Matrix decompositions for regression analysis

Douglas Bates

2010-09-07 Tue

Contents

1 Matrix decompositions

1.1 Orthogonal matrices

An *orthogonal* $n \times n$ matrix, $\boldsymbol{Q}$ has the property that its transpose is its inverse. That is,

$$\boldsymbol{Q}'\boldsymbol{Q} = \boldsymbol{Q}\boldsymbol{Q}' = \boldsymbol{I}_n$$

That is, the columns of $\boldsymbol{Q}$ must be orthogonal to each other and all have unit length, and the same for the rows.

Two consequences of this property are that the transformation from $\mathbb{R}^n$ to $\mathbb{R}^n$ determined by $\boldsymbol{Q}$ or $\boldsymbol{Q}'$ preserves lengths and that the determinant of $\boldsymbol{Q}$ is ± 1.

1.1.1 Preserving lengths

For any $\boldsymbol{x} \in \mathbb{R}^n$

$$\|\boldsymbol{Q}\boldsymbol{x}\|^2 = (\boldsymbol{Q}\boldsymbol{x})'\boldsymbol{Q}\boldsymbol{x} = \boldsymbol{x}'\boldsymbol{Q}'\boldsymbol{Q}\boldsymbol{x} = \boldsymbol{x}'\boldsymbol{x} = \|\boldsymbol{x}\|^2$$

Thus the transformation determined by $\boldsymbol{Q}$ or by $\boldsymbol{Q}'$ must be a *rigid* transformation, composed of reflections or rotations.

1.1.2 The determinant of an orthogonal matrix

The determinants of $n \times n$ matrices $\boldsymbol{A}$ and $\boldsymbol{B}$, written $|\boldsymbol{A}|$ and $|\boldsymbol{B}|$, are scalars with the property that

$$|\boldsymbol{A}\boldsymbol{B}| = |\boldsymbol{A}||\boldsymbol{B}|$$

Furthermore, the determinant of a *diagonal* matrix or a *triangular* matrix is simply the product of its diagonal elements. Also, $|\boldsymbol{A}| = |\boldsymbol{A}'|$.

From these properties we can derive

$$1 = |\boldsymbol{I}_n| = |\boldsymbol{Q}\boldsymbol{Q}'| = |\boldsymbol{Q}||\boldsymbol{Q}'| = |\boldsymbol{Q}|^2 \quad \Rightarrow \quad |\boldsymbol{Q}| = \pm 1.$$

As described on the Wikipedia page, the determinant of $\boldsymbol{Q}$ is the volume of the parallelepiped spanned by the columns (or by the rows) of $\boldsymbol{Q}$. We know that the columns of $\boldsymbol{Q}$ are *orthonormal* hence they span a unit volume. The sign indicates whether the transformation preserves orientation. In two dimensions a rotation preserves orientation and a reflection reverses the orientation.

1.2 The QR decomposition

An $n \times p$ matrix $\boldsymbol{X}$ has a QR decomposition consisting of an orthogonal $n \times n$ matrix $\boldsymbol{Q}$ and an $n \times p$ matrix $\boldsymbol{R}$ that is zero below the main diagonal. In the cases we consider in regression analysis the matrix $\boldsymbol{X}$ is the model matrix where n is the number of observations, p is the number of coefficients (or *parameters*) in the model and $n \geq p$. Our basic model is that the observed responses $\boldsymbol{y}$ are the realization of a vector-valued random variable $\mathcal{Y}$ with distribution

$$\mathcal{Y} \sim \mathcal{N}(\boldsymbol{X}\boldsymbol{\beta}_0, \sigma^2\boldsymbol{I}_n)$$

for some unknown value $\boldsymbol{\beta}_0$ of the coefficient vector $\boldsymbol{\beta}$.

The QR decomposition of the model matrix $\boldsymbol{X}$ is written

$$\boldsymbol{X} = \boldsymbol{Q}\begin{bmatrix}\boldsymbol{R}\\ \boldsymbol{0}\end{bmatrix} = \begin{bmatrix}\boldsymbol{Q}_1 & \boldsymbol{Q}_2\end{bmatrix}\begin{bmatrix}\boldsymbol{R}\\ \boldsymbol{0}\end{bmatrix} = \boldsymbol{Q}_1\boldsymbol{R} \tag{1}$$

where $\boldsymbol{R}$ is a $p \times p$ upper triangular matrix, $\boldsymbol{Q}_1$ is the first p columns of $\boldsymbol{Q}$ and $\boldsymbol{Q}_2$ is the last $n - p$ columns of $\boldsymbol{Q}$.

That fact that matrices $\boldsymbol{Q}$ and $\boldsymbol{R}$ must exist is proved by construction. The matrix $\boldsymbol{Q}$ is the product of p *Householder reflections* (see the Wikipedia page for the QR decomposition). The process is similar to the Gram-Schmidt orthogonalization process, but more flexible and numerically stable. If the diagonal elements of $\boldsymbol{R}$ are all non-zero (in practice this means that none of them is very small in absolute value) then $\boldsymbol{X}$ has *full column rank* and the columns of $\boldsymbol{Q}_1$ form an *orthonormal basis* of the *column span* of $\boldsymbol{X}$.

The implementation of the QR decomposition in `R` guarantees that any near-zero elements on the diagonal of $\boldsymbol{R}$ are rearranged by column permutation to occur in the trailing columns. That is, if the rank of $\boldsymbol{X}$ is $k < p$ then the first k columns of $\boldsymbol{Q}$ form an orthogonal basis for the column span of $\boldsymbol{XP}$ where $\boldsymbol{P}$ is a $p \times p$ permutation matrix.

Our text often mentions rank-deficient cases where $\text{rank}(\boldsymbol{X}) = k < p$. In practice these occur rarely because the process of building the model matrix in `R` involves a considerable amount of analysis of the model formula to remove the most common cases of rank deficiency. Nevertheless, rank deficiency can occur and is detected and handled in the `lm` function in `R`.

2 Least squares estimation

The reason that we are interested in a QR decomposition of $\boldsymbol{X}$ is because the probability model is linked to geometric properties of the response $\boldsymbol{y}$ and the column span of $\boldsymbol{X}$.

2.1 Spherical normal distributions and least squares estimators

A distribution of the form

$$\mathcal{Y} \sim \mathcal{N}(\boldsymbol{X}\boldsymbol{\beta}_0, \sigma^2\boldsymbol{I}_n) \tag{2}$$

is called a *spherical normal* distribution because the contours of constant probability density are spheres centered at $\boldsymbol{X}\boldsymbol{\beta}_0$. In other words, the probability density function of $\mathcal{Y}$, evaluated at $\boldsymbol{y}$ is determined by the distance, $\|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}_0\|^2$ with larger distances corresponding to lower densities.

The *maximum likelihood estimate*, $\widehat{\boldsymbol{\beta}}$ of the coefficient vector, $\boldsymbol{\beta}$, is

$$\widehat{\boldsymbol{\beta}} = \arg\min_{\beta} \|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}\|^2 \tag{3}$$

These estimates are also called the *least squares estimates* of $\boldsymbol{\beta}$.

Because multiplication by an orthogonal matrix like $\boldsymbol{Q}'$ preserves lengths we can write

$$\widehat{\boldsymbol{\beta}} = \arg\min_{\beta} \|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}\|^2 = \arg\min_{\beta} \|\boldsymbol{Q}'\boldsymbol{y} - \boldsymbol{Q}'\boldsymbol{X}\boldsymbol{\beta}\|^2 = \arg\min_{\beta} \|\boldsymbol{c}_1 - \boldsymbol{R}\boldsymbol{\beta}\|^2 + \|\boldsymbol{c}_2\|^2 \tag{4}$$

where $\boldsymbol{c}_1 = \boldsymbol{Q}_1'\boldsymbol{y}$ is the first p elements of $\boldsymbol{Q}\boldsymbol{y}$ and $\boldsymbol{c}_2 = \boldsymbol{Q}_2'\boldsymbol{y}$ is the last $n - p$ elements. If $\text{rank}(\boldsymbol{X}) = p$ then $\text{rank}(\boldsymbol{R}) = p$ and $\boldsymbol{R}^{-1}$ exists and we can write $\widehat{\boldsymbol{\beta}} = \boldsymbol{R}^{-1}\boldsymbol{c}_1$ (although you don't actually calculate $\boldsymbol{R}^{-1}$ to solve the triangular linear system $\boldsymbol{R}\widehat{\boldsymbol{\beta}} = \boldsymbol{c}_1$ for $\widehat{\boldsymbol{\beta}}$).

In a model fit by the `lm` or `aov` functions in `R` there is a component `e$ffects` which is $\boldsymbol{Q}'\boldsymbol{y}$. The component `$qr` is a condensed form of the QR decomposition of the model matrix $\boldsymbol{X}$. The matrix $\boldsymbol{R}$ is embedded in there but the matrix $\boldsymbol{Q}$ is a virtual matrix represented as a product of Householder reflections and not usually evaluated explicitly.

2.2 Comparison to the usual text-book formulas

Most text books state that the least squares estimates are

$$\widehat{\boldsymbol{\beta}} = (\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'\boldsymbol{y} \tag{5}$$

giving the impression that $\widehat{\boldsymbol{\beta}}$ is calculated this way. It isn't.

If you substitute $\boldsymbol{X} = \boldsymbol{Q}_1\boldsymbol{R}$ in 5 you get

$$(\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'\boldsymbol{y} = (\boldsymbol{R}'\boldsymbol{R})^{-1}\boldsymbol{R}'\boldsymbol{Q}_1'\boldsymbol{y} = \boldsymbol{R}^{-1}(\boldsymbol{R}')^{-1}\boldsymbol{R}'\boldsymbol{Q}_1'\boldsymbol{y} = \boldsymbol{R}^{-1}\boldsymbol{Q}_1'\boldsymbol{y},$$

our previous result.

Whenever you see $\boldsymbol{X}'\boldsymbol{X}$ in a formula you should mentally replace it by $\boldsymbol{R}'\boldsymbol{R}$ and similarly replace $(\boldsymbol{X}'\boldsymbol{X})^{-1}$ by $\boldsymbol{R}^{-1}(\boldsymbol{R}')^{-1}$ then see if you can simplify the result.

For example, the variance of the least squares estimator $\widehat{\boldsymbol{\beta}}$ is

$$(\widehat{\boldsymbol{\beta}}) = \sigma^2(\boldsymbol{X}'\boldsymbol{X}) = \sigma^2\boldsymbol{R}^{-1}(\boldsymbol{R}^{-1})'$$

The `R` function `chol2inv` calculates $\boldsymbol{R}^{-1}(\boldsymbol{R}^{-1})'$ directly from $\boldsymbol{R}$ (not a big deal in most cases but when p is very large it should be faster and more accurate than evaluating $\boldsymbol{R}^{-1}$ explicitly). The determinant of $\boldsymbol{X}'\boldsymbol{X}$ is

$$|\boldsymbol{X}'\boldsymbol{X}| = |\boldsymbol{R}'\boldsymbol{R}| = |\boldsymbol{R}|^2 = \left(\prod_{i=1}^{p} r_{i,i}\right)^2$$

The fitted values $\widehat{\boldsymbol{y}}$ are $\boldsymbol{Q}_1\boldsymbol{Q}_1'\boldsymbol{y}$ and thus the *hat matrix* (which puts a "hat" on $\boldsymbol{y}$ by transforming it to $\widehat{\boldsymbol{y}}$) is the $n \times n$ matrix $\boldsymbol{Q}_1\boldsymbol{Q}_1'$. Often we are interested in the diagonal elements of the hat matrix, which are the sums of the squares of rows of $\boldsymbol{Q}_1$. (In practice you don't want to calculate the entire $n \times n$ hat matrix just to get the diagonal elements when n could be very large.)

The residuals, $\widehat{\boldsymbol{e}} = \boldsymbol{y} - \widehat{\boldsymbol{y}}$, are calculated as $\widehat{\boldsymbol{e}} = \boldsymbol{Q}_2\boldsymbol{Q}_2'\boldsymbol{y}$.

The matrices $\boldsymbol{Q}_1\boldsymbol{Q}_1'$ and $\boldsymbol{Q}_2\boldsymbol{Q}_2'$ are *projection matrices*, which means that they are symmetric and *idempotent*. (A square matrix $\boldsymbol{A}$ is idempotent if $\boldsymbol{A}\boldsymbol{A} = \boldsymbol{A}$.) When $\text{rank}(\boldsymbol{X}) = p$, the hat matrix $\boldsymbol{Q}_1\boldsymbol{Q}_1'$ projects any vector in $\mathbb{R}^n$ onto the column span of $\boldsymbol{X}$. The other projection, $\boldsymbol{Q}_2\boldsymbol{Q}_2'$, is onto the subspace orthogonal to the column span of $\boldsymbol{X}$ (see the figure on the front cover of the text).

2.3 R functions related to the QR decomposition

As mentioned above, every time you fit a linear model with `lm` or `aov` or `lm.fit`, the returned object contains a `$qr` component. This is a condensed form of the decomposition, only slightly larger than $\boldsymbol{X}$ itself. Its class is `"qr"`.

There are several extractor functions for a `"qr"` object: `qr.R()`. `qr.Q()` and `qr.X()`, which regenerates the original matrix. By default `qr.Q()` returns the matrix called $\boldsymbol{Q}_1$ above with p

columns but you can specify the number of columns desired. Typical alternative choices are n or rank($\boldsymbol{X}$).

The `$rank` component of a `"qr"` object is the computed rank of $\boldsymbol{X}$ (and, hence, of $\boldsymbol{R}$). The `$pivot` component is the permutation applied to the columns. It will be `1:p` when rank($\boldsymbol{X}$) $= p$ but when rank($\boldsymbol{X}$) $< p$ it may be other than the identity permutation.

Several functions are applied to a `"qr"` object and a vector or matrix. These include `qr.coef()`, `qr.qy()`, `qr.qty()`, `qr.resid()` and `qr.fitted()`. The `qr.qy()` and `qr.qty()` functions multiply an n-vector or an $n \times m$ matrix by $\boldsymbol{Q}$ or $\boldsymbol{Q}'$ without ever forming $\boldsymbol{Q}$. Similarly, `qr.fitted()` creates $\boldsymbol{Q}_1\boldsymbol{Q}_1'\boldsymbol{x}$ and `qr.resid()` creates $\boldsymbol{Q}_2\boldsymbol{Q}_2'\boldsymbol{x}$ without forming $\boldsymbol{Q}$.

The `is.qr()` function tests an object to determine if it is of class `"qr"`.

3 Related matrix decompositions

3.1 The Cholesky decomposition

The Cholesky decomposition of a positive definite symmetric matrix, which means a $p \times p$ symmetric matrix $\boldsymbol{A}$ such that $\boldsymbol{x}'\boldsymbol{A}\boldsymbol{x} > 0\$forallnon - zero\boldsymbol{x} \in \mathbb{R}^p$ is of the form

$$\boldsymbol{A} = \boldsymbol{R}'\boldsymbol{R} = \boldsymbol{L}\boldsymbol{L}'$$

where $\boldsymbol{R}$ is an upper triangular $p \times p$ matrix and $\boldsymbol{L} = \boldsymbol{R}'$ is lower triangular. The two forms are the same decomposition: it is just a matter of whether you want $\boldsymbol{R}$, the factor on the right, or $\boldsymbol{L}$, the factor on the left. Generally statisticians write the decomposition as $\boldsymbol{R}'\boldsymbol{R}$.

The decomposition is only determined up to changes in sign of the rows of $\boldsymbol{R}$ (or, equivalently, the columns of $\boldsymbol{L}$). For definiteness we require positive diagonal elements in $\boldsymbol{R}$.

When rank($\boldsymbol{X}$) $= p$ the Cholesky decomposition $\boldsymbol{R}$ of $\boldsymbol{X}'\boldsymbol{X}$ is the equal to the matrix $\boldsymbol{R}$ from the QR decomposition up to changes in sign of rows. The matrix $\boldsymbol{X}'\boldsymbol{X}$ matrix is obviously symmetric and it is positive definite because

$$\boldsymbol{x}'(\boldsymbol{X}'\boldsymbol{X})\boldsymbol{x} = \boldsymbol{x}'(\boldsymbol{R}'\boldsymbol{R})\boldsymbol{x} = \|\boldsymbol{R}\boldsymbol{x}\|^2 \geq 0$$

with equality only when $\boldsymbol{R}\boldsymbol{x} = \boldsymbol{0}$, which, when rank($\boldsymbol{R}$) $= p$, implies that $\boldsymbol{x} = \boldsymbol{0}$.

3.2 Evaluation of the Cholesky decomposition

The `R` function `chol()` evaluates the Cholesky decomposition. As mentioned above `chol2inv()` creates $(\boldsymbol{X}'\boldsymbol{X})^{-1}$ directly from the Cholesky decomposition of $\boldsymbol{X}'\boldsymbol{X}$.

Generally the QR decomposition is preferred to the Cholesky decomposition for least squares problems because there is a certain loss of precision when forming $\boldsymbol{X}'\boldsymbol{X}$. However, when n is very large you may want to build up $\boldsymbol{X}'\boldsymbol{X}$ using blocks of rows. Also, if $\boldsymbol{X}$ is *sparse* it is an advantage to use sparse matrix techniques to evaluate and store the Cholesky decomposition.

The `Matrix` package for `R` provides even more capabilities related to the Cholesky decomposition, especially for sparse matrices.

For everything we will do in Statistics 849 the QR decomposition should be the method of choice.

3.3 The singular value decomposition

Another decomposition related to orthogonal matrices is the singular value decomposition (or SVD) in which the matrix $\boldsymbol{X}$ is reduced to a diagonal form

$$\boldsymbol{X} = \boldsymbol{U}_1 \boldsymbol{D} \boldsymbol{V}' = \boldsymbol{U} \begin{bmatrix} \boldsymbol{D} \\ \boldsymbol{0} \end{bmatrix} \boldsymbol{V}'$$

where $\boldsymbol{U}$ is an $n \times n$ orthogonal matrix, $\boldsymbol{D}$ is a $p \times p$ diagonal matrix with non-negative diagonal elements (which are called the *singular values* of $\boldsymbol{X}$) and $\boldsymbol{V}$ is a $p \times p$ orthogonal matrix. As for $\boldsymbol{Q}$ and $\boldsymbol{Q}_1$, $\boldsymbol{U}_1$ consists of the first p columns of $\boldsymbol{U}$. For definiteness we order the diagonal elements of $\boldsymbol{D}$, which must be non-negative, in decreasing order.

The singular value decomposition of $\boldsymbol{X}$ is related to the eigendecomposition or spectral decomposition of $\boldsymbol{X}'\boldsymbol{X}$ because

$$\boldsymbol{X}'\boldsymbol{X} = \boldsymbol{V}\boldsymbol{D}\boldsymbol{U}_1'\boldsymbol{U}_1\boldsymbol{D}\boldsymbol{V}' = \boldsymbol{V}\boldsymbol{D}^2\boldsymbol{V}'$$

implying that the eigenvalues of $\boldsymbol{X}'\boldsymbol{X}$ are the squares of the singular values of $\boldsymbol{X}$ and the right singular vectors, which are the columns of $\boldsymbol{V}$ are also the eigenvectors of $\boldsymbol{X}'\boldsymbol{X}$

Calculation of the SVD is an iterative (as opposed to a direct) computation and potentially more computing intensive than the QR decomposition, although modern methods for evaluating the SVD are very good indeed.

Symbolically we can write the least squares solution in the full-rank case as

$$\widehat{\boldsymbol{\beta}} = \boldsymbol{V}\boldsymbol{D}^{-1}\boldsymbol{U}_1'\boldsymbol{y}$$

where $\boldsymbol{D}^{-1}$ is a diagonal matrix whose diagonal elements are the inverses of the diagonal elements of $\boldsymbol{D}$.

The pseudo-inverse or *generalized inverse* of $\boldsymbol{X}$, written $\boldsymbol{X}^-$ is calculated from the pseudo-inverse of the diagonal matrix, $\boldsymbol{D}$. In theory the diagonal elements of $\boldsymbol{D}^-$ are $1/di, i$ when $d_{i,i} \neq 0$ and 0 when $d_{i,i} = 0$. However, we can't count on $d_{i,i}$ being 0 even when, in theory, it should be. You need to decide when the singular values are close to zero, which is actually a very difficult problem. At best we can some heuristics, based on the ratio of $d_{i,i}/d_{1,1}$ to decide when a diagonal element is "effectively zero".

The use of the pseudo-inverse seems to be a convenient way to handle rank-deficient $\boldsymbol{X}$ matrices but, as mentioned above, the best way to handle rank-deficient $\boldsymbol{X}$ matrices is not to produce them in the first place.