

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

Felipe Osorio

felipe.osorio@uv.cl

Aspectos numéricos de estimación LS en regresión lineal

Considere el **modelo de regresión lineal**:

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

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

$$\hat{\boldsymbol{\beta}} = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{Y}, \quad \text{con} \quad \text{Cov}(\hat{\boldsymbol{\beta}}) = \sigma^2 (\mathbf{X}^\top \mathbf{X})^{-1}.$$

Además,

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

Adicionalmente, si $\boldsymbol{\epsilon} \sim N_n(\mathbf{0}, \sigma^2 \mathbf{I})$, entonces¹

$$\begin{aligned} \hat{\boldsymbol{\beta}} &\sim N_p(\boldsymbol{\beta}, \sigma^2 (\mathbf{X}^\top \mathbf{X})^{-1}), \\ \frac{(n-p)s^2}{\sigma^2} &\sim \chi^2(n-p). \end{aligned}$$

¹En cuyo caso, $\hat{\boldsymbol{\beta}}$ 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:

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

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

$$\mathbf{X}^\top \mathbf{X} \hat{\beta} = \mathbf{X}^\top \mathbf{Y}.$$

Métodos habituales para obtener $\hat{\beta}$, 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

²Goodnight (1979). The American Statistician 33, 149-158.

³Único método disponible en función **lm** de R.

⁴McIntosh (1982). Lecture Notes in Statistics 10. Springer, New York.

Aspectos numéricos de estimación LS en regresión lineal

Observaciones:

- ▶ Cholesky⁵ y Sweep requieren formar las matrices:

$$\mathbf{X}^\top \mathbf{X}, \mathbf{X}^\top \mathbf{y}, \quad \text{y} \quad \begin{pmatrix} \mathbf{X}^\top \mathbf{X} & \mathbf{X}^\top \mathbf{Y} \\ \mathbf{Y}^\top \mathbf{X} & \mathbf{Y}^\top \mathbf{Y} \end{pmatrix},$$

respectivamente.

- ▶ QR⁶ y SVD descomponen la matriz de diseño \mathbf{X} y resuelven sistemas lineales (triangular y diagonal, respectivamente) mucho más pequeños ($n \gg p$).
- ▶ Note que $\kappa(\mathbf{X}^\top \mathbf{X}) = \kappa^2(\mathbf{X})$.
- ▶ 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 $A \in \mathbb{R}^{p \times p}$ es matriz **simétrica y definida positiva**, entonces existe una única matriz triangular superior $G \in \mathbb{R}^{p \times p}$ con elementos diagonales positivos tal que

$$A = G^T G$$

Observación:

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

$$G^T z = b, \quad \text{y} \quad Gx = z.$$

En efecto,

$$Ax = (G^T G)x = G^T (Gx) = G^T z = b.$$

Resultado 1 (Factorización Cholesky):

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

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

Observación:

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

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

En efecto,

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

Factorización Cholesky

Algoritmo 1: Factorización Cholesky

Entrada: Matriz $\mathbf{A} \in \mathbb{R}^{p \times p}$.

Salida : Factor Cholesky $\mathbf{G} \in \mathbb{R}^{p \times p}$.

```
1 begin
2    $g_{11} = \sqrt{a_{11}}$ .
3   for  $j = 2$  to  $p$  do
4      $| \quad g_{1j} = a_{1j}/g_{11}$ .
5   end
6   for  $i = 2$  to  $p$  do
7      $| \quad g_{ii} = \sqrt{a_{ii} - \sum_{k=1}^{i-1} g_{ki}^2}$ ,
8     for  $j = i + 1$  to  $n$  do
9        $| \quad g_{ij} = (a_{ij} - \sum_{k=1}^{i-1} g_{ki}g_{kj})/g_{ii}$ 
10    end
11  end
12 end
```

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

$$\mathbf{X}^\top \mathbf{X} \hat{\beta} = \mathbf{X}^\top \mathbf{Y}. \quad (1)$$

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

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

como

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

Es decir, podemos escribir:

$$RSS = \|\mathbf{Y} - \mathbf{X}\hat{\beta}\|^2 = \mathbf{Y}^\top \mathbf{Y} - \hat{\beta}^\top \mathbf{X}^\top \mathbf{X} \hat{\beta}.$$

Factorización Cholesky en estimación LS

Podemos resolver (1) usando la **descomposición Cholesky**, de

$$\mathbf{X}^\top \mathbf{X} = \mathbf{U}^\top \mathbf{U}, \quad (\text{DPOTRF})$$

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

$$\mathbf{U}^\top \mathbf{z} = \mathbf{X}^\top \mathbf{Y}, \quad \text{y} \quad \mathbf{U} \hat{\boldsymbol{\beta}} = \mathbf{z}, \quad (\text{DTRTRS})$$

para obtener s^2 considere

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

Invirtiendo \mathbf{U} (in-place)⁸, podemos hacer

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

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

Definición 1 (Operador Sweep):

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

⁹Disponible en la función [sweep.operator](#) de la biblioteca [fastmatrix](#).

Propiedades:

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

Observaciones:

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

Operador Sweep

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

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

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

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

con

$$\begin{aligned} B_{11} &= A_{11}^{-1}, & B_{12} &= A_{11}^{-1} A_{12}, \\ B_{21} &= -A_{21} A_{11}^{-1}, & B_{22} &= A_{22} - A_{21} A_{11}^{-1} A_{12}. \end{aligned}$$

Operador Sweep en estimación LS

Considere

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

luego

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

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

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

Definición 2 (Descomposición QR):

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

$$A = QR,$$

donde

$$R = \begin{pmatrix} R_1 \\ 0 \end{pmatrix}$$

con $R_1 \in \mathbb{R}^{p \times p}$ matriz triangular superior, aquí suponemos que $n \geq p$.

Observación:

Si $A = QR$ entonces

$$A^\top A = R^\top Q^\top QR = R^\top R = R_1^\top R_1,$$

y R_1 corresponde al factor Cholesky de $A^\top A$.

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

- ▶ $QQ^T = Q^T Q = I$.
- ▶ $\langle Qx, Qy \rangle = x^T Q^T Q y = x^T y = \langle x, y \rangle$.
- ▶ $\|Qx\| = \|x\|$.
- ▶ Si $B = Q^T A Q$, entonces A y B tienen los mismos valores propios para Q matriz ortogonal.

Existen diversas variantes del algoritmo para implementar la [descomposición QR](#). A continuación veremos una basada en [transformaciones Householder](#).

Problema 1:

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

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

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

Definición 3 (Reflexión):

Sea \mathbf{u} y \mathbf{v} **vectores ortonormales** y \mathbf{x} vector generado por \mathbf{u} y \mathbf{v} . Entonces

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

para escalares c_1, c_2 . El vector

$$\tilde{\mathbf{x}} = -c_1 \mathbf{u} + c_2 \mathbf{v},$$

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

Definición 4 (Transformación Householder):

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

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

Note que $\mathbf{H}\mathbf{x} = \tilde{\mathbf{x}}$, es decir \mathbf{H} es un reflector.

La **transformación Householder** satisface las siguientes propiedades:

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

Observación:

La operación $\mathbf{H}\mathbf{x}$ puede ser obtenida usando un `axpy`.¹⁰

¹⁰ Actualización del tipo: $\mathbf{y} \leftarrow \alpha \mathbf{x} + \mathbf{y}$.

Descomposición QR

La **descomposición QR** de una matriz $A \in \mathbb{R}^{n \times p}$ ($n > p$), puede ser construída a través de una **secuencia de matrices** Q_1, \dots, Q_p tales que

$$Q_p \cdots Q_1 A = \begin{pmatrix} R \\ 0 \end{pmatrix},$$

donde **todas** Q_1, \dots, Q_p son ortogonales. De este modo,

$$A = Q_1^\top \cdots Q_p^\top \begin{pmatrix} R \\ 0 \end{pmatrix} = Q \begin{pmatrix} R \\ 0 \end{pmatrix}.$$

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

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

Descomposición QR

Algoritmo 2: Descomposición QR

Entrada: Matriz $A \in \mathbb{R}^{n \times p}$.

Salida : Factores Q y R , matrices ortogonal y triangular superior, respectivamente.

```
1 begin
2   Hacer  $Q = I_n$  y  $R = A$ 
3   for  $i = 1$  to  $p$  do
4      $x = (R_{1i}, \dots, R_{pi})^\top$ 
5      $Q_i = \begin{pmatrix} I_{i-1} & \mathbf{0} \\ \mathbf{0} & M(x) \end{pmatrix}$ 
6     /*  $M(x)$  obtenido usando reflexiones Householder */
7      $Q = Q_i Q$ 
8      $R = Q_i R$ 
9   end
10   $Q = Q^\top$ 
11   $R = (R_{ij})$  para  $i, j = 1, \dots, p$ .
12 end
```

Descomposición QR en estimación LS

Considere la **descomposición QR** de \mathbf{X} , como:

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

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

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

El **estimador LS** minimiza la función objetivo:

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

Es fácil notar que:

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

Descomposición QR en estimación LS

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

$$R_1 \hat{\beta} = c_1. \quad (\text{DTRTRS})$$

El mínimo de la función objetivo está dado por $\|c_2\|^2$. Note además que

$$s^2 = \frac{1}{n-p} \|c_2\|^2, \quad (\text{DLASSQ})$$

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

$$X^T X = (R_1^T, 0) Q^T Q \begin{pmatrix} R_1 \\ 0 \end{pmatrix} = R_1^T R_1.$$

De este modo,

$$\text{Cov}(\hat{\beta}) = \sigma^2 (R_1^T R_1)^{-1} = \sigma^2 R_1^{-1} R_1^{-T}.$$

Descomposición valor singular (SVD)

Definición 5 (SVD):

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

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

donde $D_r = \text{diag}(\delta_1, \dots, \delta_r)$ con $\delta_1 \geq \delta_2 \geq \dots \geq \delta_r > 0$, que son llamados **valores singulares** de A .

Observación:

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

$$A = U D V^\top,$$

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

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

$$\mathbf{X} = \mathbf{U}\mathbf{D}\mathbf{V}^\top, \quad (\text{DGESVD})$$

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

De este modo, podemos escribir el modelo:

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

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

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

donde

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

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

El **estimador LS** de α en el modelo canónico es:

$$\hat{\alpha} = D^{-1}Z, \quad \Rightarrow \quad \hat{\beta} = V\hat{\alpha}.$$

Además,

$$\|Y - X\hat{\beta}\|^2 = \|Y - UDV^T\hat{\beta}\|^2 = \|Z - D\hat{\alpha}\|^2.$$

Finalmente,

$$\text{Cov}(\hat{\beta}) = \sigma^2(X^T X)^{-1} = \sigma^2(VD^2V^T)^{-1} = \sigma^2VD^{-2}V^T.$$

Observación:

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

$$\hat{\alpha} = D^-Z,$$

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

Gradientes conjugados en regresión lineal

Considere:

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

El **método Gradientes Conjugados (CG)** produce la secuencia:

$$\beta^{(k+1)} = \beta^{(k)} + \lambda_k \mathbf{p}_k, \quad k = 0, 1, \dots$$

El algoritmo básico considera:

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

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

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

Gradientes conjugados en regresión lineal

Se ha sugerido usar:

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

y actualizar

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

Para hacer el proceso más simple es recomendable calcular

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

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

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

Gradientes conjugados en regresión lineal

Algoritmo 3: Gradientes conjugados para regresión lineal.

Entrada : Datos X y y

Parámetros: Tolerancia τ .

```
1 begin
2   Hacer  $\beta = 0$ ,  $p = g = -X^T y$ ,  $\delta = 0$  y  $\gamma = \|g\|^2$ 
3   while  $\gamma > \tau$  do
4     Calcular  $h = X^T X p$  y  $u = p^T X^T X p = p^T h$ 
5      $v = g^T g$ 
6      $\lambda = -v/u$ 
7      $\beta = \beta + \lambda p$ 
8      $g = g + \lambda h$ 
9      $\delta = g^T g / v$ 
10     $p = g + \delta p$ 
11  end
12  return  $\hat{\beta} = \beta$ 
13 end
```

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

Ejemplo (Puntajes adaptativos de Gesell):

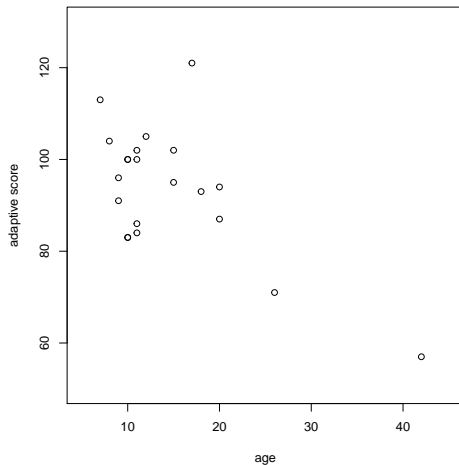
Estudio sobre la enfermedad cardíaca cianótica en niños. En este contexto x es la **edad de los niños** (en meses) al momento de decir su primera palabra, y Y es el **puntaje adaptativo de Gesell** (permite evaluar la etapa de desarrollo de un niño) para un conjunto de $n = 21$ niños.¹⁴

```
1 # base de datos
2 > load("gesell.rda")
3 > gesell
4   age score
5 1   15    95
6 2   26    71
7 3   10    83
8 4    9    91
9 5   15   102
10 6   20    87
11
12 ...
13
14 20   11    86
15 21   10   100
16
```

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

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

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



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

```
1 # Graficando los datos
2 > plot(score ~ age, data = gesell, ylim = c(50,130), xlim = c(5,45),
3 +       ylab = "adaptive score")
4
5 # Ajustando un modelo de regresión
6 > fm <- lm(score ~ age, data = gesell)
7
8 # Salida
9 > fm
10
11 Call:
12 lm(formula = score ~ age, data = gesell)
13
14 Coefficients:
15 (Intercept)          age
16    109.874         -1.127
```

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

```
1 # Salida un poco (no mucho) más extensa
2 > summary(fm)
3
4 Call:
5 lm(formula = score ~ age, data = gesell)
6
7 Residuals:
8      Min       1Q   Median       3Q      Max
9 -15.604  -8.731   1.396   4.523  30.285
10
11 Coefficients:
12             Estimate Std. Error t value Pr(>|t|)
13 (Intercept)  109.8738     5.0678   21.681 7.31e-15 ***
14 age          -1.1270     0.3102   -3.633 0.00177 **
15
16 Residual standard error: 11.02 on 19 degrees of freedom
17 Multiple R-squared:  0.41, Adjusted R-squared:  0.3789
18 F-statistic: 13.2 on 1 and 19 DF, p-value: 0.001769
```

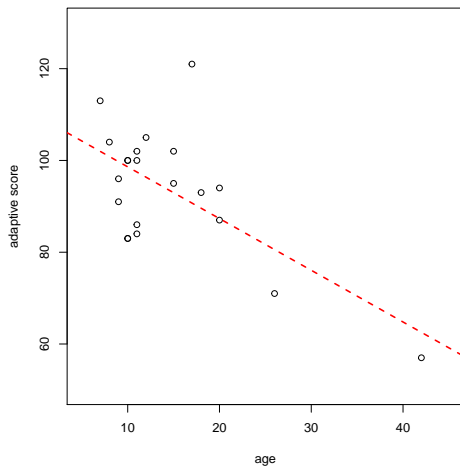
Puntajes adaptativos de Gesell (Mickey, Dunn y Clark, 1967)

```
1 # Explorando los objetos en R:
2 > attributes(fm)
3 $names
4 [1] "coefficients" "residuals" "effects" "rank"
5 [5] "fitted.values" "assign" "qr" "df.residual"
6 [9] "xlevels" "call" "terms" "model"
7
8 $class
9 [1] "lm"
10
11 > o <- summary(fm)
12 > attributes(o)
13 $names
14 [1] "call" "terms" "residuals" "coefficients"
15 [5] "aliased" "sigma" "df" "r.squared"
16 [9] "adj.r.squared" "fstatistic" "cov.unscaled"
17
18 $class
19 [1] "summary.lm"
```


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

```
1 # Extraer residuos y valores predichos
2 > res <- resid(fm)
3 > fit <- fitted(fm)
4
5 # Calcular algunas medidas globales
6 > logLik(fm)
7 'log Lik.' -79.14632 (df=3)
8
9 > deviance(fm) # sum(res^2)
10 [1] 2308.586
11
12 # Recta de regresión ajustada
13 > plot(score ~ age, data = gesell, ylim = c(50,130), xlim = c(5,45),
14 +       ylab = "adaptive score")
15 > 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 tetra cálcico (X_3) y silicato dicálcico (X_4).

¹⁶Industrial and Engineering Chemistry 24, 1207-1214.

Cemento Portland (Woods, Steinour y Starke, 1932)

```
1 # base de datos
2 > load("portland.rda")
3 > portland
4       y  x1  x2  x3  x4
5 1   78.5   7 26   6 60
6 2   74.3   1 29 15 52
7 3  104.3  11 56   8 20
8 4   87.6  11 31   8 47
9 5   95.9   7 52   6 33
10 6  109.2  11 55   9 22
11 7  102.7   3 71  17   6
12 8   72.5   1 31  22 44
13 9   93.1   2 54  18 22
14 10 115.9  21 47   4 26
15 11   83.8   1 40  23 34
16 12 113.3  11 66   9 12
17 13 109.4  10 68   8 12
18
19 # en efecto,
20 > apply(portland[,-1], 1, sum)
21 1  2  3  4  5  6  7  8  9 10 11 12 13
22 99 97 95 97 98 97 97 98 96 98 98 98 98
23
```

Cemento Portland (Woods, Steinour y Starke, 1932)

```
1 # carga biblioteca 'fastmatrix'
2 # disponible en: https://faosorios.github.io/fastmatrix/
3
4 > library(fastmatrix)
5 > fm <- ols(y ~ -1 + x1 + x2 + x3 + x4, data = portland,
6 +          method = "sweep")
7 > fm
8
9 Call:
10 ols(formula = y ~ -1 + x1 + x2 + x3 + x4, data = portland,
11      method = "sweep")
12
13 Coefficients:
14      x1      x2      x3      x4
15 2.1930  1.1533  0.7585  0.4863
16
17 Degrees of freedom: 13 total; 9 residual
18 Residual standard error: 2.417739
19
```

Cemento Portland (Woods, Steinour y Starke, 1932)

```
1 # alternativamente:
2 > fm <- ols(y ~ -1 + ., data = portland, method = "cg")
3 > fm
4
5 Call:
6 ols(formula = y ~ -1 + ., data = portland, method = "cg")
7
8 Coefficients:
9      x1      x2      x3      x4
10 2.1930  1.1533  0.7585  0.4863
11
12 Degrees of freedom: 13 total; 9 residual
13 Residual standard error: 2.417739
14
```

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

Cemento Portland (Woods, Steinour y Starke, 1932)

```
1 # Salida de función 'summary'
2 > summary(fm)
3
4 Call:
5 ols(formula = y ~ x1 + x2 + x3 + x4, data = portland)
6
7 Residuals:
8      Min       1Q   Median       3Q      Max
9 -3.1750 -1.6709  0.2508  1.3783  3.9254
10
11 Coefficients:
12             Estimate Std. Error t value Pr(>|t|)
13 (Intercept)  62.4054   70.0710   0.8906  0.3991
14 x1           1.5511    0.7448   2.0827  0.0708
15 x2           0.5102    0.7238   0.7049  0.5009
16 x3           0.1019    0.7547   0.1350  0.8959
17 x4          -0.1441    0.7091  -0.2032  0.8441
18
19 Residual standard error: 2.446 on 8 degrees of freedom
20 Log-likelihood: -26.92
21
```

Cemento Portland (Woods, Steinour y Starke, 1932)

```
1 # Explorando los objetos en R:
2 > attributes(fm)
3 $names
4 [1] "coefficients" "residuals" "fitted.values" "RSS"
5 [5] "cov.unscaled" "dims" "call" "xlevels"
6 [9] "terms"
7
8 $class
9 [1] "ols"
10
11 > o <- summary(fm)
12 > attributes(o)
13 $names
14 [1] "call" "terms" "residuals" "coefficients"
15 [5] "sigma" "df" "cov.unscaled" "logLik"
16
17 $class
18 [1] "summary.ols"
```


Cemento Portland (Woods, Steinour y Starke, 1932)

```
1 # Extraer residuos y valores predichos
2 > res <- resid(fm)
3 > fit <- fitted(fm)
4
5 # log-verosimilitud
6 > logLik(fm)
7 'log Lik.' -26.91834 (df=6)
8
9 # Suma de cuadrados residual (RSS)
10 > deviance(fm)
11 [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.