Matrix Computations

Much of the development and formulation of the mathematical and statistical models that biostatisticians use relies heavily on matrix notation (*see* **Matrix Algebra**). For example, matrix notation is often the preferred way to describe the mathematics underlying many of the procedures in statistical packages (*see* **Software, Biostatistical**). Routines are now widely available that implement standard matrix operations including matrix multiplication and the solution of systems of linear equations, facilitating computer implementation of matrix formulas presented in the literature. Some knowledge of the alternative available **algorithms** is helpful both for implementers of matrix formulas and for users of existing statistical package implementations. In the following we comment on some alternative widely used approaches to linear **least squares** and related calculations.

Areas where matrix computations have a large place include **regression** methods, **multivariate analysis**, **maximum likelihood** estimation, **robust** estimation, smoothing, and **optimization**. Linear matrix computational methods are more generally important because nonlinear problems are frequently handled by solving a sequence of linearized problems. Numerical linear algebra is, effectively, another name for matrix computations.

Modern numerical matrix algebra gains much of its power from the use of a relatively small number of matrix decompositions, whose numerical properties are well understood. Major aims are guaranteed accuracy, speed of computation (efficiency), and the ability to handle all inputs [6, 7, 9]. The article [9] discusses several topics that we omit or only mention in passing.

Implementing Matrix Computations

Matrix computations must reckon with the finite precision of computer arithmetic. Most common computers now implement the IEEE standard for **floating point arithmetic**, which has around seven decimal digits single-precision and around 16 decimal digit double-precision arithmetic. The double-precision standard is a sound basis on which to build accurate and reliable algorithms.

Technical accuracy and efficiency issues are reasons for providing expert "black box" implementations of what might appear simple calculations such as $||\mathbf{x}|| = (\mathbf{x}'\mathbf{x})^{1/2}$ and matrix multiplication. Specifications for sets of lower-level routines have been established in the **numerical analysis** literature, where they are known as BLAS (basic linear algebra subroutines) [1]. The BLAS, or other such lower-level routines, then make effective building blocks in the creation of higher-level routines.

Understanding Matrix Methods

There are often, in matrix computations just as elsewhere, several different ways to solve the same problem. Knowledge of matrix algorithms may allow the substitution of one algorithm for another when required. For example, a published formula may involve a matrix operation not found in available software. Additional information that is required from a routine may be available, for someone who understands the algorithm, as an adaptation of existing output.

Often it is helpful to know what accuracy can reasonably be expected from a calculation. When results from different algorithms for the same problem differ numerically (perhaps in decimal places after the third or fourth), which is more accurate? What characteristics of input data may lead to such differences in accuracy? Knowledge of the algorithm may be even more important when a calculation fails.

Matrix Inversion

The use of matrix inverses is a convenience in writing down matrix formulas. However, direct implementation of such formulas rarely leads to algorithms that are optimal for practical computation. For example, solving $\mathbf{Sb} = \mathbf{c}$ for $\mathbf{b}$ is preferable to forming $\mathbf{S}^{-1}$ and computing $\mathbf{b} = \mathbf{S}^{-1}\mathbf{c}$. Avoiding unnecessary matrix inversion reduces computational effort, leading to a small improvement in precision. There is a choice of default actions where the inverse does not exist. Later in this article we illustrate approaches which avoid the explicit calculations of matrix inverses.

Linear Least Squares

Linear **least squares** has been the context for much of the discussion of statistical matrix computations. As

Encyclopedia of Biostatistics, Online © 2005 John Wiley & Sons, Ltd.
This article is © 2005 John Wiley & Sons, Ltd.
This article was published in the *Encyclopedia of Biostatistics* in 2005 by John Wiley & Sons, Ltd.
DOI: 10.1002/0470011815.b2a14022

well as being important for linear least squares, the matrix computations we describe are important building blocks for many other statistical computations.

We consider the contrived example

$$(\mathbf{X}|\mathbf{y}) = \left[\begin{array}{rrr|r} 1 & 7 & 8 & 6 \\ 1 & -3 & 4 & 4 \\ 1 & 2 & 2 & 0 \\ 1 & 2 & 2 & 6 \\ 1 & 7 & 6 & 5 \\ 1 & 2 & 4 & 7 \\ 1 & -3 & 2 & 3 \\ 1 & 2 & 4 & 1 \\ 1 & 2 & 4 & 4 \end{array}\right].$$

Given $\mathbf{X}(n \times p)$ and $\mathbf{y}(n \times 1)$, least squares calculations determine $\mathbf{b}(p \times 1)$ such that

$$(\mathbf{y} - \mathbf{Xb})'(\mathbf{y} - \mathbf{Xb}) = ||\mathbf{y} - \mathbf{Xb}||^2 \qquad (1)$$

is a minimum. In the example above one minimizes the sum of squares $[6 - (b_1 + 7b_2 + 8b_3)]^2 + [4 - (b_1 - 3b_2 + 4b_3)]^2 + \cdots$

Algebraically, the linear least squares problem (1) is equivalent to solving what are called the normal equations, i.e.

$$\mathbf{X}'\mathbf{Xb} = \mathbf{X}'\mathbf{y}. \qquad (2)$$

If $\mathbf{S} = \mathbf{X}'\mathbf{X}$ is singular, theoretical arguments show that the normal equations are consistent, but rather than just one solution there are an infinity of solutions. An example appears below in the section on linear dependencies.

We describe and contrast two approaches to the linear least squares problem, one of which forms and solves the normal equations, while the other (the QR method) avoids formation of the normal equations.

A Normal Equation Approach

An effective way to solve the normal equations is to use the Cholesky algorithm, which modifies Gaussian elimination to take advantage of the symmetry of the normal equation matrix of coefficients. Diagrammatically, the steps are

$$(\mathbf{X}|\mathbf{y}) \rightarrow \begin{pmatrix} \mathbf{X}'\mathbf{X} & \mathbf{X}'\mathbf{y} \\ (\mathbf{X}'\mathbf{y})' & \mathbf{y}'\mathbf{y} \end{pmatrix} \rightarrow \begin{pmatrix} \mathbf{R} & \mathbf{d} \\ \mathbf{0}' & r_{yy} \end{pmatrix}. \qquad (3)$$

The normal equations $\mathbf{X}'\mathbf{Xb} = \mathbf{X}'\mathbf{y}$ reduce to $\mathbf{Rb} = \mathbf{d}$, where $\mathbf{R}$ is $p \times p$ upper triangular, i.e. below diagonal elements are zero. (It might also be described as right triangular, which perhaps justifies the symbol $\mathbf{R}$.) It is convenient to take $\mathbf{R}$ to be the upper triangular matrix which is formed by the Cholesky decomposition of $\mathbf{X}'\mathbf{X}$, i.e. $\mathbf{R}'\mathbf{R} = \mathbf{X}'\mathbf{X}$. On the right-hand side of (3), $\mathbf{R}$ is augmented with an additional row and column, to form an array which is the Cholesky decomposition of $(\mathbf{X}|\mathbf{y})'(\mathbf{X}|\mathbf{y})$.

For our numerical example the system of equations $\mathbf{Rb} = \mathbf{d}$ is

$$\begin{pmatrix} 3 & 6 & 12 \\ 0 & 10 & 4 \\ 0 & 0 & 4 \end{pmatrix} \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix} = \begin{pmatrix} 12 \\ 2 \\ 2 \end{pmatrix}.$$

Calculations are completed by solving first for b_3 $\left(=\frac{1}{2}\right)$, then for $b_2 (= 0)$ in terms of b_3, and finally for $b_1 (= 2)$ in terms of b_2 and b_3.

The QR Method for Linear Least Squares

Our description will emphasize points of contact with the normal equations approach. The QR method omits the intermediate step in (3). It determines

$$\mathbf{Q}(\mathbf{X}|\mathbf{y}) = \begin{pmatrix} \mathbf{R} & \mathbf{d} \\ \mathbf{0} & \mathbf{z} \end{pmatrix}, \qquad (4)$$

where $\mathbf{Q}(n \times n)$ is a product of orthogonal matrices and is hence orthogonal, i.e. $\mathbf{Q}'\mathbf{Q} = \mathbf{QQ}' = \mathbf{I} = \mathrm{diag}(1, \ldots, 1)$. The vector $\mathbf{z}$ is $(n - p) \times 1$. If we insist that $\mathbf{R}$ have positive diagonal elements, then it is algebraically identical to the matrix $\mathbf{R}$ formed by the Cholesky decomposition of $\mathbf{X}'\mathbf{X}$. The quantity $\mathbf{z}'\mathbf{z}$ is the sum of squares of residuals from the regression, and equals r_{yy}^2.

Other Methods for Least Squares

Other methods for least squares include the once popular Gauss–Jordan scheme, which calculates $(\mathbf{X}'\mathbf{X})^{-1}$ as well as $\mathbf{b}$. There are in addition a range of iterative methods for least squares, which have found particular application in large sparse problems [3, 6, 9].

DOI: 10.1002/0470011815.b2a14022

Linear Dependencies

In the data set

$$(\mathbf{X}|\mathbf{y}) = \left[\begin{array}{ccc|c} 1 & -2 & -4 & -1 \\ 1 & 1 & -1 & 0 \\ 1 & 2 & 0 & 4 \\ 1 & 5 & 3 & 7 \end{array}\right],$$

the third column is the difference between the second column and twice the first column. Linear dependencies, of which this is a trivial example, arise in least squares problems when one or more variables are a linear combination of earlier variables. The normal equations are

$$\begin{pmatrix} 4 & 6 & -2 \\ 6 & 34 & 22 \\ -2 & 22 & 26 \end{pmatrix} \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix} = \begin{pmatrix} 10 \\ 45 \\ 25 \end{pmatrix}.$$

The matrix of coefficients $\mathbf{S} = \mathbf{X}'\mathbf{X}$ is, because the coefficients in row 3 are the difference between row 2 and twice row 1, *singular*. Hence $\mathbf{S}^{-1}$ does not exist. Nevertheless the coefficients are, because from normal equations, consistent. With b_3 chosen arbitrarily, $b_2 = 1.2 - b_3$ and $b_1 = 0.7 + 2b_3$. Such nonunique solutions occur when one variable or term in a model is a linear combination of other terms.

In **analysis of variance** applications, dependencies may arise because there are inadequate data to allow the estimation of one or more parameters associated with main effects or interactions. Alternately, one or more **explanatory** variables may be an exact linear combination of other terms in the model, and a decision is needed on which terms to include. Dependencies may be a result of an unanticipated feature of the input data, or of a mistake in the data.

Dependencies are, when working with observational data on a large number of variables, surprisingly common. They may be a huge source of frustration, especially if the program responds by exiting with an uninformative error message. Sensible default actions, and information on the coefficients of the linear relation, may be a huge help. Where column i of $\mathbf{X}$ is a linear combination of earlier columns, an easy device which will allow calculations to continue is to set b_i to zero, effectively deleting column i of $\mathbf{X}$. It would be useful to have criteria for detecting instances where a near singularity may make results nonsensical or hard to interpret. Regrettably, there are no effective simple criteria that will cover all circumstances.

Normal Equations vs. QR

At a fixed level of numerical precision the QR decomposition will solve a wider range of problems than normal equation methods. The difference is marked when there are strong dependencies between the columns of $\mathbf{X}$, leading to a large **standard error** for one or more elements of $\mathbf{b}$. A consequence of large standard error(s) is that the additional numerical precision is unlikely to be statistically meaningful.

The solution of the normal equations retains very nearly the accuracy of $\mathbf{X}'\mathbf{X}$ and $\mathbf{X}'\mathbf{y}$. Where $\mathbf{X}$ has an initial column of ones, precision may be assisted by expressing values in remaining columns as differences from the column mean, prior to forming $\mathbf{X}'\mathbf{X}$ and $\mathbf{X}'\mathbf{y}$. Careful implementations of normal equation methods take this precaution. The precision of $\mathbf{X}'\mathbf{X}$ and $\mathbf{X}'\mathbf{y}$ is then equivalent to that of an accurately formed correlation matrix. In applications in the biological and social sciences, where differences from the mean are rarely accurate to more than two or three significant digits, this seems adequate precision.

Caution may nevertheless advise use of the QR method except in those applications – unbalanced analysis of variance, for example – where columns of $\mathbf{X}$ are unlikely to be highly correlated. There is a helpful discussion in [5] which compares the normal equation method with QR (see also [6]).

QR Algorithms

Another name for the QR method is orthogonal reduction to upper triangular form. Available algorithms for QR include Householder and modified Gram–Schmidt (MGS), which proceed columnwise, and the Givens algorithm, which operates on new rows one at a time to incorporate them into the current version of $\mathbf{R}$. We discuss these in more detail below. Elements of $\mathbf{Q}$ are unlikely to be stored explicitly; instead, key quantities are stored from which $\mathbf{Q}$ can be reconstructed as required.

Algorithms for QR factorization effectively form rows of $\mathbf{R}$ as linear combinations of rows of $\mathbf{X}$ rather than as linear combinations of rows of $\mathbf{X}'\mathbf{X}$. They avoid the loss of accuracy which, in normal equation methods, may occur in the formation of $\mathbf{X}'\mathbf{X}$ and $\mathbf{X}'\mathbf{y}$.

Encyclopedia of Biostatistics, Online © 2005 John Wiley & Sons, Ltd.
This article is © 2005 John Wiley & Sons, Ltd.
This article was published in the *Encyclopedia of Biostatistics* in 2005 by John Wiley & Sons, Ltd.
DOI: 10.1002/0470011815.b2a14022

There is some additional computational cost. When p is much smaller than n, use of QR approximately doubles the number of multiplications and divisions compared with using the normal equations.

Various diagnostic and other information that may be required for least squares modeling may be computed straightforwardly from submatrices of $\mathbf{Q}$. Examples include leverage statistics (*see* **Diagnostics**), and the variance–**covariance matrix** of **residuals**. Brief details appear in a later section.

Some Key Matrix Methods

Here we discuss in more detail several algorithms that have major importance in statistical computation, including algorithms mentioned above. We emphasize the connections between algorithms which, to first appearance, are quite different.

Cholesky Decomposition

Given a positive definite matrix $\mathbf{S}$, perhaps formed as $\mathbf{X}'\mathbf{X}$, the Cholesky decomposition determines an upper triangular matrix $\mathbf{R}$ such that $\mathbf{S} = \mathbf{R}'\mathbf{R}$. Equivalently, one may form $\mathbf{S} = \mathbf{U}'\mathbf{D}\mathbf{U}$, where $\mathbf{D}$ is diagonal and $\mathbf{U}$ is upper triangular with unit diagonal.

Several algorithms are available, which differ in the order in which they form elements of $\mathbf{R}$. In the version we now describe, elements in the first row of $\mathbf{R}$ are formed as

$$r_{11} = \sqrt{s_{11}}, \qquad r_{1j} = r_{11}^{-1} s_{1j}, \quad j = 1, \ldots, p.$$

Then, for $i = 2, \ldots, p$, calculate

$$s_{ij}^{(i-1)} = s_{ij} - \sum_{k=1}^{i-1} r_{ki} r_{kj}, \quad j = i, \ldots, p,$$

and

$$r_{ii} = [s_{ii}^{(i-1)}]^{1/2}, \quad \text{if } (i < p)\ r_{ij} = r_{ii}^{-1} s_{ij}^{(i-1)}, \quad j = i + 1, \ldots, p.$$

Note that if $\mathbf{X}$ has an initial column of ones and remaining columns are centered by expressing values as differences from the column mean, then $1 - s_{ii}^{-1} s_{ii}^{(i-1)}$ is the squared multiple correlation measuring the dependence of column k of $\mathbf{X}$ on earlier columns. Where $r_{ii} = 0$, all elements in that row may be set to zero.

The Cholesky decomposition may be used in solving the generalized weighted least squares problem, where $\mathbf{W}$ is a positive-definite symmetric weighting matrix. Observe that if $\mathbf{U}$ is upper triangular such that $\mathbf{U}'\mathbf{U} = \mathbf{W}$, then

$$(\mathbf{y} - \mathbf{X}\mathbf{b})'\mathbf{W}(\mathbf{y} - \mathbf{X}\mathbf{b}) = (\mathbf{y}^* - \mathbf{X}^*\mathbf{b})'(\mathbf{y}^* - \mathbf{X}^*\mathbf{b}),$$

where $\mathbf{y}^* = \mathbf{U}\mathbf{y}, \mathbf{X}^* = \mathbf{U}\mathbf{X}$. This is now in the form of (1).

Simulation from a **multivariate normal distribution** with $p \times p$ variance–covariance matrix $\boldsymbol{\Sigma} = \mathbf{R}'\mathbf{R}$ may be handled by setting $\mathbf{u} = \mathbf{R}'\mathbf{x}$, where elements of $\mathbf{x}$ are independent normal random deviates each with mean 0 and variance 1.

The Householder QR Algorithm

The Householder QR algorithm has wide application apart from least squares. It is, for example, used in forming the singular value decomposition, which we describe below. It is usually motivated by describing the matrix $\mathbf{Q}$ of (4) as a product of Householder *reflections*

$$\mathbf{I} - \frac{2\mathbf{w}_i\mathbf{w}_i'}{\tau_i}, \quad i = 1, \ldots, p,$$

where $\tau_i = ||\mathbf{w}_i||^2$. The first reflection reduces to zero elements all elements except the first in the initial column of $\mathbf{X}$, replacing the first row of $\mathbf{X}$ by the first row of $\mathbf{R}$. The second takes the matrix so formed and reduces to zero all elements below the second row in its second column, replacing the second row of this matrix with the second row of $\mathbf{R}$. In the adaptation of the Householder method, for which we give algebraic details, one or more rows of $\mathbf{R}$ may differ from the result of applying Householder reflections by a change of sign of all elements in the row [8]. This simplifies the detailed algebraic description and simplifies the algorithm.

Let $\mathbf{x}_j^{(k-1)}$ $(j \geq k)$ be the result of applying rotations $1, \ldots, k - 1$ to column j of $\mathbf{X}$, but with elements $1, \ldots, k - 1$ set to zero when $k > 1$. Then

$$r_{kk} = \left\| \mathbf{x}_k^{(k-1)} \right\|, \qquad r_{kj} = r_{kk}^{-1} \left(\mathbf{x}_k^{(k-1)}\right)' \mathbf{x}_j^{(k-1)}, \quad j > k. \tag{5}$$

This article was published in the *Encyclopedia of Biostatistics* in 2005 by John Wiley & Sons, Ltd.
DOI: 10.1002/0470011815.b2a14022

For elements in rows after the kth we use

$$\mathbf{x}_j^{(k)} = \mathbf{x}_j^{(k-1)} - \alpha_k^{-1}\left(x_{kj}^{(k-1)}\operatorname{sgn}\left(x_{kk}^{(k-1)}\right) + r_{kj}\right)\mathbf{x}_k^{(k-1)}, \quad j > k,$$

where $\alpha_k = |x_{kk}| + r_{kk}$.

Where a column is a linear combination of earlier columns, this leads to $r_{ii} = 0$. The easiest way to deal with this is to move any such column to the final column position. More generally, the columns of $\mathbf{X}$ may be permuted so that columns which are highly dependent on earlier columns are taken last – a device known as *pivoting*. The initial order of columns can, if this is required, be restored when calculations are complete. Additional orthogonal rotations may be required to recover the matrix $\mathbf{R}$ that corresponds to the original ordering.

The Modified Gram–Schmidt QR Algorithm

If the variant of Householder just described is applied to a matrix $\mathbf{X}$ which is augmented with p initial rows of zeros, this leads, essentially, to the modified **Gram–Schmidt** (MGS) algorithm [8]. The MGS algorithm may be described in terms of residuals from repeated regressions. This statistical interpretation is a main reason for mentioning it here.

Let $\mathbf{e}_j^{(k-1)}$ be the vector of residuals when the column $j, j \geq k$, of $\mathbf{X}$ is regressed on columns $1, \ldots, k-1$. Then the MGS algorithm forms

$$r_{kk} = \left\|\mathbf{e}_k^{(k-1)}\right\|, \qquad r_{kj} = r_{kk}^{-1}\left(\mathbf{e}_k^{(k-1)}\right)'\mathbf{e}_j^{(k-1)}.$$

Thus the MGS algorithm uses least squares vectors of residuals to form elements of $\mathbf{R}$. For details see [3, 6–8, 10, 11].

The Givens QR Algorithm

This algorithm operates on $\mathbf{X}$ one row at a time, where Householder and modified Gram–Schmidt operate on columns. It is useful where the QR decomposition must from time to time be updated as new data become available.

The matrix $\mathbf{R}$ is filled initially with zeros. Planar rotations,

$$\begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix}, \tag{6}$$

then rotate rows of $\mathbf{X}$, one at a time, into the upper triangular array. Thus $\mathbf{R}$ is sequentially updated as each new row of $\mathbf{X}$ is rotated into the upper triangular scheme. The rotations which operate on row k ($k > p$) of $\mathbf{X}$ replace y_k with z_k, where z_k^2 is the increase in the residual sum of squares when row k is included. The planar rotations in the Givens QR algorithm are often called Givens rotations.

Another use for planar rotations is to remove rows that were earlier included, i.e. to *downdate* $\mathbf{R}$. A stable algorithm requires access to the matrix $\mathbf{Q}$ [3, 6]. The algorithm in [4] is as stable as possible when $\mathbf{Q}$ is not available.

Orthogonalization of the Columns of X

One way to view the QR method is that it reduces the problem of minimizing $||\mathbf{y} - \mathbf{Xb}||$ to that of minimizing $||\mathbf{y} - \mathbf{X}^*\mathbf{b}^*||$, where $\mathbf{X}^* = \mathbf{XR}^{-1}$ and $\mathbf{b}^* = \mathbf{Rb}$. It replaces $\mathbf{X}$ by a matrix $\mathbf{X}^*$ the columns of which are orthogonal, i.e. $(\mathbf{X}^*)'\mathbf{X}^*$ is the matrix $\mathbf{I} = \operatorname{diag}(1, \ldots, 1)$.

Let

$$\mathbf{Q} = \begin{pmatrix} \mathbf{Q}_1 \\ \mathbf{Q}_2 \end{pmatrix}, \tag{7}$$

where $\mathbf{Q}_1$ is $p \times n$ and $\mathbf{Q}_2$ is $(n-p) \times n$. Then it may be shown that $\mathbf{Q}_1' = \mathbf{XR}^{-1}$. Thus, if $\mathbf{Q}$ is available, the matrix $\mathbf{X}^* = \mathbf{XR}^{-1}$ can be extracted as a submatrix.

Orthogonal Polynomials

Low-order **polynomial** functions are frequently used to provide simple curvilinear models for data. If the covariate $\mathbf{x}$ has elements $x_i, i = 1, \ldots, n$, then calculations can in principle be handled as a least squares calculation in which $\mathbf{X}$ has its (i, j)th element equal to $x_i^{j-1}, i = 1, \ldots, n; j = 1, \ldots, p$. This natural representation of the problem produces a matrix $\mathbf{X}$ the columns of which are likely to be strongly **correlated**. This gives coefficients which are strongly correlated, with standard errors which are inflated by amounts which depend on the correlations.

The QR method may be used as discussed in (7) to form the matrix $\mathbf{X}^*$ with orthogonal columns. The first column of $\mathbf{X}^*$ is a constant, the second is a multiple of $\mathbf{x} - \overline{x}$, the third involves terms up to degree two in $\mathbf{x}$, and so on. Even better is

This article was published in the *Encyclopedia of Biostatistics* in 2005 by John Wiley & Sons, Ltd.
DOI: 10.1002/0470011815.b2a14022

to use recurrence formulas for systems of orthogonal polynomials to generate the columns of $\mathbf{X}^*$ (*see* **Orthogonality**). Such use of orthogonal polynomials gives independent and often more interpretable regression coefficients and avoids numerical instability (*see* **Polynomial Approximation**).

The Deletion and Addition of Columns

Removal of a column of $\mathbf{X}$ is achieved by removing the corresponding column from $\mathbf{R}$ and using a series of Givens rotations to reduce the resulting matrix to upper triangular form. The addition of a further column $\mathbf{x}_{p+1}$ to $\mathbf{X}$ is likewise straightforward, providing $\mathbf{Q}$ is available. A further QR reduction is used to reduce $(\mathbf{R}, \mathbf{Q}\mathbf{x}_{p+1})$ to upper triangular form.

Singular Value Decomposition (SVD)

This decomposition finds application in principal components analysis and in many different related multivariate calculations. It offers yet another approach to least squares calculations [6]. Given an $n \times p$ matrix $\mathbf{X}$, it forms

$$\mathbf{X} = \mathbf{U}\mathbf{G}\mathbf{V}',$$

where $\mathbf{U}$ is $n \times n$ orthogonal, $\mathbf{V}$ is $p \times p$ orthogonal, and $\mathbf{G}$ is $n \times p$ with its only nonzero elements on the uppermost diagonal, namely the singular values.

One or more singular values that are close to zero indicates that $\mathbf{X}$ is near singular, with the relevant linear relations given by the corresponding columns of $\mathbf{V}$. Note that the singular values of $\mathbf{X}'\mathbf{X}$ are the squares of those of $\mathbf{X}$. The Golub–Kahan algorithm for the singular value decomposition first uses Householder reflections to reduce $\mathbf{X}$ to upper bidiagonal form, i.e. all elements are zero except those on the diagonal or in positions immediately above the diagonal. Repeated planar rotations, (6), then reduce the above diagonal elements to zero [6].

Methods for Singular Matrices

Here we examine several technical issues that arise when matrices are singular or close to singular.

Distance from Singularity

Assume that $\mathbf{X}$ has an initial column of ones and that remaining columns are centered. A statistically motivated measure of the distance of column k of $\mathbf{X}$ from a linear combination of all other columns is the inverse $(s_{kk}s^{kk})^{-1}$ of the *variance inflation factor* $s_{kk}s^{kk}$, where s_{kk} and s^{kk} are the kth diagonal elements of $\mathbf{X}'\mathbf{X}$ and $(\mathbf{X}'\mathbf{X})^{-1}$, respectively. This variance inflation factor is the amount by which the standard error of b_k is multiplied because of correlation between column k of $\mathbf{X}$ and other columns. Note the relationship

$$s_{kk}s^{kk} = \left(1 - R^2_{k|1,\ldots,k-1,k+1,\ldots,p}\right)^{-1}, \qquad (8)$$

with the squared multiple correlation $R^2_{k|1,\ldots,k-1,k+1,\ldots,p}$ measuring the dependence of explanatory variable k upon other explanatory variables.

Which are the Linear Dependencies?

Let $\mathbf{r}_k^{(k-1)}$ consist of elements 1 to $k-1$ in column k of $\mathbf{R}$. Let $\mathbf{R}_{11}^{(k-1)}$ be the leading $(k-1) \times (k-1)$ submatrix of $\mathbf{R}$. Then the vector of coefficients in the least squares regression of column k of $\mathbf{X}$ on earlier columns is found by solving for $\mathbf{h}$ in

$$\mathbf{R}_{11}^{(k-1)}\mathbf{h} = \mathbf{r}_k^{(k-1)}.$$

(If $r_{ii} = 0$ for one or more $i < k$, then set $h_i = 0$.)

Suppose that m diagonal elements of $\mathbf{R}$ are zero. Then by determining all such vectors $\mathbf{h}$ we can construct a matrix $\mathbf{H}(p \times m)$ with maximum rank m such that

$$\mathbf{X}\mathbf{H} = \mathbf{0}. \qquad (9)$$

Columns of $\mathbf{H}$ have the form $(h_1, h_2, \ldots, h_{k-1}, -1, 0, \ldots, 0)'$. The columns of $\mathbf{H}$ are a basis for the orthogonal complement of the row space of $\mathbf{X}$. The general solution to the least squares problem is $\tilde{\mathbf{b}} = \mathbf{b} + \mathbf{H}\mathbf{c}$, where $\mathbf{c}$ is arbitrary. One way to make $\tilde{\mathbf{b}}$ unique is to choose $\mathbf{c}$ so that $\tilde{\mathbf{b}}$ has minimum length, which is itself a least squares problem [8], pp. 106–107, 119. The easiest choice is $\mathbf{c} = \mathbf{0}$.

A Reflexive g-inverse of R

Let $\mathbf{R}^-$ be obtained from $\mathbf{R}$ by replacing zero diagonal elements r_{ii} with 1, inverting the resulting matrix, and then placing zeros in the rows and columns where $r_{ii} = 0$. Then

$$\mathbf{R}\mathbf{R}^-\mathbf{R} = \mathbf{R}, \mathbf{R}^-\mathbf{R}\mathbf{R}^- = \mathbf{R}^-,$$

DOI: 10.1002/0470011815.b2a14022

which are the conditions for $\mathbf{R}^-$ to be a *reflexive g-inverse* of $\mathbf{R}$. The matrix $\mathbf{R}^-$ may be used in the calculation of variances and covariances of regression coefficients that correspond to the choice $\mathbf{c} = \mathbf{0}$ above.

Applications

We give a few examples where an elegant alternative to matrix inversion reduces computational effort. In part our aim is to move away from an exclusive focus on least squares.

Leverages and Standard Errors of Residuals

In least squares, the matrix $\mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'$ may be calculated as $\mathbf{X}\mathbf{R}^{-1}(\mathbf{X}\mathbf{R}^{-1})'$. If $(\mathbf{X}\mathbf{R}^{-1})'$ is not already available, it may be determined by solving for columns of $\mathbf{Q}_1$ in the lower triangular system of equations $\mathbf{R}'\mathbf{Q}_1 = \mathbf{X}'$ (cf. (7)). The leverage statistic h_i, which is the ith diagonal element of $\mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'$, may be calculated as the sum of squares of elements of the ith column of $\mathbf{Q}_1$. Note also that $\mathbf{I} - \mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}' = \mathbf{Q}_2'\mathbf{Q}_2$. Thus $\mathbf{Q}_2$ may be used in calculating the variance–covariance matrix of residuals.

Partial Sums of Squares and Products

We show how to form partial correlations between columns of $\mathbf{Y}$ $(n \times q)$, conditional on columns of $\mathbf{X}$ $(n \times p)$. Let $\mathbf{Z} = (\mathbf{1}, \mathbf{X}, \mathbf{Y})$, where $\mathbf{1}$ is a column of ones. Now use the QR algorithm to form

$$\mathbf{QZ} = \begin{pmatrix} \sqrt{n} & \mathbf{u}_1' & \mathbf{u}_2' \\ \mathbf{0} & \mathbf{R}_{XX} & \mathbf{R}_{XY} \\ \mathbf{0} & \mathbf{0} & \mathbf{R}_{YY} \end{pmatrix}, \tag{10}$$

where $\mathbf{u}_1'$ is $1 \times p$, $\mathbf{u}_2'$ is $1 \times q$, $\mathbf{R}_{XX}$ is $p \times p$, $\mathbf{R}_{XY}$ is $p \times q$, and $\mathbf{R}_{YY}$ is $q \times q$.

Then $\mathbf{R}_{YY}'\mathbf{R}_{YY}$ is the matrix of sums of squares and products of the q vectors of residuals from the regressions of columns of $\mathbf{Y}$ on columns of $\mathbf{X}$. The corresponding matrix of partial correlations is $\mathbf{D}^{-1/2}\mathbf{R}_{YY}'\mathbf{R}_{YY}\mathbf{D}^{-1/2}$, where $\mathbf{D}^{-1/2}$ is the diagonal matrix whose elements are the inverses of the square roots of the diagonal elements of $\mathbf{R}_{YY}'\mathbf{R}_{YY}$.

Canonical Correlation

We assume the orthogonal reduction in (10) above. Then computations may be handled by solving the symmetric eigenproblem

$$|\mathbf{R}_{XY}'\mathbf{R}_{XY} - \lambda\mathbf{R}_{YY}'\mathbf{R}_{YY}| = \mathbf{0}. \tag{11}$$

This may be rewritten as

$$|(\mathbf{R}_{XY}\mathbf{R}_{YY}^{-1})'\mathbf{R}_{XY}\mathbf{R}_{YY}^{-1} - \lambda\mathbf{1}| = \mathbf{0},$$

which can be solved by finding the singular value decomposition of $\mathbf{R}_{XY}\mathbf{R}_{YY}^{-1}$. The **canonical correlations** ϕ_i, where i runs from 1 to min [rank $(\mathbf{R}_{XX})$, rank $(\mathbf{R}_{YY})$], are given by

$$\phi_i^2 = \frac{\lambda_i}{1 + \lambda_i}$$

([8], pp. 200–202, 206–208; *see* **Eigenvalue**; **Eigenvector**).

Canonical Variate Analysis

Canonical variate analysis provides a perspective on **multivariate analysis of variance**. Let $\mathbf{Z} = (\mathbf{X}, \mathbf{Y})$, where now columns of $\mathbf{Z}$ specify the groups to which observations belong. Again the orthogonal reduction of $\mathbf{Z}$ to upper triangular form is an effective starting point for further calculations, leading to an eigenproblem of the same form as for canonical correlation [8], pp. 202–203, 208–210.

Matrix Condition Numbers

A matrix *condition number* κ for a matrix $\mathbf{S}$ provides an indication of the relative sensitivity of $\mathbf{Sd}$ or $\mathbf{S}^{-1}\mathbf{d}$ to small relative changes in the elements of $\mathbf{d}$. One possibility is the *spectral condition number* κ_2, which is the ratio $\lambda_{\max}/\lambda_{\min}$ of the largest to smallest eigenvalue of $\mathbf{S}$.

Let $k = \log_{10}\kappa_2(\mathbf{S})$. In general one can expect to lose k digits of accuracy when solving the linear system

$$\mathbf{Sx} = \mathbf{d}$$

for $\mathbf{x}$, or in computing the inverse of $\mathbf{S}$. Note that $\kappa_2(\mathbf{S}) \geq 1$, which means that relative error can never be expected to decrease in solving a linear system. A matrix whose condition number is no more than

This article was published in the *Encyclopedia of Biostatistics* in 2005 by John Wiley & Sons, Ltd.
DOI: 10.1002/0470011815.b2a14022

10 or 100 is, from a computational perspective, well-conditioned.

Numerical and Statistical Measures of Conditioning

Let κ_2 be the spectral condition number of the correlation matrix derived from $\mathbf{X}'\mathbf{X}$. Then

$$\max_{1\le i\le p}(s_{ii}s^{ii}) \le \kappa_2 \le \sum_{i=1}^{p} s_{ii}s^{ii},$$

where s_{ii} and s^{ii} are defined as in (8). This makes a connection between statistical and numerical measures of conditioning [2, 8], p. 211. The quantity $s_{ii}s^{ii}$ has the benefit that, unlike matrix condition numbers such as κ_2, it is independent of scale.

Note that determination of the minimum value $\min_{\mathbf{b}} ||\mathbf{y} - \mathbf{X}\mathbf{b}||^2$ is well-conditioned, even if $\mathbf{X}$ is singular.

Components of Larger Computations

The notes on computational methods in [5] demonstrate extensive use of matrix calculations as building blocks for a wide variety of other statistical calculations, analysis of variance with multiple error strata (*see* **Multilevel Models**), **generalized linear models** (GLMs), **generalized additive models** (GAMs), local regression smoothing (loess) (*see* **Graphical Displays**), and **nonparametric regression**; see also [11]. New complications are inevitable as matrix computational methods are used to extend the range of models available to statisticians. In models where a variance–covariance structure must be estimated, the notion of a singularity has subtleties beyond those of ordinary least squares.

Software

Many statistical packages allow the user to specify calculations as a sequence of matrix operations. SAS (in the IML Interactive Matrix Language module), SPSS (MATRIX language), STATA, **S-PLUS, R**, and Genstat are some of the statistical systems which have extensive matrix computational abilities. Statistical packages have generally stayed with normal equation methods. S-PLUS and R make extensive use of modern methods such as QR. Note also the extensive modern matrix abilities in the mathematically oriented languages of MATLAB, Gauss, and Mathematica. MATLAB has been used extensively by numerical analysts [7].

The FORTRAN subroutine package LAPACK [1], and earlier packages LINPACK, and EISPACK from which LAPACK is derived, provide high-quality software to perform calculations referred to in this article. These packages are publicly available from the NETLIB online database, and are also part of the NAG and IMSL subroutine libraries.

References

[1] Anderson, E., Bai, Z., Bischof, C., Blaxckfdord, S., Demmel, J., Dongarra, J., Du Croz, J., Greenbaum, A., Hammarling, S., McKenney, A. & Sorenen, D. (1999). *LAPACK Users' Guide*, 3rd Edn., SIAM, Philadelphia.

[2] Berk, K.N. (1977). Tolerance and condition in regression equations, *Journal of the American Statistical Association* **72**, 863–866.

[3] Björck, A. (1996). *Numerical Methods for Least Squares Problems*. SIAM, Philadelphia.

[4] Bojanczyk, A.W., Brent, R.P., van Dooren, P. & de Hoog, F.R. (1987). A note on downdating the Cholesky factorization, *SIAM Journal of Scientific and Statistical Computation* **8**, 210–221.

[5] Chambers, J.M. & Hastie, T.J. (1991). *Statistical Models in S*. Wadsworth and Brooks/Cole, Pacific Grove.

[6] Golub, G.H. & Van Loan, C.F. (1996). *Matrix Computations*, 3rd Ed. Johns Hopkins University Press, Baltimore.

[7] Higham, N.J. (1996). *Accuracy and Stability of Numerical Algorithms*. SIAM, Philadelphia.

[8] Maindonald, J.H. (1984). *Statistical Computation*. Wiley, New York.

[9] Stewart, G.W. (1982). Linear algebra, computational, in *Encyclopedia of Statistical Sciences*, Vol. 5, S. Kotz, N.L. Johnson & C.B. Read, eds. Wiley, New York, pp. 5–19.

[10] Stoer, J. & Bulirsch, R. (1992). *Introduction to Numerical Analysis*, 2nd Ed. Springer-Verlag, New York.

[11] Thisted, R.A. (1988). *Elements of Statistical Computation. Numerical Computation*. Chapman & Hall, New York.

John H. Maindonald & Gordon K. Smyth

Encyclopedia of Biostatistics, Online © 2005 John Wiley & Sons, Ltd.
This article is © 2005 John Wiley & Sons, Ltd.
This article was published in the *Encyclopedia of Biostatistics* in 2005 by John Wiley & Sons, Ltd.
DOI: 10.1002/0470011815.b2a14022