

## 3.5. MULTIPLE LINEAR REGRESSION

[THE PROBLEM](#)

[NUMERICALLY STABLE ALGORITHMS](#)

[THE MODIFIED GRAM-SCHMIDT METHOD](#)

[HOUSEHOLDER TRANSFORMATION](#)

[SINGULAR VALUE DECOMPOSITION](#)

[GENERALIZED INVERSES](#)

[CHOLESKY FACTORISATION](#)

[USE OF MATLAB](#)

[References](#)

### THE PROBLEM

Let  $n$  observations of a dependent variable  $y$  and explanatory variables  $x_1, \dots, x_k$  stored in a vector  $Y$  and a matrix  $X$ :

$$Y = \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix}$$

$$X = \begin{bmatrix} x_{11} & \cdots & x_{k1} \\ \vdots & & \vdots \\ x_{1n} & \cdots & x_{kn} \end{bmatrix}$$

The model is described by

$$y = \beta_1 x_1 + \dots + \beta_k x_k + e,$$

The  $\beta_j$  are the regression coefficients, which should be estimate from the data, and  $e$ , is the error term.

The least squares estimator is the solution of the system of *normal equations*, in matrix form

$$\hat{\beta} = (X^T X)^{-1} X^T Y. \quad (1)$$

From the numerical point of view, there is a double problem:

1° the use of  $X^T X$  and  $X^T Y$  in (1) corresponds implicitly to the "calculator" algorithm (see section 2.1) to calculate the variances and covariances but we saw that that algorithm is not numerically stable.

2° inversion of the matrix  $X^T X$  can raise numerical problems if the matrix is "almost noninvertible" (case of quasi-colinearity).

*Notice.* A simple way to improve the precision is to center the variables before carrying out the calculations, provided that the initial model has a constant.

*Example.* Let us suppose calculations with a precision  $\varepsilon$  such as  $1 + \varepsilon^2 \approx 1$  (for example, in single precision, with  $\varepsilon = 10^{-7}$ ) and the matrix

$$X = \begin{bmatrix} 1 & 1 \\ \varepsilon & 0 \\ 0 & \varepsilon \\ 0 & 0 \end{bmatrix}$$

so that

$$X^T X = \begin{bmatrix} 1 + \varepsilon^2 & 1 \\ 1 & 1 + \varepsilon^2 \end{bmatrix} \approx \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$$

is a non invertible matrix.

Numerically stable algorithms for multiple linear regression

1° don't calculate  $X^T X$  ;

2° a fortiori, don't invert  $X^T X$ .

### *NUMERICALLY STABLE ALGORITHMS*

They are based on orthogonal transformations and a factorisation of matrix  $X$  :

1° the modified Gram-Schmidt method, an alternative of a well-known method in linear algebra;

2° the Householder transformation

3° the singular value decomposition

The Givens method can also be used, particularly when the data are read in one by one.

We add the Cholesky factorisation, which is interesting for other reasons.

One can find all these algorithms in the LINPACK library.

For recent references, see Gentle (1998).

*Remark.* The least squares method can be introduced without using the normal equations, for example Thisted [1988, p. 63].

## *THE MODIFIED GRAM-SCHMIDT METHOD*

### **Description**

*Example.*

### **Indirect solution of the normal equations**

### **Implementation**

## Description

Principle: build an orthonormalised basis in a metric space in the  $n$ -dimensional space generated by the columns of  $X$

Classical Gram-Schmidt method (Chambers, 1977):

1° as first base vector take the first column of  $X$ , duly normalised

2° as second base vector take a vector orthogonal with the first base vector, in the space generated by the first two columns: it is obtained by taking the residual of the orthogonal projection of the second column on the first, followed by a normalisation

3° and so on, the  $i$ -th vector being built in the space of first  $i$  columns, orthogonal with  $(i - 1)$  first base vectors and being normalised.

The  $i$ -th column of  $X$  can be written as a linear combination of the elements of that orthonormalised base using the first  $(i - 1)$  elements only.

The factorisation of  $X$  has the following form:

$$\begin{array}{ccc} X & = & Q \cdot R \\ n \times k & & n \times n \quad n \times k \end{array}$$

where

$Q$  is an orthogonal matrix, such that  $Q^T = Q^{-1}$ ,

$R = \begin{bmatrix} R_1 \\ 0 \end{bmatrix}$ , where  $R_1$  is a upper triangular matrix.

*Example.*

$$X = \begin{bmatrix} 1 & 1 \\ \varepsilon & 0 \\ 0 & \varepsilon \\ 0 & 0 \end{bmatrix}$$

First

$$x_1 = \begin{bmatrix} 1 \\ \varepsilon \\ 0 \\ 0 \end{bmatrix} \Rightarrow \|x_1\| = \sqrt{1 + \varepsilon^2} \approx 1$$

$$\Rightarrow q_1 = \begin{bmatrix} 1 \\ \varepsilon \\ 0 \\ 0 \end{bmatrix}$$

$$\Rightarrow x_1 = q_1.$$



Denoting the scalar product by  $\langle \cdot, \cdot \rangle$ :

$$x_2 = \begin{bmatrix} 1 \\ 0 \\ \varepsilon \\ 0 \end{bmatrix} \Rightarrow \langle x_2, q_1 \rangle = \begin{bmatrix} 1 & 0 & \varepsilon & 0 \end{bmatrix} \begin{bmatrix} 1 \\ \varepsilon \\ 0 \\ 0 \end{bmatrix} = 1$$

$$w_2 = x_2 - \langle x_2, q_1 \rangle q_1 = \begin{bmatrix} 1 \\ 0 \\ \varepsilon \\ 0 \end{bmatrix} - \begin{bmatrix} 1 \\ \varepsilon \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ -\varepsilon \\ \varepsilon \\ 0 \end{bmatrix}$$

$$\Rightarrow \|w_2\| = \sqrt{2\varepsilon^2} = \sqrt{2}\varepsilon$$

$$q_2 = \frac{w_2}{\sqrt{2}\varepsilon} = \begin{bmatrix} 0 \\ -1/\sqrt{2} \\ 1/\sqrt{2} \\ 0 \end{bmatrix} \Rightarrow x_2 = q_1 + \sqrt{2}\varepsilon q_2$$

hence the factorisation

$$X = \begin{bmatrix} 1 & 1 \\ \varepsilon & 0 \\ 0 & \varepsilon \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 & \bullet & \bullet \\ \varepsilon & -1/\sqrt{2} & \bullet & \bullet \\ 0 & 1/\sqrt{2} & \bullet & \bullet \\ 0 & 0 & \bullet & \bullet \end{bmatrix} \cdot \begin{bmatrix} 1 & 1 \\ 0 & \sqrt{2}\varepsilon \\ 0 & 0 \\ 0 & 0 \end{bmatrix}.$$

### Indirect solution of the normal equations

A solution is obtained by solving the triangular system

$$R\hat{\beta} = Q^T Y.$$

Indeed, left multiply by  $R^T$ :

$$R^T R\hat{\beta} = R^T Q^T Y,$$

hence

$$\hat{\beta} = (R^T R)^{-1} R^T Q^T Y = (R^T Q^{-1} Q R)^{-1} R^T Q^T Y = (R^T Q^T Q R)^{-1} R^T Q^T Y = (X^T X)^{-1} X^T Y.$$

*Example.*

$$X = \begin{bmatrix} 1 & 1 \\ \varepsilon & 0 \\ 0 & \varepsilon \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 & \bullet & \bullet \\ \varepsilon & -1/\sqrt{2} & \bullet & \bullet \\ 0 & 1/\sqrt{2} & \bullet & \bullet \\ 0 & 0 & \bullet & \bullet \end{bmatrix} \cdot \begin{bmatrix} 1 & 1 \\ 0 & \sqrt{2}\varepsilon \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$$

Consequently,

$$Q^T Y = \begin{bmatrix} 1 & \varepsilon & 0 & 0 \\ 0 & -1/\sqrt{2} & 1/\sqrt{2} & 0 \\ \bullet & \bullet & \bullet & \bullet \\ \bullet & \bullet & \bullet & \bullet \end{bmatrix} \cdot \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix} = \begin{bmatrix} y_1 + \varepsilon y_2 \\ -1/\sqrt{2}y_2 + 1/\sqrt{2}y_3 \\ \bullet \\ \bullet \end{bmatrix}$$

hence

$$R\hat{\beta} = \begin{bmatrix} 1 & 1 \\ 0 & \sqrt{2}\varepsilon \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} \hat{\beta}_1 \\ \hat{\beta}_2 \end{bmatrix} = \begin{bmatrix} y_1 + \varepsilon y_2 \\ -1/\sqrt{2}y_2 + 1/\sqrt{2}y_3 \\ \bullet \\ \bullet \end{bmatrix}$$

and finally

$$\begin{cases} \hat{\beta}_1 + \hat{\beta}_2 = y_1 + \varepsilon y_2 \\ \sqrt{2}\varepsilon\hat{\beta}_2 = -\frac{1}{\sqrt{2}}y_2 + \frac{1}{\sqrt{2}}y_3 \end{cases}.$$

This is an easy to solve a triangular system

The solution is obtained without loss of precision

*Remark.* 1. There is an inconvenience: the whole matrix  $X$  should be in memory, as well as  $Q$  (more precisely  $k$  rows). However, it is possible to add an explanatory variable without doing all the calculations.

2. A problem is the lack of orthogonality of  $Q$  when  $X$  is badly conditioned. It should be orthogonalised. This is the reason for the modification in the modified Gram-Schmidt method.

## Implementation

Here is MGSREG (from TSE, see Mélard et Pasteels [1994]) which implements the modified Gram-Schmidt method for multiple linear regression.

[Mgsreg.htm](#)

## *HOUSEHOLDER TRANSFORMATION*

### 2-dimensional case

### k dimensional case

### Householder transformation of a matrix

### Example.

### Numerical aspects

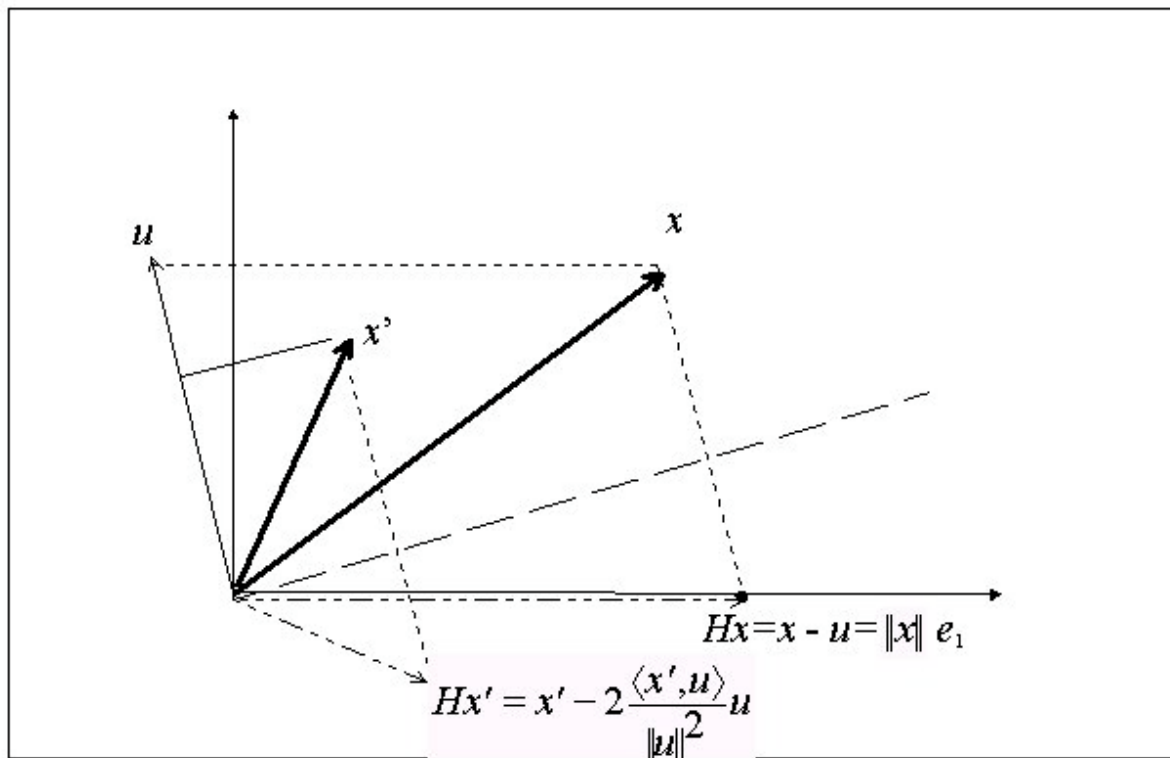
### Application to regression

The Householder transformation is a reflection, i. e. a symmetry with respect to a hyperplan, sending a vector in a lower-dimensional space.

## 2-dimensional case

Let  $e_1, e_2$  an orthonormal basis. We want a reflection  $H$  which maps vector  $x$  on  $Hx$  on the first axis, e. g.  $Hx = \|x\| e_1$ . It suffices to take vector  $u = x - \|x\| e_1$ , and perform a symmetry with respect to the straight line orthogonal to  $u$ . Hence  $Hx = x - u$ . Remark that the solution is not unique. More precisely, there is a second solution for which  $Hx = -\|x\| e_1$ , defined by  $u = x + \|x\| e_1$ . An arbitrary vector,  $x'$ , will not be mapped on the first axis and  $Hx' \neq x' - u$ .

Geometrically, it can be seen that  $Hx' = x' - 2 \frac{\langle x', u \rangle}{\|u\|^2} u$ .



***k* dimensional case**

Consider  $x = (x_1 \dots x_n)^T$  in an orthonormalised basis. We are looking for transformation  $H$  which

1° preserves the first  $j - 1$  components of  $x$ ;

2° put zeros in the last  $n - j$  components;

3° preserves the norm.

More precisely

$$H: x = \begin{bmatrix} x_1 \\ \vdots \\ x_{j-1} \\ x_j \\ x_{j+1} \\ \vdots \\ x_n \end{bmatrix} \rightarrow Hx = \begin{bmatrix} x_1 \\ \vdots \\ x_{j-1} \\ \pm s_j \\ 0 \\ \vdots \\ 0 \end{bmatrix},$$

where  $s_j = \sqrt{x_j^2 + \dots + x_n^2}$ .



The reflection is a symmetry with respect to the hyperplan which is orthogonal to vector

$$u = x - Hx = \begin{bmatrix} x_1 \\ \vdots \\ x_{j-1} \\ x_j \\ x_{j+1} \\ \vdots \\ x_n \end{bmatrix} - \begin{bmatrix} x_1 \\ \vdots \\ x_{j-1} \\ \pm s_j \\ 0 \\ \vdots \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ x_j \mp s_j \\ x_{j+1} \\ \vdots \\ x_n \end{bmatrix}.$$

More generally, if  $x'$  is an arbitrary vector, the orthogonal projection of  $x'$  onto  $u$  should be determined, and

$$Hx' = x' - 2 \frac{\langle x', u \rangle}{\|u\|^2} u = x' - 2 \frac{u^T x'}{u^T u} u = \left( I - 2 \frac{uu^T}{u^T u} \right) x',$$

because  $u^T x' u = u u^T x'$ .

That expression defines  $H$  as a matrix:  $H = I - 2 \frac{uu^T}{u^T u}$

Remark that  $H = H^T$ .

The reflection is an isometry since

$$HH^T = HH = \left( I - 2 \frac{uu^T}{u^T u} \right) \left( I - 2 \frac{uu^T}{u^T u} \right) = \left( I + 4 \frac{uu^T uu^T}{\|u\|^4} - 4 \frac{uu^T}{\|u\|^2} \right) = I,$$

noting that scalar  $u^T u$  equals  $\|u\|^2$ . Consequently  $H = H^T = H^{-1}$  is a symmetric matrix.

*Remark.* For  $x' = u + Hx$ , we have  $u^T x' = \mp s_j x_j + x_j^2 + \dots + x_n^2 = \mp s_j x_j + s_j^2$ .

But  $\|u\|^2 = (x_j \mp s_j)^2 + x_{j+1}^2 + \dots + x_n^2 = 2(\mp s_j x_j + s_j^2)$ .

Hence  $u^T x' / \|u\|^2 = 1/2$  and, after simplification  $Hx' = x' - u$ .

*Remark.* Householder reflection is also used to transform a symmetric matrix in tridiagonal form. This is the starting point of an algorithm for computing eigenvalues and eigenvectors (Press *et al.*, 1993).

### Householder transformation of a matrix

Consider a matrix  $X$ , with dimensions  $n \times k$ , composed of columns  $[X_1 \dots X_k]$

We apply successively  $k$  reflections  $H_j$ , where the  $j$ -th reflection puts zeros in the last  $n - j$  components.

As the following reflections  $H_{j+1}, \dots, H_k$  will preserve the first  $j, \dots, k - 1$  components, zeros will be preserved.

Consequently

$$R^* \stackrel{\text{d\u00e9f}}{=} H_k H_{k-1} \dots H_1 X = \begin{bmatrix} * & * & * & & \\ 0 & * & * & & \\ & 0 & * & \dots & \\ & & 0 & & \\ \vdots & \vdots & \vdots & & \\ 0 & 0 & 0 & & \end{bmatrix}.$$

Note that the product  $H_k H_{k-1} \dots H_1 = Q^{*T}$  is an orthogonal matrix and a factorisation  $X = Q^* R^*$ , has been obtained which is similar but not identical to the  $QR$  factorisation obtained in the modified Gram-Schmidt method.

*Example.*

$$X = [X_1 \quad X_2] = \begin{bmatrix} x_{11} & x_{12} \\ x_{21} & x_{22} \\ x_{31} & x_{32} \\ x_{41} & x_{42} \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ \varepsilon & 0 \\ 0 & \varepsilon \\ 0 & 0 \end{bmatrix}$$

$$s_1^2 = 1 + \varepsilon^2 \approx 1$$

$$\Rightarrow u_1 = \begin{bmatrix} x_{11} \mp s_1 \\ x_{21} \\ x_{31} \\ x_{41} \end{bmatrix} = \begin{bmatrix} 2 \\ \varepsilon \\ 0 \\ 0 \end{bmatrix} \Rightarrow \|u_1\|^2 = 4 + \varepsilon^2 \approx 4$$

$$H_1 X_1 = \begin{bmatrix} \pm s_1 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} -1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$u_1^T X_2 = [2 \quad \varepsilon \quad 0 \quad 0] \cdot \begin{bmatrix} 1 \\ 0 \\ \varepsilon \\ 0 \end{bmatrix} = 2$$

$$H_1 X_2 = X_2 - 2 \frac{2}{4} u_1 = \begin{bmatrix} -1 \\ -\varepsilon \\ \varepsilon \\ 0 \end{bmatrix} \Rightarrow u_1^T X_1 = 2 + \varepsilon^2 \approx 2.$$

$$H_1 X_2 = X_2 - 2 \frac{2}{4} u_1 = \begin{bmatrix} 1 \\ 0 \\ \varepsilon \\ 0 \end{bmatrix} - \begin{bmatrix} 2 \\ \varepsilon \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} -1 \\ -\varepsilon \\ \varepsilon \\ 0 \end{bmatrix}$$

$$H_2 H_1 X_1 = H_1 X_1 = \begin{bmatrix} -1 \\ 0 \\ 0 \\ 0 \end{bmatrix}.$$

$$H_2 H_1 X_2 = ?$$

$$s_2^2 = \varepsilon^2 + \varepsilon^2 \approx 2\varepsilon^2 \Rightarrow u_2 = \begin{bmatrix} 0 \\ -\varepsilon + \sqrt{2}\varepsilon \\ \varepsilon \\ 0 \end{bmatrix}.$$

$$H_2 H_1 X_2 = \begin{bmatrix} -1 \\ -\sqrt{2}\varepsilon \\ 0 \\ 0 \end{bmatrix}.$$

$$\Rightarrow R^* = H_2 H_1 X = \begin{bmatrix} -1 & -1 \\ 0 & -\sqrt{2}\varepsilon \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$$

The coefficients of  $R^*$  are those of  $R$ , in absolute value.

## Numerical aspects

1. Multiplicity of the solutions are due to the choice of signs  $\pm$  at each reflection. In the example, the first sign must be - to avoid division by 0. It is generally recommended, to improve numerical stability, to select for  $s_j$  the sign of  $x_j$ . In the example, the second time, we should have taken the upper sign, producing the following changes:

$H_2H_1X_2.= ?$

$$s_2^2 = \varepsilon^2 + \varepsilon^2 \approx 2\varepsilon^2 \Rightarrow u_2 = \begin{bmatrix} 0 \\ -\varepsilon - \sqrt{2}\varepsilon \\ \varepsilon \\ 0 \end{bmatrix}.$$

$$H_2H_1X_2.= \begin{bmatrix} -1 \\ \sqrt{2}\varepsilon \\ 0 \\ 0 \end{bmatrix}.$$

$$\Rightarrow R^* = H_2H_1X := \begin{bmatrix} -1 & -1 \\ 0 & \sqrt{2}\varepsilon \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$$

2. If  $s_j = 0$ , then  $\|u_j\| = 0$  and a division by 0 will occur. This is a case of multicollinearity. In that case, perform a permutation of rows and columns in order to have rows of zeros at the positions  $r + 1$  to  $k$  of  $R^*$ .
3. Numerical stability can be bad in certain circumstances. In order to improve it, a permutation of rows can be done to improve conditioning. In column  $j$ , the pivot is determined, i. e. the maximum among rows  $j$  to  $n$ . If it is located on row  $l > j$ , let us permute rows  $l$  and  $j$ . In practice, it is not necessary to perform physically the permutations of rows and columns, just use pointers.

### Application to regression

It is better to perform Householder reflections to matrix  $[X_1 \dots X_k Y]$  obtained by juxtaposing column  $Y$ .

Regression with a constant consists, as we know, to consider a column of 1 in  $X$ .

Suppose that it is the first column.

Since  $s_1^2 = n$ , and we need to preserve the norm of  $X_1$ , we have

$$H_1 X_1 = \begin{bmatrix} \sqrt{n} \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$
$$u_1 = \begin{bmatrix} 1 - \sqrt{n} \\ 1 \\ \vdots \\ 1 \end{bmatrix}$$



Effect of  $H_1$  on the other columns, for example on  $Y$ ?

Given that  $\|u_1\|^2 = 2\sqrt{n}(\sqrt{n} - 1)$  and  $u_1^T Y = \sum y_i - \sqrt{n}y_1$ , we have

$$H_1 Y = Y - 2 \frac{\sum y_i - \sqrt{n}y_1}{2\sqrt{n}(\sqrt{n} - 1)} \begin{bmatrix} 1 - \sqrt{n} \\ 1 \\ \vdots \\ 1 \end{bmatrix} = \begin{bmatrix} (\sum y_i) / \sqrt{n} \\ * \\ \vdots \\ * \end{bmatrix}$$

where we know that the sum of squares of the other elements is equal to

$$\sum y_i^2 - \left( \frac{\sum y_i}{\sqrt{n}} \right)^2 = ns_y^2.$$

## *SINGULAR VALUE DECOMPOSITION*

[Definition](#)

[Properties](#)

[Algorithm](#)

[Use in multiple regression](#)

**Definition**

The SVD concerns an arbitrary matrix  $X$  with dimensions  $n \times k$ . Suppose that  $n > k$ . There exists always two orthogonal matrices  $U$ ,  $n \times n$ , et  $V$ ,  $k \times k$ , and a matrix  $D$ ,  $n \times k$ ,  $D^T = (D^*{}^T \ O)$  where  $D^*$  is a  $k \times k$  diagonal matrix, with diagonal elements  $d_i \geq 0$ , such that

$$U^T X V = D = \begin{bmatrix} D^* \\ O \end{bmatrix}.$$

The factorisation can be written  $X = U D V^T$ .

## Properties

In the special case where  $n = k$ , and matrix  $X$  is square,  $A$ , the singular values are the eigenvalues.

The non-zero singular values of  $X$  are the square roots of the non zero eigenvalues of the symmetric matrix  $A = X^T X$ .

For a symmetric positive definite matrix  $A$ , the  $d_i$  are the eigenvalues of  $A$ ,  $U = V$ , and the columns of  $V$  are the eigenvectors of  $A$ . Then:  $A = V D V^T$ .

The ratio  $\max\{d_i\}/\min\{d_i\}$  is a measure of the degree of conditioning of the matrix.

The singular value decomposition is very important from the theoretical point of view but also from the numerical point of view.

It allows to detect the rank deficiency of a matrix with a higher precision (QR factorisation can also be used but it is less accurate).

The SVD also allows to determine the null space of a matrix.

Suppose that the  $n \times k$ ,  $n > k$ , matrix  $X$  with rank  $r$ , is such that the singular values are  $d_1 \geq \dots \geq d_r \geq d_{r+1} = \dots = d_k = 0$ .

The null space of  $X$ , which is the set of vectors  $x$  such that  $X x = 0$ , is then the subspace spanned by the column vectors  $v_{r+1}, \dots, v_k$  of  $V$ .

The space spanned by the column vectors of  $X$  is identical to the one spanned by the first  $r$  columns of  $U$ :  $u_1, \dots, u_r$ .

That factorisation has another advantage: it leads to the definition of a function of a matrix.

For example, the square root of a symmetric positive definite matrix  $A = V D V^T$  is defined by  $A^{1/2} = V D^{1/2} V^T$ , where  $D^{1/2}$  is the diagonal matrix composed of the square roots of the elements of  $D$ , the eigenvalues of  $A$ .

Similarly,  $e^A$  can be defined. Note also  $A^2 = V D V^T V D V^T = V D^2 V^T$  or  $A^{-1} = V D^{-1} V^T$  (supposing strictly positive  $d_i$ 's) are compatible with the usual definition.

## Algorithm

The algorithm of Golub-Reinsch is too complex to be given here. See Golub and Van Loan [1983]. It makes use of Householder reflections to transform the matrix in a tridiagonal matrix. It is included in library EISPACK.

## Use in multiple regression

We have  $X^T X = V D U^T U D V^T = V D^2 V^T$ . Furthermore the  $d_i^2$  are eigenvalues of  $X^T X$  with corresponding eigenvector  $v_i$ ,  $i$ -th column of  $V$ . Supposing  $d_1 \geq \dots \geq d_r \geq d_{r+1} = \dots = d_k = 0$ , it can be shown that

$$\hat{\beta} = \sum_{i=1}^r \frac{u_i^T y}{d_i} v_i.$$

*Remark.* In the case of multicollinearity, that approach can be used to determine a generalised inverse.

## GENERALIZED INVERSES

### Definition

### Application to regression

#### **Definition**

Let a  $n \times k$  matrix  $A$ . Define the ( $g_1$ -) inverse of  $A$ , every matrix  $A^-$  such that  $AA^-A = A$ . Matrix  $A^-$  is not unique.

*Remark.* There are other definitions of a pseudo-inverse  $A^+$  of a matrix  $A$ . Consider the four properties:

1.  $AA^+A = A$
2.  $A^+AA^+ = A^+$
3.  $(AA^+)^T = AA^+$
4.  $(A^+A)^T = A^+A$

A  $g_i$ -inverse fulfils properties 1 to  $i$ .

A  $g_4$ -inverse is also called a Moore-Penrose inverse. Such a matrix is unique.

**Block inverse**

Suppose a matrix of rank  $r$ .

By means of a permutation of rows and columns, it can be put under the following form

$$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}$$

so that  $A_{11}$  is invertible. Let us show that matrix

$$A^- = \begin{bmatrix} A_{11}^{-1} & 0 \\ 0 & 0 \end{bmatrix}$$

is a generalised inverse. Indeed, since  $A$  and  $A_{11}$  are both rank  $r$  matrices, there exists a matrix  $K$  such that  $[A_{21} \ A_{22}] = K [A_{11} \ A_{12}]$  and  $K = A_{21} A_{11}^{-1}$ , hence  $A_{22} = A_{21} A_{11}^{-1} A_{12}$ . Now

$$AA^-A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} A_{11}^{-1} & 0 \\ 0 & 0 \end{bmatrix} A = \begin{bmatrix} I & 0 \\ A_{21}A_{11}^{-1} & 0 \end{bmatrix} \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{21}A_{11}^{-1}A_{12} = A_{22} \end{bmatrix} = A.$$

### Application to regression

In the case where  $X$  is not of maximum rank, there is a phenomenon of multicollinearity. The least squares method has then an infinite number of solutions.

One of these solutions can be written in terms of a generalised inverse:  $\hat{\beta} = (X^T X)^- X^T Y$ .

This is not an estimator of  $\beta$ .

However, an estimator of  $E(y) = x \beta$  is determined uniquely by prediction  $x \hat{\beta}$ , in spite of multicollinearity.

More generally an estimable function is a function of the parameters which can be uniquely estimated.



*Example.* Suppose that  $y = m + \alpha_1 x_1 + \alpha_2 x_2 + e$ , and  $X$  has the form:

$$X = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \\ 1 & 0 & 1 \\ 1 & 0 & 1 \end{bmatrix},$$

so that

$$X^T X = \begin{bmatrix} 6 & 3 & 3 \\ 3 & 3 & 0 \\ 3 & 0 & 3 \end{bmatrix}.$$

Let us compute  $(X^T X)^-$ . The matrix has rank 2. Since

$$\begin{bmatrix} 3 & 0 \\ 0 & 3 \end{bmatrix}^{-1} = \begin{bmatrix} \frac{1}{3} & 0 \\ 0 & \frac{1}{3} \end{bmatrix}$$

, we have

$$(X^T X)^- = \begin{bmatrix} 0 & 0 & 0 \\ 0 & \frac{1}{3} & 0 \\ 0 & 0 & \frac{1}{3} \end{bmatrix}$$

hence

$$\hat{\beta} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & \frac{1}{3} & 0 \\ 0 & 0 & \frac{1}{3} \end{bmatrix} \cdot \begin{bmatrix} y_1 + \dots + y_6 \\ y_1 + y_2 + y_3 \\ y_4 + y_5 + y_6 \end{bmatrix} = \begin{bmatrix} 0 \\ \frac{1}{3}(y_1 + y_2 + y_3) \\ \frac{1}{3}(y_4 + y_5 + y_6) \end{bmatrix}.$$

Note that the first and second elements of  $\hat{\beta}$  estimate uniquely  $m + \alpha_1$  and  $m + \alpha_2$ , respectively, hence  $\alpha_1 - \alpha_2$  is estimated by  $(y_1 + y_2 + y_3 - y_4 - y_5 - y_6) / 3$ , in a unique way. These are three estimable functions.

It is well known that omitting the column of 1's in matrix  $X$ , or imposing constraint  $\alpha_1 + \alpha_2$ , to suppress multicollinearity.

## CHOLESKY FACTORISATION

### Definition

### Algorithm

### square root free variant

#### **Definition**

Let us factorise a symmetric definite positive matrix  $A$ . It is such that the quadratic form  $v^T A v > 0$  for all vector  $v$  of  $\mathfrak{R}^k$  such that  $v \neq 0$ .

Such a matrix has all its eigenvalues strictly positive.

$A$  is factorised as a product of a lower triangular matrix  $L$ , by its transposed, a upper triangular matrix:  $A = LU = LL^T$ .

A square root free Cholesky factorisation is:  $A = L^*D^*U^* = L^*DL^{*T}$ , where matrix  $D^*$  is diagonal and the main diagonal of  $L^*$  is composed of 1.

We can then factorise  $L$  as  $L = L^*D$  and define  $D^* = DD^T = D^2$ .

## Algorithm

Let us start with a 3×3 example:

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ \bullet & a_{22} & a_{23} \\ \bullet & \bullet & a_{33} \end{bmatrix} = \begin{bmatrix} l_{11} & 0 & 0 \\ l_{21} & l_{22} & 0 \\ l_{31} & l_{32} & l_{33} \end{bmatrix} \begin{bmatrix} l_{11} & l_{21} & l_{31} \\ 0 & l_{22} & l_{32} \\ 0 & 0 & l_{33} \end{bmatrix}.$$

The following relations are used, in the given order:

$$\begin{aligned} a_{11} &= l_{11}^2 & l_{11} &= \sqrt{a_{11}} \\ a_{12} &= l_{11}l_{21} & l_{21} &= a_{12} / l_{11} \\ a_{22} &= l_{21}^2 + l_{22}^2 & l_{22} &= \sqrt{a_{22} - l_{21}^2} \\ a_{13} &= l_{11}l_{31} & l_{31} &= a_{13} / l_{11} \\ a_{23} &= l_{21}l_{31} + l_{22}l_{32} & l_{32} &= \frac{a_{23} - l_{21}l_{31}}{l_{22}} \\ a_{33} &= l_{31}^2 + l_{32}^2 + l_{33}^2 & l_{33} &= \sqrt{a_{33} - l_{31}^2 - l_{32}^2} \end{aligned}$$

which can be summarised by

$$\begin{cases} l_{ik} = \frac{a_{ki} - \sum_{j=1}^{k-1} l_{kj}l_{ij}}{l_{kk}}, & k < i \\ l_{ii}^2 = a_{ii} - \sum_{j=1}^{i-1} l_{ij}^2 \end{cases}$$

*Remark.* In order to determine a generalised inverse of  $A$ , the Cholesky factorisation can be used by controlling the denominator of the ratio. There is numerical instability if

$$|d_{ii}| = \left| a_{ii} - \sum_{j=1}^{i-1} l_{ij}^2 \right| < \eta |a_{ii}|,$$

where  $\eta$  depends on the machine precision (for example  $10^{-5}$ , in simple precision). Then  $l_{ii}$  is replaced by 0. For an implementation, see Healy [1968a, 1968b].

*Remark.* The square root free variant is as follows:

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ \bullet & a_{22} & a_{23} \\ \bullet & \bullet & a_{33} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ l_{21}^* & 1 & 0 \\ l_{31}^* & l_{32}^* & 1 \end{bmatrix} \begin{bmatrix} d_{11}^* & 0 & 0 \\ 0 & d_{22}^* & 0 \\ 0 & 0 & d_{33}^* \end{bmatrix} \begin{bmatrix} 1 & l_{21}^* & l_{31}^* \\ 0 & 1 & l_{32}^* \\ 0 & 0 & 1 \end{bmatrix}.$$

using the relations:

$$\begin{aligned} a_{11} &= d_{11}^* & d_{11}^* &= a_{11} \\ a_{12} &= d_{11}^* l_{21}^* & l_{21}^* &= a_{12} / d_{11}^* \\ a_{22} &= d_{11}^* l_{21}^{*2} + d_{22}^* & d_{22}^* &= a_{22} - d_{11}^* l_{21}^{*2} \\ a_{13} &= d_{11}^* l_{31}^* & l_{31}^* &= a_{13} / d_{11}^* \\ a_{23} &= d_{11}^* l_{21}^* l_{31}^* + d_{22}^* l_{21}^* l_{32}^* & l_{32}^* &= \frac{a_{23} - d_{11}^* l_{21}^* l_{31}^*}{d_{22}^* l_{21}^*} \\ a_{33} &= d_{11}^* l_{31}^{*2} + d_{22}^* l_{32}^{*2} + d_{33}^* & d_{33}^* &= a_{33} - d_{11}^* l_{31}^{*2} - d_{22}^* l_{32}^{*2} \end{aligned}$$

which can be summarised by:

$$\begin{cases} l_{ik}^* = \frac{a_{ki} - \sum_{j=1}^{k-1} d_{ii}^* l_{kj}^* l_{ij}^*}{d_{ii}^*}, & k < i \\ d_{ii}^* = a_{ii} - \sum_{j=1}^{i-1} d_{ii}^* l_{ij}^{*2} \end{cases}$$

### Use in multiple regression

We have a symmetric definite positive matrix  $A = X^T X$ .

Suppose factorisation  $X^T X = LU$ .

The normal equations are written as  $L(U\hat{\beta}) = X^T Y$  which is a triangular system in  $U\hat{\beta}$ , which can be solved immediately by recurrence:  $U\hat{\beta} = L^{-1} X^T Y$ .

The system in  $\hat{\beta}$  is also in triangular form

*Remark.* The algorithm is numerically less stable than modified Gram-Schmidt or Householder because we need  $X^T X$ .

### *UTILISATION OF MATLAB*

To solve the system of equations  $A*x = b$ , we write

$$x = A \setminus b$$

If matrix  $A$  is  $n \times k$ , avec  $n > k$ , the system is impossible and MATLAB computes the least squares solution

To obtain the usual statistics ( $R^2$ , standard errors, etc.), it is better to perform a QR factorisation (using a Householder transformation) as follows (excerpt from REGRESS.M, in the Statistics Toolbox of MATLAB):

[Regress.htm](#)

Functions of MATLAB which correspond to algorithms considered in this section.

|                    |  |
|--------------------|--|
| rank(X)            | rank of X  |
| norm(X) <i>or</i>  |  |
| norm(X, 2)         | largest singular value of X  |
| norm(X, 1)         | largest sum of columns of X  |
| norm(X, inf)       | largest sum of rows of X   |
| norm(X, 'fro')     | Fröbenius norm   |
| [U, S, V] = svd(X) | singular value decomposition of X, where U and V are orthogonal matrices and S is diagonal, such that $X = U S V^T$          |
| R = chol(A)        | Cholesky factorisation of matrix A (hermitian or symmetric) positive definite; R is upper triangular such that $A = R^T R$ , |
| [R, p] = chol(A)   | factorisation such that $A(1:p-1, 1:p-1) = R^T R$  |
| Y = pinv(X)        | (g4-) pseudo-inverse or Moore-Penrose inverse  |
| L = eig(X)         | vector of eigenvectors of X  |
| [V, L] = eig(X)    | matrix V of eigenvectors of X and vector L of eigenvalues  |
|                    | Remark. In both cases eig(X, Y) gives the solutions of $X v = Y v L$ (where L is a matrix)                                   |
| null(X)            | null space of columns of X   |
| orth(X)            |  |

## References

- J. M. CHAMBERS, «Computational Methods for Data Analysis», Wiley, New York, 1977.
- J. E. GENTLE, «Numerical Linear Algebra for Applications in Statistics», Springer-Verlag, 1998.
- G. H. GOLUB and C. F. VAN LOAN, «Matrix Computations», John Hopkins University Press, Baltimore, 1983.
- P. GRIFFITHS and I. D. HILL (editors), «Applied Statistics Algorithms», Ellis Horwood, Chichester, 1985.
- N. J. HIGHAM, «Accuracy and stability of numerical algorithms», Society for Industrial and Applied Mathematics, Philadelphia, 1996.
- M. J. R. HEALY, Algorithm AS6, Triangular decomposition of a symmetric matrix, J. Roy. Statist. Soc. Ser. C Applied Statistics, 1968a. Reprinted in GRIFFITH and HILL, pp. 43-45.
- M. J. R. HEALY, Algorithm AS7, Inversion of a positive semi-definite symmetric matrix, J. Roy. Statist. Soc. Ser. C Applied Statistics, 1968b. Reprinted in GRIFFITH and HILL, pp. 46-48.
- W. J. KENNEDY, Jr. and J. E. GENTLE, «Statistical Computing», Marcel Dekker, New York, 1980.
- G. MELARD et J.-M. PASTEELS, Manuel d'utilisateur de Time Series Expert (TSE version 2.2), Institut de Statistique, Université Libre de Bruxelles, Bruxelles.
- W. H. PRESS, B. P. FLANNERY, S. A. TEUKOLSKY, and W. T. VETTERLING, «Numerical Recipes in FORTRAN 77: The Art of Scientific Computing», Cambridge University Press, Cambridge, 1986.
- R. A. THISTED, «Elements of Statistical Computing: Numerical Computation», Chapman and Hall, New York, 1988.