS AND COMPUTATIONAL EFFORT

In Algorithm A, Steps 2 through 5 comprise the bulk of the computational effort. Most of this effort involves forming dot products of the form $\Sigma_i r_i s_i$. Thus it is convenient to count a floating point multiplication followed by a floating point addition in the calculation of a dot product as a "dot-product operation". A dot-product operation on a pair of matrix elements need only be carried out if both elements are nonzeros. This provides a basis for quantifying the computational effort involved.

The following lemma quantifies the connectivity among triples of columns of Z. It is analogous to Lemma 2, and its proof will be omitted.

Lemma 4: For a subset ϕ of {1, ..., c}, let Z_ϕ be defined by Equation (14). For i = 1, ..., c, let Θ_i be a subset of {1, ..., r}, and let h_i, Θ_ϕ and h_ϕ be defined by Equations (10), (12) and (13). Moreover, for a subset ψ of {1, .., c}, let W_ψ, Θ_ψ and h_ψ be defined similarly. Let i and j belong to ϕ and let j and k belong to ψ. Assume that, if i = k, then $\phi = \psi$. Consider triples of columns, one in Z_i, one in Z_j and one in Z_k, such that

the one in Z_i is connected to the one in Z_j by columns in Z_ϕ, and the one in Z_k is connected to the one in Z_j by columns in Z_ψ. Then the number of such triples is bounded from above by

$$[\min(q_i h_j h_k, h_i q_j h_k, h_i h_j q_k)]/(h_\phi h_\psi) \qquad (i\neq j, j\neq k, k\neq i) \qquad (19)$$

$$(1/2)\ [\min(q_i h_k, h_i q_k)](1+h_i/h_\phi)/h_\psi \qquad (i=j, j\neq k) \qquad (20)$$

$$(1/2)\ [\min(q_i h_j, h_i q_j)](1+h_j/h_\psi)/h_\phi \qquad (i\neq j, j=k) \qquad (21)$$

$$(1/2)\ [\min(q_i h_j, h_i q_j)](1+h_i/h_\phi)/h_\phi \qquad (i=k, j\neq k) \qquad (22)$$

$$(1/6)\ q_i(1+h_i/h_\phi)(1+2h_i/h_\phi) \qquad (i=j=k) \qquad (23)$$

Using Theorem 1, storage requirements can be assessed by considering the storage requirement corresponding to each pair of submatrices Z_i and Z_j. In a similar way, the required number of dot-product operations for some calculations can be assessed by considering triples of submatrices Z_i, Z_j and Z_k. Using this approach, Theorems 2 through 5 bound the number of dot-product operations required to carry out Steps 2 through 5, respectively.

As in Section 4, we will consider mainly the requirements associated with matrix Z only. Thus, as in Section 4, references will be to the matrix $B = Z'Z$ and its Cholesky factor F.

Theorems 2 through 5 will be stated without proof. They follow immediately from Lemmas 2 and 4, after writing out the full-matrix algorithms for Steps 2 through 5, which are well-known (for example, Lawson and Hanson 1974, Chapters 11, 12 and 19).

Theorem 2: Let B and F be defined as in Theorem 1. In Lemma 4, let ϕ and ψ be given by

$$\phi = \{\ 1,\ 2,\ \ldots,\ j\ \} \cup \{\ i\ \}$$

and

$$\psi = \{ 1, 2, \ldots, j \} \cup \{ k \} ,$$

where $j \leq i$ and $j \leq k$. Then, for $i \leq k$, the number of dot-product operations required to compute F_{ik} that involve elements of both F_{ji} and F_{jk} is bounded by Equations (19), (20), (21), (22) or (23) as appropriate.

Theorem 3: Under the conditions of Theorem 2, the number of dot-product operations necessary to solve $F'Fw = Z'y$ for w, given F and $Z'y$, is equal to twice the number of nonzeros in F.

Thus bounds on the number of dot-product operations necessary to solve $F'Fw = Z'y$ can be obtained from Theorem 1.

Theorem 4: Let F^{ik} be the ik^{th} submatrix of F^{-1}, where F is defined as in Theorem 1. In Lemma 4, let ϕ and ψ be given by

$$\phi = \{ 1, 2, \ldots, j \}$$

and

$$\psi = \{ 1, 2, \ldots, j \} \cup \{ k \} ,$$

where $i \leq j \leq k$. Then the number of dot-product operations required to compute F^{ik} from elements of both F^{ij} and F^{jk} is bounded from above by Equations (19), (20), (21), (22) or (23), as appropriate.

Theorem 5: Let F^{-1} be partitioned as in Theorem 4. In Lemma 2, let the set ϕ be given by

$$\phi = \{ 1, 2, \ldots, j \} .$$

Then, for $i \leq j$, the number of nonzero elements in F^{ij} is bounded by Expression (15) for $i \neq j$, and by Expression (16) for $i = j$.

Moreover, the number of dot-product operations required to compute the diagonal elements of $(F'F)^{-1}$ is equal to the number of nonzero elements in F^{-1}.

In addition to the dot-product operations, Algorithm A involves additional operations such as the square-roots in the Cholesky factorization. The amount of this additional calculation is proportional to q, the number of columns of Z, and can therefore be regarded as small.

As discussed in the previous section, a factorial design is often large because a single factor has a large number of levels. This case is considered in the following corollary to Theorems 2 through 5. The proof is similar to that of Corollary 3 and will be omitted.

Corollary 4: Under the conditions of Corollary 3, the amount of computation necessary to carry out Algorithm A is $O(n_1)$.

The amount of computation required using full-matrix algorithms would normally be $O(n_1^3)$. Just as Corollary 3 points to a reduction in storage requirements, Corollary 4 indicates that a considerable reduction in computational effort can be obtained through a sparse-matrix implementation.

7. EXAMPLE: THE TWO-WAY CROSSED DESIGN

Consider the two-way crossed design (r = 2). Let Z_2 represent the main effect of factor 1, Z_3 represent the main effect of factor 2, and Z_1 their interaction. Thus $\theta_1 = \{1,2\}$, $\theta_2 = \{1\}$ and $\theta_3 = \{2\}$. Assuming no "missing cells", the number of nonzeros in each of the partitions of F, obtained from Expressions (15) and (16), are shown below:

i	j	ϕ	number of elements in F_{ij}	number of nonzeros in F_{ij}

1 1	{1}	$n_1n_2(1+n_1n_2)/2$*	n_1n_2	
1 2	{1,2}	$n_1^2n_2$	n_1n_2	
1 3	{1,3}	$n_1n_2^2$	n_1n_2	
2 2	{1,2}	$n_1(1+n_1)/2$*	n_1	
2 3	{1,2,3}	n_1n_2	n_1n_2	(full)
3 3	{1,2,3}	$n_2(1+n_2)/2$*	$n_2(1+n_2)/2$	(full)

*upper triangle only

The total number of nonzeros in F is $O(n_1)$, as predicted by Corollary 3. On the other hand, the total number of nonzeros is $O(n_2^2)$. This is of lesser importance, since one would select the factors so that $n_1 \geq n_2$.

For the random crossed design, the matrix X is the single column [1 1 ... 1]'. Thus G_{zx} and G_{xx} are full single-column matrices.

Just as the table above was constructed from Theorem 1, similar tables can be constructed from Theorems 2 through 5 in order to quantify the reduction in computational effort. For this effort to be meaningful, it is necessary to be able to neglect operation counts of order n_1n_2 or lower. Otherwise, the computational effort is materially affected by operations not included in Theorems 2 through 5. Thus, n_1 and n_2 must be large. If we also let $n_1 >> n_2$, then the ratio of the full-matrix operation count to the sparse-matrix operation count is approximately $(2/21)n_1^2n_2$. Of course, the factor n_1^2 is predicted by Corollary 4.

8. COLUMN ORDERING

It is clear from Lemma 3 that the number of nonzeros in F depends on the ordering of the columns of Z. One algorithm for

obtaining a good ordering for general sparse symmetric matrices is the *Minimum Degree Ordering Algorithm* (Tinney 1969). This algorithm selects the first column of Z to minimize the number of nonzeros in the first row of F; given that selection, the second column of Z is selected to minimize the number of nonzeros in the second row of F; etc. The algorithm can be costly to execute. For most factorial designs, good orderings can be had more cheaply.

A less-costly approach is to consider reordering the small number of submatrices $Z_1, \ldots, Z_c$ to minimize the sum of the upper bounds given in Theorem 1. The following theorem reduces the number of orderings to be considered.

Theorem 6: Let $Z = Z_1, \ldots, Z_c$ be a factorial design matrix. If, for some $i < j$, $\theta_i \subset \theta_j$, then a reordering of the submatrices of Z exists that does not increase the number of nonzeros in the Cholesky factor of $Z'Z$.

Proof: Consider the reordering

$$Z_1, \ldots, Z_{i-1}, Z_j, Z_i, \ldots, Z_{j-1}, Z_{j+1}, \ldots, Z_c .$$

Since $\theta_i \subset \theta_j$, it follows that every column of W_j is adjacent to exactly one column of W_i. Moreover, if a column of W_j is adjacent to a column of W_i, the column of W_i has a nonzero corresponding to every nonzero of the column of W_j. Thus deletion of mutual rows of both columns will only break their adjacency when the column from W_j becomes zero. Thus every column of Z_j is adjacent to exactly one column of Z_i.

Thus, any nonzeros in F induced by connections through Z_j are also induced by connections through Z_i. Thus moving Z_j to its new location possibly reduces the number of nonzeros in F_{mj} ($m = 1, \ldots, j$) without increasing the number of nonzeros in any of the other submatrices of F. The theorem follows.

Theorem 6 shows that orderings in which $i < j$ and $\theta_i \subset \theta_j$ need not be considered.

As a practical matter, quite good column orderings are produced by simply ordering the submatrices $Z_1, \ldots, Z_c$ so that Z_1 has the largest number of columns, Z_2 the next largest, etc. Heuristically, this assigns to the largest submatrices of F the largest values of h_ϕ. Moreover, this procedure never produces an ordering for which, for some $i < j$, $\theta_i \subset \theta_j$.

It is generally preferable to keep the columns of X to the right of the columns of Z. Otherwise, the nonzeros produced by the imposition of side conditions will not be confined to the submatrix G_{xx}. This consideration will rarely violate the ordering principle implied by Theorem 6.

9. THE COMPLETE ALGORITHM

Algorithm A described the iterative part of the calculation of the restricted maximum-likelihood estimates of the variance components. The complete calculation is as follows.

Algorithm B: Given the matrices $Z_1, \ldots, Z_c$ and X, and the vector y, the following algorithm finds restricted maximum-likelihood estimates of the variances $\sigma_1^2, \ldots, \sigma_c^2$:

Step 1: Renumber the matrices $Z_1, \ldots, Z_c$ in decreasing order by number of columns.

Step 2: If X has factorial structure, renumber its submatrices in decreasing order by number of columns.

Step 3: Determine the nonzero structure of the matrix C'C, including nonzeros imposed by any side conditions.

Step 4: Determine the nonzero structure of G, the Cholesky factor of C'C.

Step 5: Obtain starting values for the variance estimates, $\hat{\sigma}_1^2, \ldots, \hat{\sigma}_{c+1}^2$.

Step 6: Obtain the normal equations, apart from the additive diagonal matrix $D^{-1}R$.

Step 7: Perform Algorithm A repeatedly until the variance component estimates converge.

In Step 5, one should avoid full-matrix algorithms for obtaining starting values for the variance estimates. One set of estimates that are amenable to a sparse-matrix implementation are given as the solution of the simultaneous equations

$$\sum_{j=1}^{c+1} ||Z_i'Z_j||^2 \hat{\sigma}_j^2 = ||Z_i'd||^2 \quad (i = 1, \ldots, c+1), \qquad (24)$$

where Z_{c+1} is taken to be the identity matrix, and d is the vector of residuals to the least-squares solution for $\hat{\alpha}$ of the system of equations $y = X\hat{\alpha}$. These estimates represent one step from zero starting values of Anderson's (1973) iterative algorithm for the maximum-likelihood estimates. They are also somewhat related to the MIVQUE estimates (again from zero starting values) of LaMotte (1973) and Rao (1972). The solution to (24) would need adjustment if any of the elements were zero or negative.

The suggested convergence criterion in Step 7 is

$$\left| \Delta\hat{\sigma}_i^2 \right| < \delta \sum_j \hat{\sigma}_j^2 \qquad (i = 1, \ldots, c+1)$$

where $\Delta\hat{\sigma}_i^2$ is the change in $\hat{\sigma}_i^2$ from the previous iteration, and δ is a small constant. The use of this criterion implies that we will not seek small relative changes in a small variance component. Otherwise, iteration would never terminate if a component were headed towards zero.

BIBLIOGRAPHY

T. W. Anderson (1973). "Asymptotically Efficient Estimation of Covariance Matrices with Linear Structure", Annals of Statist., 1, pp 135-141

R. D. Bock (1963). "Programming Univariate and Multivariate Analysis of Variance", Technometrics, 5, pp 95-117.

J. E. Dennis and R. B. Schnabel (1979). "Least Change Secant Updates for Quasi-Newton Methods" SIAM Review, 21, pp 443-459

S. C. Eisenstat, M. C. Gursky, M. H. Schultz and A. H. Sherman (1981). "Yale Sparse Matrix Package: I Symmetric Matrices" Rep. 112, Dept. of Computer Science, Yale University, New Haven.

S. C. Eisenstat, M. H. Schultz, A. H. Sherman (1981). "Algorithms and Data Structures for Sparse Symmetric Gaussian Elimination" SIAM J. on Scientific and Statistical Computing, 2, pp 225-237.

W. H. Fellner (1986). "Robst Estimation of Variance Components" Technometrics, 28, pp 51-60.

A. George and M. T. Heath (1980). "Solution of Sparse, Linear Least Squares Problems Using Givens Rotations", Linear Algebra and its Applications, 34, pp 69-84.

A. George and J. W. Liu (1978). "User Guide for SPARSPAK: Waterloo Sparse Linear Equations Package", Rep. CS-78-30, Dept. of Computer Science, University of Waterloo.

A. George and J. W. Liu (1981). Computer Solution of Large Sparse Positive Definite Systems, Prentice-Hall, Englewood Cliffs.

D. A. Harville (1977). "Maximum Likelihood Approaches to Variance Component Estimation and to Related Problems", J. Amer. Statist. Assoc., 72, pp 320-340.

W. J. Hemmerle and H. O. Hartley (1973). "Computing Maximum Likelihood Estimates for the Mixed A.O.V. Model Using the W Transformation", Technometrics, 15, pp 819-831.

C. R. Henderson (1963). "Selection Index and Expected Genetic Advance", in Statistical Genetics and Plant Breeding, National Academy of Sciences - National Research Council Publication No. 892, pp 141-163.

R. I. Jennrich and P. F. Sampson (1976). "Newton-Raphson and Related Algorithms for Maximum Likelihood Variance Component Estimation", Technometrics, 18, pp 11-17.

L. R. LaMotte (1973). "Quadratic Estimation of Variance Components", Biometrics, 29, pp 311-330.

C. L. Lawson and R. J. Hanson (1974). Solving Least Squares Problems, Prentice-Hall, Englewood Cliffs, New Jersey.

S. V. Parter (1961). "The Use of Linear Graphs in Gauss Elimination", SIAM Review, 3, pp 119-130.

H. D. Patterson and R. Thompson (1971). "Recovery of Inter-Block Information when Block Sizes are Unequal", Biometrika, 58, pp 545-554.

C. R. Rao (1972). "Estimation of Variance and Covariance Components in Linear Models", J. Amer. Statist. Assoc., 67, pp 112-115.

W. F. Tinney (1969). "Comments on Using Sparsity Techniques for Power System Problems", in Sparse Matrix Proceedings, IBM Research Rep. RAI 3-12-69.

APPENDIX: COMPUTING THE DIAGONAL ELEMENTS OF $(C'C)^{-1}$

Let G be the upper-triangular Cholesky factor of C'C (=G'G), and let g_{ij} be its ij^{th} element. Assume that, for each row of G, a list of the nonzero elements is stored. Let ω be the set of subscripts (ij) corresponding to nonzeros in G.

Algorithm C, below, computes s_m, the m^{th} diagonal element of $(C'C)^{-1}$ in such a way as to avoid all "looping" over zero

elements, including initialization. In effect, the algorithm combines Steps 4 and 5 of Algorithm A. In this algorithm, the vector x is a working array of length equal to the order of C'C.

Algorithm C: Given Cholesky factor G and subscript m, the following algorithm computes the m^{th} diagonal element of $(G'G)^{-1}$:

$k \leftarrow m$

$\lambda \leftarrow \{ j \mid (mj) \in \omega, j > m \}$

Until λ is empty, do:

$x_k \leftarrow 0$

$k \leftarrow \min(j \mid j \in \lambda)$

$\lambda \leftarrow \{ j \mid (kj) \in \omega, j > k \}$

$k \leftarrow m$

$x_m \leftarrow 1 / g_{mm}$

$\lambda \leftarrow \{ j \mid (mj) \in \omega, j > m \}$

Until λ is empty, do:

For all $j \in \lambda$, do:

$x_j \leftarrow x_j - x_k g_{kj}$

$x_k \leftarrow x_k / g_{kk}$

$s_m \leftarrow s_m + x_k^2$

$k \leftarrow \min(j \mid j \in \lambda)$

$\lambda \leftarrow \{ j \mid (kj) \in \omega, j > k \}$

Return s_m and stop.

Received by Editor December, 1985; Revised November, 1986.

Recommended by M. S. Patel, University of Nairobi, Nairobi, Kenya.

Refereed by J. W. Odhiambo, University of Nairobi, Nairobi, Kenya.