IECD-325: Aspectos numéricos de estimación LS en regresión lineal

Felipe Osorio

felipe.osorio@uv.cl

Considere el modelo de regresión lineal:

$$\boldsymbol{Y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{\epsilon},$$

donde $\boldsymbol{X} \in \mathbb{R}^{n\times p}$ con $\operatorname{rg}(\boldsymbol{X}) = p$ y $\mathsf{E}(\boldsymbol{\epsilon}) = \boldsymbol{0}$ y $\mathsf{Cov}(\boldsymbol{\epsilon}) = \sigma^2\boldsymbol{I}$. El estimador mínimos cuadrados (LS) de $\boldsymbol{\beta}$ es:

$$\widehat{\boldsymbol{\beta}} = (\boldsymbol{X}^\top\boldsymbol{X})^{-1}\boldsymbol{X}^\top\boldsymbol{Y}, \qquad \text{con} \qquad \mathsf{Cov}(\widehat{\boldsymbol{\beta}}) = \sigma^2(\boldsymbol{X}^\top\boldsymbol{X})^{-1}.$$

Además,

$$s^2 = \frac{1}{n-p}\|\boldsymbol{Y} - \boldsymbol{X}\widehat{\boldsymbol{\beta}}\|^2.$$

Adicionalmente, si $\boldsymbol{\epsilon} \sim \mathsf{N}_n(\boldsymbol{0}, \sigma^2\boldsymbol{I})$, entonces[1]

$$\widehat{\boldsymbol{\beta}} \sim \mathsf{N}_p(\boldsymbol{\beta}, \sigma^2(\boldsymbol{X}^\top\boldsymbol{X})^{-1}),$$
$$\frac{(n-p)s^2}{\sigma^2} \sim \chi^2(n-p).$$

[1] En cuyo caso, $\widehat{\boldsymbol{\beta}}$ corresponde al estimador ML de $\boldsymbol{\beta}$.

Aspectos numéricos de estimación LS en regresión lineal

El procedimiento puede ser expresado como la solución del problema:

$$\min_{\beta \in \mathbb{R}^p} Q(\boldsymbol{\beta}), \qquad \text{con} \qquad Q(\boldsymbol{\beta}) = \|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}\|_2^2,$$

lo que lleva a las ecuaciones de estimación $\boldsymbol{X}^\top(\boldsymbol{Y} - \boldsymbol{X}\widehat{\boldsymbol{\beta}}) = \boldsymbol{0}$, o bien

$$\boldsymbol{X}^\top \boldsymbol{X}\widehat{\boldsymbol{\beta}} = \boldsymbol{X}^\top \boldsymbol{Y}.$$

Métodos habituales para obtener $\widehat{\boldsymbol{\beta}}$, son:

- Descomposición Cholesky y operador Sweep.[2]
- Descomposiciones QR[3] y SVD.
- Menos común, en regresión, es el uso del método gradientes conjugados (CG).[4]

[2] Goodnight (1979). The American Statistician **33**, 149-158.
[3] Único método disponible en función `lm` de R.
[4] McIntosh (1982). Lecture Notes in Statistics 10. Springer, New York.

Observaciones:

- Cholesky[5] y Sweep requieren formar las matrices:

$$\boldsymbol{X}^\top\boldsymbol{X}, \boldsymbol{X}^\top\boldsymbol{y}, \qquad \text{y} \qquad \begin{pmatrix} \boldsymbol{X}^\top\boldsymbol{X} & \boldsymbol{X}^\top\boldsymbol{Y} \\ \boldsymbol{Y}^\top\boldsymbol{X} & \boldsymbol{Y}^\top\boldsymbol{Y} \end{pmatrix},$$

  respectivamente.

- QR[6] y SVD descomponen la matriz de diseño $\boldsymbol{X}$ y resuelven sistemas lineales (triangular y diagonal, respectivamente) mucho más pequeños ($n >> p$).

- Note que $\kappa(\boldsymbol{X}^\top\boldsymbol{X}) = \kappa^2(\boldsymbol{X})$.

- Una implementación cuidadosa de CG sólo requiere productos matriz-vector/ operaciones entre vectores.[7]

- Existe código confiable y con excelente desempeño para álgebra lineal numérica en las bibliotecas BLAS, LAPACK, rutinas que pueden ser invocadas desde (por ejemplo) R y Matlab.

[5] $np^2/2 + p^3/6$ flops, $p(p+3)/2$ almacenamiento
[6] $np^2 + p^3/3$ flops, $np + p$ almacenamiento
[7] $O(p)$ flops, $4p$ almacenamiento

Resultado 1 (Factorización Cholesky):

Sea $\boldsymbol{A} \in \mathbb{R}^{p \times p}$ es matriz simétrica y definida positiva, entonces existe una única matriz triangular superior $\boldsymbol{G} \in \mathbb{R}^{p \times p}$ con elementos diagonales positivos tal que

$$\boldsymbol{A} = \boldsymbol{G}^\top \boldsymbol{G}$$

Observación:

Note que si usamos la factorización Cholesky para resolver el sistema $\boldsymbol{Ax} = \boldsymbol{b}$. Entonces debemos resolver los sistemas triangulares

$$\boldsymbol{G}^\top \boldsymbol{z} = \boldsymbol{b}, \qquad \text{y} \qquad \boldsymbol{Gx} = \boldsymbol{z}.$$

En efecto,

$$\boldsymbol{Ax} = (\boldsymbol{G}^\top \boldsymbol{G})\boldsymbol{x} = \boldsymbol{G}^\top (\boldsymbol{Gx}) = \boldsymbol{G}^\top \boldsymbol{z} = \boldsymbol{b}.$$

Algoritmo 1: Factorización Cholesky

Entrada: Matriz $\boldsymbol{A} \in \mathbb{R}^{p \times p}$.
Salida : Factor Cholesky $\boldsymbol{G} \in \mathbb{R}^{p \times p}$.

begin
    $g_{11} = \sqrt{a_{11}}$.
    for $j = 2$ **to** p **do**
        $g_{1j} = a_{1j}/g_{11}$.
    end
    for $i = 2$ **to** p **do**
        $g_{ii} = \sqrt{a_{ii} - \sum_{k=1}^{i-1} g_{ki}^2}$,
        for $j = i + 1$ **to** n **do**
            $g_{ij} = \big(a_{ij} - \sum_{k=1}^{i-1} g_{ki}g_{kj}\big)/g_{ii}$
        end
    end
end

Las ecuaciones de estimación para obtener $\widehat{\boldsymbol{\beta}}$ son $\boldsymbol{X}^\top(\boldsymbol{Y} - \boldsymbol{X}\widehat{\boldsymbol{\beta}}) = \boldsymbol{0}$, o bien

$$\boldsymbol{X}^\top \boldsymbol{X}\widehat{\boldsymbol{\beta}} = \boldsymbol{X}^\top \boldsymbol{Y}. \tag{1}$$

Sea $RSS = Q(\widehat{\boldsymbol{\beta}})$ y note que

$$RSS = \|\boldsymbol{Y} - \boldsymbol{X}\widehat{\boldsymbol{\beta}}\|^2 = \boldsymbol{Y}^\top \boldsymbol{Y} - \boldsymbol{Y}^\top \boldsymbol{X}\widehat{\boldsymbol{\beta}} - \widehat{\boldsymbol{\beta}}^\top \boldsymbol{X}^\top \boldsymbol{Y} + \widehat{\boldsymbol{\beta}}^\top \boldsymbol{X}^\top \boldsymbol{X}\widehat{\boldsymbol{\beta}},$$

como

$$\begin{aligned} \boldsymbol{Y}^\top \boldsymbol{X}\widehat{\boldsymbol{\beta}} &= \boldsymbol{Y}^\top \boldsymbol{X}(\boldsymbol{X}^\top \boldsymbol{X})^{-1}\boldsymbol{X}^\top \boldsymbol{Y} = \boldsymbol{Y}^\top \boldsymbol{H}^2 \boldsymbol{Y} = \widehat{\boldsymbol{\beta}}^\top \widehat{\boldsymbol{Y}} \\ &= \widehat{\boldsymbol{\beta}}^\top \boldsymbol{X}^\top \boldsymbol{X}\widehat{\boldsymbol{\beta}}. \end{aligned}$$

Es decir, podemos escribir:

$$RSS = \|\boldsymbol{Y} - \boldsymbol{X}\widehat{\boldsymbol{\beta}}\|^2 = \boldsymbol{Y}^\top \boldsymbol{Y} - \widehat{\boldsymbol{\beta}}^\top \boldsymbol{X}^\top \boldsymbol{X}\widehat{\boldsymbol{\beta}}.$$

Podemos resolver (1) usando la descomposición Cholesky, de

$$\boldsymbol{X}^\top \boldsymbol{X} = \boldsymbol{U}^\top \boldsymbol{U}, \qquad \text{(DPOTRF)}$$

con $\boldsymbol{U}$ matrix triangular superior. De este modo, debemos resolver los sistemas triangulares:

$$\boldsymbol{U}^\top \boldsymbol{z} = \boldsymbol{X}^\top \boldsymbol{Y}, \qquad \text{y} \qquad \boldsymbol{U}\widehat{\boldsymbol{\beta}} = \boldsymbol{z}, \qquad \text{(DTRTRS)}$$

para obtener s^2 considere

$$RSS = \|\boldsymbol{Y} - \boldsymbol{X}\widehat{\boldsymbol{\beta}}\|^2 = \boldsymbol{Y}^\top \boldsymbol{Y} - \boldsymbol{z}^\top \boldsymbol{z}. \qquad \text{(DDOT)}$$

Invirtiendo $\boldsymbol{U}$ (in-place)[8], podemos hacer

$$\boldsymbol{U}^{-1}\boldsymbol{U}^{-\top} = (\boldsymbol{X}^\top \boldsymbol{X})^{-1}.$$

[8] Haciendo $\boldsymbol{U} \leftarrow \boldsymbol{U}^{-1}$ (DTRTRI), tenemos $(\boldsymbol{X}^\top \boldsymbol{X})^{-1} = \boldsymbol{U}\boldsymbol{U}^\top$ (DGEMM)

Definición 1 (Operador Sweep):

Sea $\boldsymbol{A} = (a_{ij})$ matriz cuadrada $p \times p$, aplicando el operador Sweep[9] sobre el k-ésimo elemento diagonal de $\boldsymbol{A}$ $(a_{kk} \neq 0)$ permite obtener la matriz $\boldsymbol{B}$, definida como:

$$\begin{aligned} b_{kk} &= \frac{1}{a_{kk}}, & & \\ b_{ik} &= -\frac{a_{ik}}{a_{kk}}, & & i \neq k, \\ b_{kj} &= \frac{a_{kj}}{a_{kk}}, & & j \neq k, \\ b_{ij} &= a_{ij} - \frac{a_{ik} a_{kj}}{a_{kk}}, & & i, j \neq k, \end{aligned}$$

y escribimos $\boldsymbol{B} = \text{Sweep}(k)\boldsymbol{A}$.

[9] Disponible en la función `sweep.operator` de la biblioteca `fastmatrix`.

Operador Sweep

Propiedades:

- $\text{Sweep}(k)\,\text{Sweep}(k)\boldsymbol{A} = \boldsymbol{A}$.
- $\text{Sweep}(k)\,\text{Sweep}(r)\boldsymbol{A} = \text{Sweep}(r)\,\text{Sweep}(k)\boldsymbol{A}$.
- $\boldsymbol{A}^{-1} = \prod_{i=1}^{n} \text{Sweep}(i)\boldsymbol{A}$.

Observaciones:

- Si $\boldsymbol{A}$ es matriz simétrica, el operador Sweep preserva la simetría de $\boldsymbol{A}$.
- Existen varias definiciones ligeramente diferentes del operador Sweep.
- Problemas de inestabilidad pueden ocurrir cuando algún a_{kk} es cercano a cero.

Operador Sweep

Considere $\boldsymbol{A} \in \mathbb{R}^{p \times p}$ matriz particionada como:

$$\boldsymbol{A} = \begin{pmatrix} \boldsymbol{A}_{11} & \boldsymbol{A}_{12} \\ \boldsymbol{A}_{21} & \boldsymbol{A}_{22} \end{pmatrix},$$

donde $\boldsymbol{A}_{11} \in \mathbb{R}^{r \times r}$ $(r < p)$. Suponga que se aplica el operador Sweep sobre los elementos diagonales de $\boldsymbol{A}_{11}$. De este modo,

$$\boldsymbol{B} = \prod_{i=1}^{r} \text{Sweep}(i)\boldsymbol{A} = \begin{pmatrix} \boldsymbol{B}_{11} & \boldsymbol{B}_{12} \\ \boldsymbol{B}_{21} & \boldsymbol{B}_{22} \end{pmatrix},$$

con

$$\begin{aligned} \boldsymbol{B}_{11} &= \boldsymbol{A}_{11}^{-1}, & \boldsymbol{B}_{12} &= \boldsymbol{A}_{11}^{-1}\boldsymbol{A}_{12}, \\ \boldsymbol{B}_{21} &= -\boldsymbol{A}_{21}\boldsymbol{A}_{11}^{-1}, & \boldsymbol{B}_{22} &= \boldsymbol{A}_{22} - \boldsymbol{A}_{21}\boldsymbol{A}_{11}^{-1}\boldsymbol{A}_{12}. \end{aligned}$$

Operador Sweep en estimación LS

Considere

$$\boldsymbol{Z} = (\boldsymbol{X}, \boldsymbol{Y}) \in \mathbb{R}^{n\times(p+1)},$$

luego

$$\boldsymbol{Z}^\top\boldsymbol{Z} = \begin{pmatrix} \boldsymbol{X}^\top\boldsymbol{X} & \boldsymbol{X}^\top\boldsymbol{Y} \\ \boldsymbol{Y}^\top\boldsymbol{X} & \boldsymbol{Y}^\top\boldsymbol{Y} \end{pmatrix} \in \mathbb{R}^{(p+1)\times(p+1)}.$$

Aplicando el operador Sweep sobre los primeros p elementos diagonales de $\boldsymbol{Z}^\top\boldsymbol{Z}$, obtenemos:

$$\begin{aligned} \boldsymbol{B} &= \prod_{i=1}^{p} \mathsf{Sweep}(i)\boldsymbol{Z}^\top\boldsymbol{Z} \\ &= \begin{pmatrix} (\boldsymbol{X}^\top\boldsymbol{X})^{-1} & (\boldsymbol{X}^\top\boldsymbol{X})^{-1}\boldsymbol{X}^\top\boldsymbol{Y} \\ -\boldsymbol{Y}^\top\boldsymbol{X}(\boldsymbol{X}^\top\boldsymbol{X})^{-1} & \boldsymbol{Y}^\top\boldsymbol{Y} - \boldsymbol{Y}^\top\boldsymbol{X}(\boldsymbol{X}^\top\boldsymbol{X})^{-1}\boldsymbol{X}^\top\boldsymbol{Y} \end{pmatrix} \\ &= \begin{pmatrix} (\boldsymbol{X}^\top\boldsymbol{X})^{-1} & \widehat{\boldsymbol{\beta}} \\ \widehat{\boldsymbol{\beta}}^\top & RSS \end{pmatrix}. \end{aligned}$$

Definición 2 (Descomposición QR):

Sea $\boldsymbol{A} \in \mathbb{R}^{n \times p}$, entonces existe $\boldsymbol{Q} \in \mathcal{O}_n$ y $\boldsymbol{R} \in \mathbb{R}^{n \times p}$, tal que

$$\boldsymbol{A} = \boldsymbol{Q}\boldsymbol{R},$$

donde

$$\boldsymbol{R} = \begin{pmatrix} \boldsymbol{R}_1 \\ \boldsymbol{0} \end{pmatrix}$$

con $\boldsymbol{R}_1 \in \mathbb{R}^{p \times p}$ matriz triangular superior, aquí suponemos que $n \geq p$.

Observación:

Si $\boldsymbol{A} = \boldsymbol{Q}\boldsymbol{R}$ entonces

$$\boldsymbol{A}^\top \boldsymbol{A} = \boldsymbol{R}^\top \boldsymbol{Q}^\top \boldsymbol{Q}\boldsymbol{R} = \boldsymbol{R}^\top \boldsymbol{R} = \boldsymbol{R}_1^\top \boldsymbol{R}_1,$$

y $\boldsymbol{R}_1$ corresponde al factor Cholesky de $\boldsymbol{A}^\top \boldsymbol{A}$.

Algunas propiedades fundamentales de las matrices ortogonales, son las siguientes:

- $\boldsymbol{Q}\boldsymbol{Q}^\top = \boldsymbol{Q}^\top\boldsymbol{Q} = \boldsymbol{I}$.
- $\langle \boldsymbol{Q}\boldsymbol{x}, \boldsymbol{Q}\boldsymbol{y}\rangle = \boldsymbol{x}^\top\boldsymbol{Q}^\top\boldsymbol{Q}\boldsymbol{y} = \boldsymbol{x}^\top\boldsymbol{y} = \langle \boldsymbol{x}, \boldsymbol{y}\rangle$.
- $\|\boldsymbol{Q}\boldsymbol{x}\| = \|\boldsymbol{x}\|$.
- Si $\boldsymbol{B} = \boldsymbol{Q}^\top\boldsymbol{A}\boldsymbol{Q}$, entonces $\boldsymbol{A}$ y $\boldsymbol{B}$ tienen los mismos valores propios para $\boldsymbol{Q}$ matriz ortogonal.

Existen diversar variantes del algoritmo para implementar la descomposición QR. A continuación veremos una basada en transformaciones Householder.

Problema 1:

Para $\boldsymbol{x} \in \mathbb{R}^p$, $\boldsymbol{x} \neq \boldsymbol{0}$, hallar una matriz ortogonal $\boldsymbol{M} \in \mathbb{R}^{n \times n}$ tal que

$$\boldsymbol{M}^\top \boldsymbol{x} = \|\boldsymbol{x}\| \, \boldsymbol{e}_1,$$

donde $\boldsymbol{e}_1 = (1, 0, \dots, 0)^\top$ denota el primer vector unidad.

Definición 3 (Reflexión):

Sea $\boldsymbol{u}$ y $\boldsymbol{v}$ vectores ortonormales y $\boldsymbol{x}$ vector generado por $\boldsymbol{u}$ y $\boldsymbol{v}$. Entonces

$$\boldsymbol{x} = c_1 \boldsymbol{u} + c_2 \boldsymbol{v},$$

para escalares c_1, c_2. El vector

$$\widetilde{\boldsymbol{x}} = -c_1 \boldsymbol{u} + c_2 \boldsymbol{v},$$

el llamado una reflexión de $\boldsymbol{x}$ a través de la línea definida por el vector $\boldsymbol{v}$ (o $\boldsymbol{u}^\perp$).

Definición 4 (Transformación Householder):

Sea $\boldsymbol{x} = c_1\boldsymbol{u} + c_2\boldsymbol{v}$, con $\boldsymbol{u}$ y $\boldsymbol{v}$ vectores generadores de $\boldsymbol{x}$ y considere la matriz

$$\boldsymbol{H} = \boldsymbol{I} - \lambda\boldsymbol{u}\boldsymbol{u}^\top, \qquad \lambda = 2/\boldsymbol{u}^\top\boldsymbol{u}.$$

Note que $\boldsymbol{H}\boldsymbol{x} = \widetilde{\boldsymbol{x}}$, es decir $\boldsymbol{H}$ es un reflector.

La transformación Householder satisface las siguientes propiedades:

- $\boldsymbol{H}\boldsymbol{u} = -\boldsymbol{u}$.
- $\boldsymbol{H}\boldsymbol{v} = \boldsymbol{v}$ para cualquier $\boldsymbol{v}$ ortogonal a $\boldsymbol{u}$.
- $\boldsymbol{H}^\top = \boldsymbol{H}$.
- $\boldsymbol{H}^{-1} = \boldsymbol{H}^\top$.

Observación:

La operación $\boldsymbol{H}\boldsymbol{x}$ puede ser obtenida usando un axpy.[10]

[10] Actualización del tipo: $\boldsymbol{y} \leftarrow \alpha\boldsymbol{x} + \boldsymbol{y}$.

La descomposición QR de una matriz $\boldsymbol{A} \in \mathbb{R}^{n \times p}$ $(n > p)$, puede ser construída a través de una secuencia de matrices $\boldsymbol{Q}_1, \dots, \boldsymbol{Q}_p$ tales que

$$\boldsymbol{Q}_p \cdots \boldsymbol{Q}_1 \boldsymbol{A} = \begin{pmatrix} \boldsymbol{R} \\ \boldsymbol{0} \end{pmatrix},$$

donde todas $\boldsymbol{Q}_1, \dots, \boldsymbol{Q}_p$ son ortogonales. De este modo,

$$\boldsymbol{A} = \boldsymbol{Q}_1^\top \cdots \boldsymbol{Q}_p^\top \begin{pmatrix} \boldsymbol{R} \\ \boldsymbol{0} \end{pmatrix} = \boldsymbol{Q} \begin{pmatrix} \boldsymbol{R} \\ \boldsymbol{0} \end{pmatrix}.$$

A continuación se describe el algoritmo para obtener la descomposición QR usando transformaciones Householder.[11] Sea $\boldsymbol{M}(\boldsymbol{x})$ la matriz ortogonal desde el Problema 1 basada en un vector $\boldsymbol{x}$.

[11] Otro método popular para obtener la descomposición QR es usando rotaciones Givens.

Descomposición QR

Algoritmo 2: Descomposición QR

Entrada: Matriz $\boldsymbol{A} \in \mathbb{R}^{n\times p}$.
Salida : Factores $\boldsymbol{Q}$ y $\boldsymbol{R}$, matrices ortogonal y triangular superior, respectivamente.

begin
    Hacer $\boldsymbol{Q} = \boldsymbol{I}_n$ y $\boldsymbol{R} = \boldsymbol{A}$
    for $i = 1$ **to** p **do**
        $\boldsymbol{x} = (R_{1i}, \dots, R_{pi})^\top$
        $\boldsymbol{Q}_i = \begin{pmatrix} \boldsymbol{I}_{i-1} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{M}(\boldsymbol{x}) \end{pmatrix}$
        /* $\boldsymbol{M}(\boldsymbol{x})$ obtenido usando reflexiones Householder */
        $\boldsymbol{Q} = \boldsymbol{Q}_i\boldsymbol{Q}$
        $\boldsymbol{R} = \boldsymbol{Q}_i\boldsymbol{R}$
    end
    $\boldsymbol{Q} = \boldsymbol{Q}^\top$
    $\boldsymbol{R} = (R_{ij})$ para $i, j = 1, \dots, p$.
end

Descomposición QR en estimación LS

Considere la descomposición QR de $\boldsymbol{X}$, como:

$$\boldsymbol{X} = \boldsymbol{Q}\boldsymbol{R}, \qquad \boldsymbol{R} = \begin{pmatrix} \boldsymbol{R}_1 \\ \boldsymbol{0} \end{pmatrix}, \qquad \text{(DGEQRF)}$$

con $\boldsymbol{R}_1 \in \mathbb{R}^{p\times p}$ matriz triangular superior $(n > p)$. Si $\operatorname{rg}(\boldsymbol{X}) = p$, entonces $\boldsymbol{R}_1$ es no singular. Además, considere la transformación:

$$\boldsymbol{Q}^\top \boldsymbol{Y} = \boldsymbol{c}, \qquad \boldsymbol{c} = \begin{pmatrix} \boldsymbol{c}_1 \\ \boldsymbol{c}_2 \end{pmatrix}. \qquad \text{(DORMQR)}$$

El estimador LS minimiza la función objetivo:

$$\begin{aligned} \|\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{\beta}\|^2 &= \|\boldsymbol{Q}^\top(\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{\beta})\|^2 = \|\boldsymbol{Q}^\top\boldsymbol{Y} - \boldsymbol{Q}^\top\boldsymbol{Q}\boldsymbol{R}\boldsymbol{\beta}\|^2 \\ &= \|\boldsymbol{c} - \boldsymbol{R}\boldsymbol{\beta}\|^2, \end{aligned}$$

Es fácil notar que:

$$\|\boldsymbol{c} - \boldsymbol{R}\boldsymbol{\beta}\|^2 = \|\boldsymbol{c}_1 - \boldsymbol{R}_1\boldsymbol{\beta}\|^2 + \|\boldsymbol{c}_2\|^2.$$

Finalmente, el estimador de mínimos cuadrados $\widehat{\boldsymbol{\beta}}$ está dado por la solución del sistema triangular:

$$\boldsymbol{R}_1\widehat{\boldsymbol{\beta}} = \boldsymbol{c}_1. \qquad \text{(DTRTRS)}$$

El mínimo de la función objetivo está dado por $\|\boldsymbol{c}_2\|^2$. Note además que

$$s^2 = \frac{1}{n-p}\,\|\boldsymbol{c}_2\|^2, \qquad \text{(DLASSQ)}$$

corresponde al estimador insesgado de σ^2. Por otro lado,

$$\boldsymbol{X}^\top\boldsymbol{X} = (\boldsymbol{R}_1^\top, \boldsymbol{0})\boldsymbol{Q}^\top\boldsymbol{Q}\begin{pmatrix}\boldsymbol{R}_1\\ \boldsymbol{0}\end{pmatrix} = \boldsymbol{R}_1^\top\boldsymbol{R}_1.$$

De este modo,

$$\mathsf{Cov}(\widehat{\boldsymbol{\beta}}) = \sigma^2(\boldsymbol{R}_1^\top\boldsymbol{R}_1)^{-1} = \sigma^2\boldsymbol{R}_1^{-1}\boldsymbol{R}_1^{-\top}.$$

Definición 5 (SVD):

Sea $\boldsymbol{A} \in \mathbb{R}^{n\times p}$ con $\operatorname{rg}(\boldsymbol{A}) = r$, entonces existen matrices $\boldsymbol{U} \in \mathcal{O}_n$, $\boldsymbol{V} \in \mathcal{O}_p$, tal que

$$\boldsymbol{A} = \boldsymbol{U} \begin{pmatrix} \boldsymbol{D}_r & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{pmatrix} \boldsymbol{V}^\top,$$

donde $\boldsymbol{D}_r = \operatorname{diag}(\delta_1, \dots, \delta_r)$ con $\delta_1 \geq \delta_2 \geq \cdots \geq \delta_r > 0$, que son llamados valores singulares de $\boldsymbol{A}$.

Observación:

SVD para $\boldsymbol{A} \in \mathbb{R}^{n\times p}$ con $\operatorname{rg}(\boldsymbol{A}) = r$ puede ser escrita como:

$$\boldsymbol{A} = \boldsymbol{U}\boldsymbol{D}\boldsymbol{V}^\top,$$

con $\boldsymbol{U} \in \mathbb{R}^{n\times p}$ tal que $\boldsymbol{U}^\top\boldsymbol{U} = \boldsymbol{I}_p$, $\boldsymbol{D} = \operatorname{diag}(\delta_1, \dots, \delta_r)$ y $\boldsymbol{V} \in \mathcal{O}_r$.

SVD en estimación LS

Considere la descomposición valor singular (SVD)[12] de $\boldsymbol{X}$,

$$\boldsymbol{X} = \boldsymbol{U}\boldsymbol{D}\boldsymbol{V}^\top, \tag{DGESVD}$$

donde $\boldsymbol{U} \in \mathbb{R}^{n\times p}$ tal que $\boldsymbol{U}^\top\boldsymbol{U} = \boldsymbol{I}_p$, $\boldsymbol{D} = \operatorname{diag}(\delta_1, \dots, \delta_p)$ y $\boldsymbol{V} \in \mathcal{O}_p$.

De este modo, podemos escribir el modelo:

$$\boldsymbol{Y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{\epsilon} = \boldsymbol{U}\boldsymbol{D}\boldsymbol{\alpha} + \boldsymbol{\epsilon},$$

con $\boldsymbol{\alpha} = \boldsymbol{V}^\top\boldsymbol{\beta}$. Haciendo $\boldsymbol{Z} = \boldsymbol{U}^\top\boldsymbol{Y}$, tenemos el modelo en forma canónica:

$$\boldsymbol{Z} = \boldsymbol{D}\boldsymbol{\alpha} + \boldsymbol{\eta}, \qquad \boldsymbol{\eta} = \boldsymbol{U}^\top\boldsymbol{\epsilon},$$

donde

$$\mathsf{E}(\boldsymbol{\eta}) = \boldsymbol{0}, \qquad \mathsf{Cov}(\boldsymbol{\eta}) = \sigma^2\boldsymbol{U}^\top\boldsymbol{U} = \sigma^2\boldsymbol{I}_p.$$

[12] Linpack contiene la rutina DSVDC que es menos eficiente que su contraparte DGESVD desde LAPACK.

El estimador LS de $\boldsymbol{\alpha}$ en el modelo canónico es:

$$\widehat{\boldsymbol{\alpha}} = \boldsymbol{D}^{-1}\boldsymbol{Z}, \qquad \Rightarrow \qquad \widehat{\boldsymbol{\beta}} = \boldsymbol{V}\widehat{\boldsymbol{\alpha}}.$$

Además,

$$\|\boldsymbol{Y} - \boldsymbol{X}\widehat{\boldsymbol{\beta}}\|^2 = \|\boldsymbol{Y} - \boldsymbol{U}\boldsymbol{D}\boldsymbol{V}^\top\widehat{\boldsymbol{\beta}}\|^2 = \|\boldsymbol{Z} - \boldsymbol{D}\widehat{\boldsymbol{\alpha}}\|^2.$$

Finalmente,

$$\mathsf{Cov}(\widehat{\boldsymbol{\beta}}) = \sigma^2(\boldsymbol{X}^\top\boldsymbol{X})^{-1} = \sigma^2(\boldsymbol{V}\boldsymbol{D}^2\boldsymbol{V}^\top)^{-1} = \sigma^2\boldsymbol{V}\boldsymbol{D}^{-2}\boldsymbol{V}^\top.$$

Observación:

Cuando $\operatorname{rg}(\boldsymbol{X}) < p$, podemos considerar

$$\widehat{\boldsymbol{\alpha}} = \boldsymbol{D}^{-}\boldsymbol{Z},$$

y luego obtener $\widehat{\boldsymbol{\beta}} = \boldsymbol{V}\widehat{\boldsymbol{\alpha}}$.

Gradientes conjugados en regresión lineal

Considere:

$$\phi(\boldsymbol{\beta}) = \frac{1}{2}\|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}\|^2 = \frac{1}{2}(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta})^\top(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}).$$

El método Gradientes Conjugados (CG) produce la secuencia:

$$\boldsymbol{\beta}^{(k+1)} = \boldsymbol{\beta}^{(k)} + \lambda_k \boldsymbol{p}_k, \qquad k = 0, 1, \dots.$$

El algoritmo básico considera:

$$\lambda_k = \frac{\boldsymbol{p}_k^\top \boldsymbol{g}_k}{\boldsymbol{p}_k^\top \boldsymbol{X}^\top \boldsymbol{X}\boldsymbol{p}_k}, \qquad \boldsymbol{g}_k = \boldsymbol{X}^\top(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}^{(k)}).$$

(En efecto, $\partial\phi(\boldsymbol{\beta})/\partial\boldsymbol{\beta} = -\boldsymbol{X}^\top(\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{\beta})$) y actualizamos la dirección de búsqueda como:

$$\boldsymbol{p}_{k+1} = \boldsymbol{g}_{k+1} + \delta_k \boldsymbol{p}_k, \qquad \delta_k = -\frac{\boldsymbol{g}_{k+1}^\top \boldsymbol{p}_k}{\boldsymbol{p}_k^\top \boldsymbol{X}^\top \boldsymbol{X}\boldsymbol{p}_k}.$$

Se ha sugerido usar:

$$\lambda_k = \frac{\boldsymbol{p}_k^\top \boldsymbol{X}^\top \boldsymbol{y}}{\boldsymbol{p}_k^\top \boldsymbol{X}^\top \boldsymbol{X} \boldsymbol{p}_k},$$

y actualizar

$$\boldsymbol{p}_{k+1} = \boldsymbol{g}_{k+1} + \delta_{k+1} \boldsymbol{p}_k, \qquad \delta_{k+1} = -\frac{\boldsymbol{p}_k^\top \boldsymbol{X}^\top \boldsymbol{X} \boldsymbol{g}_k}{\boldsymbol{p}_k^\top \boldsymbol{X}^\top \boldsymbol{X} \boldsymbol{p}_k}.$$

Para hacer el proceso más simple es recomendable calcular

$$\boldsymbol{h}_k = \boldsymbol{X}^\top \boldsymbol{X} \boldsymbol{p}_k.$$

De este modo el requerimiento de almacenamiento del algoritmo es sólo $4p$.[13]

[13] Es decir, no hace falta formar la matriz $\boldsymbol{X}^\top \boldsymbol{X}$.

Gradientes conjugados en regresión lineal

Algoritmo 3: Gradientes conjugados para regresión lineal.

Entrada : Datos $\boldsymbol{X}$ y $\boldsymbol{y}$
Parámetros: Tolerancia τ.

begin
  Hacer $\boldsymbol{\beta} = \boldsymbol{0}$, $\boldsymbol{p} = \boldsymbol{g} = -\boldsymbol{X}^\top \boldsymbol{y}$, $\delta = 0$ y $\gamma = \|\boldsymbol{g}\|^2$
  while $\gamma > \tau$ **do**
    Calcular $\boldsymbol{h} = \boldsymbol{X}^\top \boldsymbol{X} \boldsymbol{p}$ y $u = \boldsymbol{p}^\top \boldsymbol{X}^\top \boldsymbol{X} \boldsymbol{p} = \boldsymbol{p}^\top \boldsymbol{h}$
    $v = \boldsymbol{g}^\top \boldsymbol{g}$
    $\lambda = -v/u$
    $\boldsymbol{\beta} = \boldsymbol{\beta} + \lambda\, \boldsymbol{p}$
    $\boldsymbol{g} = \boldsymbol{g} + \lambda\, \boldsymbol{h}$
    $\delta = \boldsymbol{g}^\top \boldsymbol{g}/v$
    $\boldsymbol{p} = \boldsymbol{g} + \delta\, \boldsymbol{p}$
  end
  return $\widehat{\boldsymbol{\beta}} = \boldsymbol{\beta}$
end

Puntajes adaptativos de Gesell (Mickey, Dunn y Clark, 1967)[15]

Ejemplo (Puntajes adaptativos de Gesell):

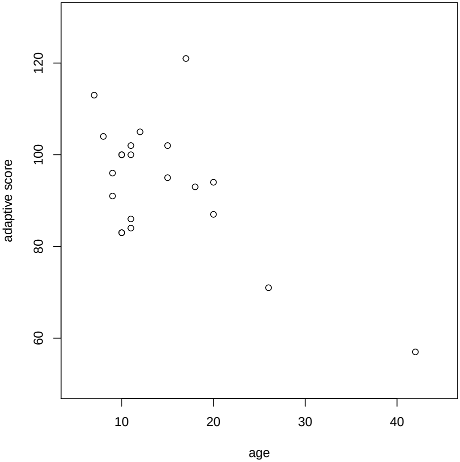
Estudio sobre la enfermedad cardiaca cianótica en niños. En este contexto x es la edad de los niños (en meses) al momento de decir su primera palabra, y Y es el puntaje adaptativo de Gesell (permite evaluar la etapa de desarrollo de un niño) para un conjunto de $n = 21$ niños.[14]

```
# base de datos
> load("gesell.rda")
> gesell
   age  score
1   15    95
2   26    71
3   10    83
4    9    91
5   15   102
6   20    87

...

20  11    86
21  10   100

```

[14] También podemos hacer: `gesell <- read.csv("gesell.csv")`

[15] Computers and Biomedical Research **1**, 105-109.

Puntajes adaptativos de Gesell (Mickey, Dunn y Clark, 1967)



Puntajes adaptativos de Gesell (Mickey, Dunn y Clark, 1967)

```
# Graficando los datos
> plot(score ~ age, data = gesell, ylim = c(50,130), xlim = c(5,45),
+      ylab = "adaptive score")

# Ajustando un modelo de regresión
> fm <- lm(score ~ age, data = gesell)

# Salida
> fm

Call:
lm(formula = score ~ age, data = gesell)

Coefficients:
(Intercept)          age
    109.874       -1.127
```

Puntajes adaptativos de Gesell (Mickey, Dunn y Clark, 1967)

```
# Salida un poco (no mucho) más extensa
> summary(fm)

Call:
lm(formula = score ~ age, data = gesell)

Residuals:
    Min      1Q  Median      3Q     Max
-15.604  -8.731   1.396   4.523  30.285

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 109.8738     5.0678  21.681 7.31e-15 ***
age          -1.1270     0.3102  -3.633  0.00177 **

Residual standard error: 11.02 on 19 degrees of freedom
Multiple R-squared:   0.41, Adjusted R-squared:  0.3789
F-statistic:  13.2 on 1 and 19 DF,  p-value: 0.001769
```

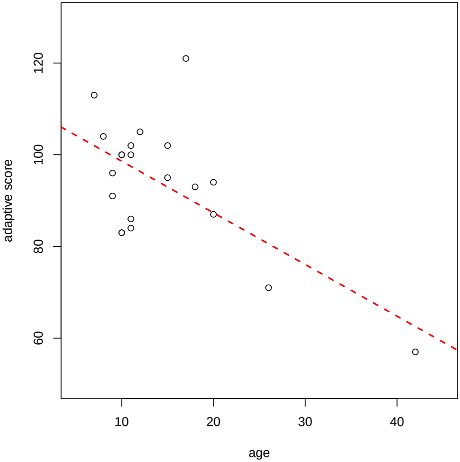
Puntajes adaptativos de Gesell (Mickey, Dunn y Clark, 1967)

```
# Explorando los objetos en R:
> attributes(fm)
$names
 [1] "coefficients"  "residuals"     "effects"       "rank"
 [5] "fitted.values" "assign"        "qr"            "df.residual"
 [9] "xlevels"       "call"          "terms"         "model"

$class
[1] "lm"

> o <- summary(fm)
> attributes(o)
$names
 [1] "call"          "terms"         "residuals"     "coefficients"
 [5] "aliased"       "sigma"         "df"            "r.squared"
 [9] "adj.r.squared" "fstatistic"    "cov.unscaled"

$class
[1] "summary.lm"
```

Puntajes adaptativos de Gesell (Mickey, Dunn y Clark, 1967)

```
# Extraer residuos y valores predichos
> res <- resid(fm)
> fit <- fitted(fm)

# Calcular algunas medidas globales
> logLik(fm)
'log Lik.' -79.14632 (df=3)

> deviance(fm) # sum(res^2)
[1] 2308.586

# Recta de regresión ajustada
> plot(score ~ age, data = gesell, ylim = c(50,130), xlim = c(5,45),
+       ylab = "adaptive score")
> abline(coef(fm), lty = 2, lwd = 2, col = "red")
```

Puntajes adaptativos de Gesell (Mickey, Dunn y Clark, 1967)



Ejemplo (Datos de cemento Portland):

Estudio experimental relacionando la emisión de calor durante la producción y endurecimiento de 13 muestras de cementos Portland. Woods et al. (1932) consideraron cuatro compuestos para los clinkers desde los que se produce el cemento.

La respuesta (Y) es la emisión de calor después de 180 días de curado, medido en calorías por gramo de cemento. Los regresores son los porcentajes de los cuatro compuestos: aluminato tricálcico (X_1), silicato tricálcico (X_2), ferrito aluminato tetracálcico (X_3) y silicato dicálcico (X_4).

[16] Industrial and Engineering Chemistry **24**, 1207-1214.

Cemento Portland (Woods, Steinour y Starke, 1932)

```
# base de datos
> load("portland.rda")
> portland
       y x1 x2 x3 x4
1   78.5  7 26  6 60
2   74.3  1 29 15 52
3  104.3 11 56  8 20
4   87.6 11 31  8 47
5   95.9  7 52  6 33
6  109.2 11 55  9 22
7  102.7  3 71 17  6
8   72.5  1 31 22 44
9   93.1  2 54 18 22
10 115.9 21 47  4 26
11  83.8  1 40 23 34
12 113.3 11 66  9 12
13 109.4 10 68  8 12

# en efecto,
> apply(portland[,-1], 1, sum)
 1  2  3  4  5  6  7  8  9 10 11 12 13
99 97 95 97 98 97 97 98 96 98 98 98 98

```

Cemento Portland (Woods, Steinour y Starke, 1932)

```
# carga biblioteca 'fastmatrix'
# disponible en: https://faosorios.github.io/fastmatrix/

> library(fastmatrix)
> fm <- ols(y ~ -1 + x1 + x2 + x3 + x4, data = portland,
+            method = "sweep")
> fm

Call:
ols(formula = y ~ -1 + x1 + x2 + x3 + x4, data = portland,
    method = "sweep")

Coefficients:
    x1      x2      x3      x4
2.1930  1.1533  0.7585  0.4863

Degrees of freedom: 13 total; 9 residual
Residual standard error: 2.417739

```

Cemento Portland (Woods, Steinour y Starke, 1932)

```
# alternativamente:
> fm <- ols(y ~ -1 + ., data = portland, method = "cg")
> fm

Call:
ols(formula = y ~ -1 + ., data = portland, method = "cg")

Coefficients:
    x1      x2      x3      x4
2.1930  1.1533  0.7585  0.4863

Degrees of freedom: 13 total; 9 residual
Residual standard error: 2.417739

```

Métodos disponibles: Gradiente conjugados (`"cg"`), Cholesky (`"chol"`), QR (`"qr"`), SVD (`"svd"`) y Sweep (`"sweep"`).

Cemento Portland (Woods, Steinour y Starke, 1932)

```
# Salida de función 'summary'
> summary(fm)

Call:
ols(formula = y ~ x1 + x2 + x3 + x4, data = portland)

Residuals:
    Min       1Q   Median       3Q      Max
-3.1750  -1.6709   0.2508   1.3783   3.9254

Coefficients:
              Estimate  Std. Error  t value  Pr(>|t|)
(Intercept) 62.4054   70.0710     0.8906   0.3991
x1           1.5511    0.7448     2.0827   0.0708
x2           0.5102    0.7238     0.7049   0.5009
x3           0.1019    0.7547     0.1350   0.8959
x4          -0.1441    0.7091    -0.2032   0.8441

Residual standard error: 2.446 on 8 degrees of freedom
Log-likelikood: -26.92

```

Cemento Portland (Woods, Steinour y Starke, 1932)

```
# Explorando los objetos en R:
> attributes(fm)
$names
[1] "coefficients"  "residuals"     "fitted.values" "RSS"
[5] "cov.unscaled"  "dims"          "call"          "xlevels"
[9] "terms"

$class
[1] "ols"

> o <- summary(fm)
> attributes(o)
$names
[1] "call"          "terms"         "residuals"    "coefficients"
[5] "sigma"         "df"            "cov.unscaled" "logLik"

$class
[1] "summary.ols"
```

Cemento Portland (Woods, Steinour y Starke, 1932)

```
# Extraer residuos y valores predichos
> res <- resid(fm)
> fit <- fitted(fm)

# log-verosimilitud
> logLik(fm)
'log Lik.' -26.91834 (df=6)

# Suma de cuadrados residual (RSS)
> deviance(fm)
[1] 47.86364
```

Observación:

Más adelante retomaremos el conjunto de datos de cemento Portland para evaluar el efecto de colinealidad entre las variables regresoras.

Referencias bibliográficas

Barlow, J.S. (1993).
Numerical aspects of solving linear least squares problems.
In C.R. Rao (Ed.), *Handbook of Statistics, Vol. 9*. Elsevier, 303-376.

Goodnight, J.H. (1979).
A tutorial on the SWEEP operator.
The American Statistician **33**, 149-158.

McIntosh, A. (1982).
Fitting Linear Models: An Application of Conjugate Gradients Algorithms.
Springer, New York.

Mickey, M.R., Dunn, O.J., Clark, V. (1967).
Note on the use of stepwise regression in detecting outliers.
Computers and Biomedical Research **1**, 105-109.

Woods, H., Steinour, H.H., Starke, H.R. (1932).
Effect of composition of Portland cement on heat evolved during hardening.
Industrial Engineering and Chemistry **24**, 1207–1214.