

UNA INTRODUCCIÓN A LOS PROBLEMAS DE MÍNIMOS CUADRADOS

F. VADILLO

RESUMEN. Los mínimos cuadrados es una herramienta muy utilizada desde su invención por Gauss y Legendre hacia el año 1800. En términos de álgebra lineal se trata de "resolver" sistemas lineales $\mathbf{Ax} = \mathbf{b}$ con más ecuaciones que incógnitas; el término resolver se entiende en el sentido de minimizar la suma de los cuadrados de las componentes del vector residuo $\mathbf{r} = \mathbf{Ax} - \mathbf{b}$.

Estas notas se dividen en dos partes, en la primera se estudia el problema de ajuste de curvas a conjuntos de datos que es uno de los principales orígenes de los problemas de mínimos cuadrados. En la segunda parte se comentan los métodos numéricos habituales para resolver los problemas de mínimos cuadrados.

ÍNDICE

1. Introducción	1
2. Ajustes de curvas por mínimos cuadrados	2
2.1. Rectas de regresión por mínimos cuadrados	2
2.2. Ajustes de otras curvas por mínimos cuadrados	3
2.3. Combinaciones lineales en mínimos cuadrados	5
3. Solución mínimo cuadrática de sistemas sobre-determinados	6
4. Descomposición valores singulares de una matriz	8
5. La factorización QR de una matriz	10
6. Métodos para calcular factorizaciones QR de matrices	11
6.1. Método de Gram-Schmidt	11
6.2. Reflectores de Householder	11
6.3. Computación de los valores singulares de una matriz	12
Referencias	14

1. INTRODUCCIÓN

El término mínimos cuadrados describe el problema muy frecuente de resolver sistemas de ecuaciones lineales sobre-determinados, esto es, sistemas lineales con más ecuaciones que incógnitas; en este caso, en lugar de resolver las ecuaciones de manera exacta, habitualmente no existe tal solución, se busca sólo minimizar la norma euclídea del vector residuo.

Received by the editors 2 de noviembre de 2022.
www.ehu.es/~mepvaarf.
Departamento de Matemáticas de la UPV/EHU.

Se considera un sistema lineal de m ecuaciones lineales con n incógnitas

$$(1.1) \quad \begin{pmatrix} a_{1,1} & \cdots & a_{1,n} \\ \vdots & \ddots & \vdots \\ \vdots & \ddots & \vdots \\ a_{m,1} & \cdots & a_{m,n} \end{pmatrix} \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} b_1 \\ \vdots \\ b_m \end{pmatrix},$$

que se supone es sobre-determinado, es decir, tiene más ecuaciones que incógnitas ($m > n$) por lo que normalmente no tiene una solución exacta. En notación matricial se escribe

$$(1.2) \quad \mathbf{A} \mathbf{x} = \mathbf{b}.$$

El problema mínimo cuadrático calcula el "mejor" \mathbf{x} en el sentido de minimizar el vector residuo $\mathbf{r} = \mathbf{A}\mathbf{x} - \mathbf{b}$ y se plantea de la siguiente manera: **Dada la matriz $\mathbf{A} \in \mathbb{R}^{m \times n}$ con $m > n$ y el vector $\mathbf{b} \in \mathbb{R}^m$, la solución mínimo cuadrática es el vector $\mathbf{x} \in \mathbb{R}^n$ que minimice la norma euclídea del residuo $\|\mathbf{r}\|_2$.**

Los paquetes de MATLAB: Curve Fitting Toolbox y Optimization Toolbox también incluyen funciones que minimizar otras norma, por ejemplo, la norma uno o la norma infinito del residuo del problema.

2. AJUSTES DE CURVAS POR MÍNIMOS CUADRADOS

2.1. Rectas de regresión por mínimos cuadrados. Una de las fuentes habituales de problemas de mínimos cuadrados son los problemas de ajustes de curvas. En la ciencia y la ingeniería los experimentos proporcionan un conjunto de datos $(x_1, y_1), \dots, (x_n, y_n)$, con las abscisas $\{x_k\}$ diferentes. El problema que se plantea es el de determinar una función $y = f(x)$ que relacione los datos lo mejor posible en algún sentido. Evidentemente, el resultado dependerá del tipo de función que se elija, por ejemplo, en la regresión $f(x) = ax + b$ es una recta, y para ajustar los parámetros libres se pueden minimizar uno de los siguientes tres valores:

El error máximo: $E_\infty(f) = \max\{|f(x_k) - y_k| : 1 \leq k \leq n\}$.

El error medio: $E_1(f) = \frac{1}{n} \sum_{k=1}^n |f(x_k) - y_k|$.

El error medio cuadrático: $E_2(f) = \sqrt{\frac{1}{n} \sum_{k=1}^n (f(x_k) - y_k)^2}$.

En el método de mínimos cuadrados el error que se minimiza es el error medio cuadrático. Por tanto, la recta de regresión ajusta los parámetros a y b para minimizar

$$E(a, b) = \sum_{k=1}^n (ax_k + b - y_k)^2,$$

que son la solución del sistema de ecuaciones lineales

$$\begin{aligned}\frac{\partial E}{\partial a} = 0 &\Rightarrow \left(\sum_{k=1}^n x_k^2 \right) a + \left(\sum_{k=1}^n x_k \right) b = \sum_{k=1}^n x_k y_k, \\ \frac{\partial E}{\partial b} = 0 &\Rightarrow \left(\sum_{k=1}^n x_k \right) a + nb = \sum_{k=1}^n y_k,\end{aligned}$$

denominadas **ecuaciones normales de Gauss** del problema.

Ejemplo 2.1. La población en los Estados Unidos de América durante el siglo XX ha seguido la evolución indicada en la tabla adjunta, se pide hallar la recta de regresión y pronostican el número de habitante en al año 2010.

año	millones de habitantes
1900	75.995
1910	91.972
1920	105.711
1930	123.203
1940	131.669
1950	150.697
1960	179.323
1970	203.212
1980	226.505
1990	249.633
2000	281.422

El programa `lsline.m` de [6] resuelve directamente el sistema normal y la recta de regresión que resulta es

$$y = 2,0253 \cdot x - 3783,9,$$

Para el año 2010 pronostica 286.912 millones de habitantes. En la figura ?? se ha representado los puntos y la recta de regresión.

2.2. Ajustes de otras curvas por mínimos cuadrados. Suponga que ahora se quisieran ajustar los datos $(x_1, y_1), \dots, (x_n, y_n)$ por el método de mínimos cuadrados usando una curva exponencial

$$(2.1) \quad y = c \cdot e^{ax},$$

donde a y c son ahora los parámetros a elegir. El valor que se debe minimizar es

$$E(a, c) = \sum_{k=1}^n (c \cdot e^{ax_k} - y_k)^2.$$

Igualando a cero las dos derivadas parciales primeras de $E(a, c)$ se llega al sistema de las dos ecuaciones

$$\begin{aligned}c \sum_{k=1}^n x_k e^{2ax_k} - \sum_{k=1}^n x_k y_k e^{ax_k} &= 0, \\ c \sum_{k=1}^n e^{ax_k} - \sum_{k=1}^n y_k e^{ax_k} &= 0,\end{aligned}$$

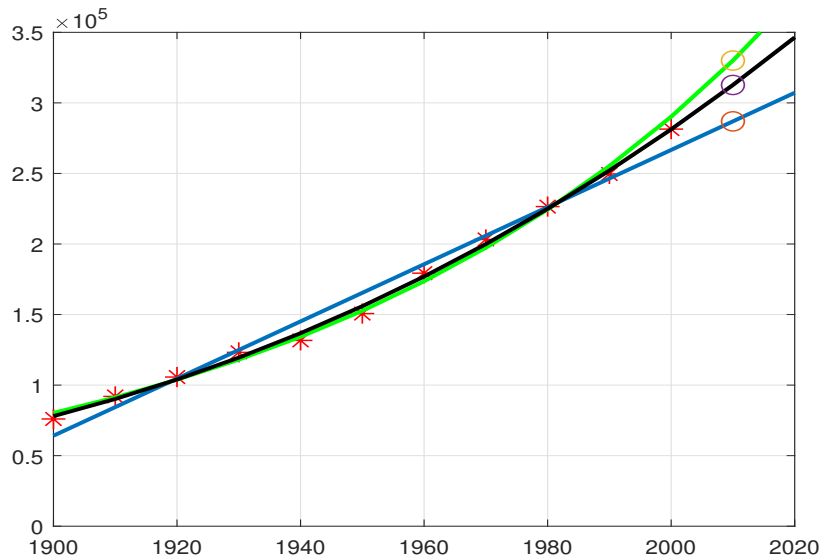


FIGURA 1. Recta de regresión en azul, ajuste exponencial en verde y ajuste polinomial en negro.

ecuaciones **no lineales para las incógnitas a y c** , ecuaciones evidentemente complicadas de resolver. Para evitar esta dificultad, lo más sencillo es **linealizar los datos**. Suponiendo que los datos $y_k > 0$, se toman logaritmos en (2.1) para obtener $\ln y = ax + \ln c$, y después se hace el cambio de variable

$$u = x, \quad v = \ln y, \quad b = \ln c,$$

con lo que resulta la recta

$$v = au + b,$$

con los datos transformados $(u_k, v_k) = (x_k, \ln(y_k))$ para $k = 1, \dots, n$, problema lineal que ya se sabe resolver.

Ejemplo 2.2. En el ejemplo de la población americana, la aproximación exponencial que se obtiene es

$$(2.2) \quad y = 1,9908 \cdot 10^{-9} \cdot e^{0,0129x},$$

dibujada en color verde en la figura 1 y la estimación en el año 2010 es de 329.9944 millones de habitantes.

La técnica de linealización de los datos se utiliza para ajustar otros tipos de curvas con un cambio de variable adecuado. Por ejemplo, si $f(x) = \frac{a}{x} + b$ el cambio que linealiza el problema es $u = \frac{1}{x}$ y $v = y$. En la tabla adjunta se muestran algunos cambios de variables que linealizan los problemas.

Función $y = f(x)$	Cambio de variables
$y = \frac{a}{x} + b$	$u = \frac{1}{x}, v = y$
$y = \frac{d}{x+c}$	$u = xy, v = y$
$y = \frac{1}{ax+b}$	$u = x, v = \frac{1}{y}$
$y = \frac{x}{ax+b}$	$u = \frac{1}{x}, v = \frac{1}{y}$
$y = a \ln x + b$	$u = \ln x, v = y$
$y = c \cdot e^{ax}$	$u = x, v = \ln y$
$y = \frac{1}{(ax+b)^2}$	$u = x, v = \frac{1}{\sqrt{y}}$
$y = c \cdot x \cdot e^{dx}$	$u = x, v = \ln\left(\frac{y}{x}\right)$
$y = \frac{l}{1+c \cdot e^{ax}}$	$u = x, v = \ln\left(\frac{l}{y} - 1\right)$

2.3. Combinaciones lineales en mínimos cuadrados. El problema de las combinaciones lineales en mínimos cuadrados se formula de la siguiente manera: dados m puntos $\{(x_k, y_k)\}$ y un conjunto de n funciones linealmente independientes $\{f_j(x)\}$, lo habitual es que $m > n$, se trata de encontrar n coeficientes $\{c_j\}$ tales que la función

$$(2.3) \quad f(x) = \sum_{j=1}^n c_j f_j(x),$$

minimice la suma de los cuadrados de los residuos

$$E(c_1, \dots, c_n) = \sum_{k=1}^m (f(x_k) - y_k)^2 = \sum_{k=1}^m \left(\left(\sum_{j=1}^n c_j f_j(x_k) \right) - y_k \right)^2.$$

Dicho mínimo se alcanza donde $\frac{\partial E}{\partial c_i} = 0$ para $i = 1, \dots, n$, es decir, los coeficientes c_j son la solución del sistema de **ecuaciones normales de Gauss**

$$\sum_{k=1}^m \left(\left(\sum_{j=1}^n c_j f_j(x_k) \right) - y_k \right) f_i(x_k) = 0, \quad i = 1, \dots, n.$$

Intercambiando el orden de los sumatorios el sistema anterior se puede escribir de la forma

$$(2.4) \quad \sum_{j=1}^n \left(\sum_{k=1}^m f_i(x_k) f_j(x_k) \right) c_j = \sum_{k=1}^m f_i(x_k) y_k, \quad i = 1, \dots, n.$$

Definiendo la siguiente matriz y vectores

$$\mathbf{F} = \begin{pmatrix} f_1(x_1) & \cdots & f_n(x_1) \\ \vdots & \ddots & \vdots \\ \vdots & \ddots & \vdots \\ f_1(x_m) & \cdots & f_n(x_m) \end{pmatrix}, \quad \mathbf{c} = \begin{pmatrix} c_1 \\ \vdots \\ c_n \end{pmatrix}, \quad \mathbf{y} = \begin{pmatrix} y_1 \\ \vdots \\ y_m \end{pmatrix},$$

este sistema de ecuaciones lineales de $n \times n$ se puede escribir en notación matricial

$$(2.5) \quad \mathbf{F}^T \mathbf{F} \mathbf{c} = \mathbf{F}^T \mathbf{y},$$

donde \mathbf{F}^T indica la matriz transpuesta y la incógnita es el vector \mathbf{c} , que es también la solución mínimo cuadrática del sistema sobre-determinado

$$(2.6) \quad \mathbf{F} \mathbf{c} = \mathbf{y},$$

con más ecuaciones que incógnitas porque $m > n$.

Ejemplo 2.3. Siguiendo con el ejemplo de la población americana en el siglo XX, se tratara de ajustar los datos a un polinomio de grados 3

$$(2.7) \quad y \approx c_3 x^3 + c_2 x^2 + c_1 x + c_0.$$

La solución del sistema normal es:

$$\begin{aligned} c_0 &= -4,25822238574126, & c_1 &= 0,00802427140030, \\ c_2 &= -0,00000496111699, & c_3 &= 0,00000000101028, \end{aligned}$$

con la gráfica en color negro de la figura 1 que pronostica 312.6913 millones de habitantes para el año 2010.

El programa `censusgui.m` de [7] dibuja ajustes polinomiales de diferentes grados y calcula la estimación de población para el año 2010.

3. SOLUCIÓN MÍNIMO CUADRÁTICA DE SISTEMAS SOBRE-DETERMINADOS

El resto de este capítulo se buscarán métodos eficaces para calcular la solución mínimo cuadrática de sistemas de ecuaciones lineales sobre-determinados $\mathbf{Ax} = \mathbf{b}$ con $\mathbf{A} \in \mathbb{R}^{m \times n}$, $m > n$. Por lo tanto, se debe minimizar la función en la variable \mathbf{x}

$$\begin{aligned} \|\mathbf{r}(\mathbf{x})\|_2^2 &= (\mathbf{Ax} - \mathbf{b})^T (\mathbf{Ax} - \mathbf{b}) = (\mathbf{x}^T \mathbf{A}^T - \mathbf{b}^T) (\mathbf{Ax} - \mathbf{b}) \\ &= \mathbf{x}^T \mathbf{A}^T \mathbf{Ax} - \mathbf{x}^T \mathbf{A}^T \mathbf{b} - \mathbf{b}^T \mathbf{Ax} + \mathbf{b}^T \mathbf{b} \\ &= \mathbf{x}^T \mathbf{A}^T \mathbf{Ax} - 2 \mathbf{x}^T \mathbf{A}^T \mathbf{b} + \mathbf{b}^T \mathbf{b}, \end{aligned}$$

Cuyos puntos estacionarios resuelven el sistema lineal

$$\frac{\partial}{\partial \mathbf{x}} \|\mathbf{r}(\mathbf{x})\|_2^2 = 2 \mathbf{A}^T \mathbf{Ax} - 2 \mathbf{A}^T \mathbf{b} = 0,$$

denominado **sistema normal** del problema

$$(3.1) \quad \mathbf{A}^T \mathbf{Ax} = \mathbf{A}^T \mathbf{b},$$

donde $\mathbf{A}^T \mathbf{A}$ es una matriz simétrica $n \times n$, es decir, que si hubiera miles de observaciones (m fuera muy grande) y unos pocos parámetros (n pequeña), el sistema normal es pequeño. Además, sin $\text{rank}(\mathbf{A}) = n$, la matriz $\mathbf{A}^T \mathbf{A}$ es definida positiva y se puede aplicar Cholesky que es más barato.

Además, la solución del sistema normal (3.1) verifica que $\mathbf{x} = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \mathbf{b}$, esta matriz que multiplica al vector \mathbf{b} es denominada la **pseudoinversa de la matriz \mathbf{A}** que se suele escribir de la forma

$$(3.2) \quad \mathbf{A}^\dagger = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T.$$

El concepto de pseudoinversa de una matriz es debido a Moore y Penrose y generaliza y extiende el clásico concepto de matriz inversa de una matriz cuadrada. Observe que para una matriz cuadrada $\mathbf{C} \in \mathbb{R}^{n \times n}$ no singular

$$\mathbf{C}^\dagger = (\mathbf{C}^T \mathbf{C})^{-1} \mathbf{C}^T = \mathbf{C}^{-1} (\mathbf{C}^T)^{-1} \mathbf{C}^T = \mathbf{C}^{-1}.$$

En general, la pseudoinversa es inversa por la izquierda pero no por la derecha porque

$$\begin{aligned} \mathbf{A}^\dagger \mathbf{A} &= (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \mathbf{A} = \mathbf{I}, \\ \mathbf{A} \mathbf{A}^\dagger &= \mathbf{A} (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T. \end{aligned}$$

Finalmente el algoritmo para resolver el problema de mínimos cuadrados via la ecuación normal quedaría:

Algoritmo 3.1. Dadas la matriz \mathbf{A} y el vector \mathbf{b} .

1. Calcule $\mathbf{B} = \mathbf{A}^T \mathbf{A}$ e $\mathbf{y} = \mathbf{A}^T \mathbf{b}$.
2. Compute la factorización de Cholesky de $\mathbf{B} = \mathbf{G} \mathbf{G}^T$.
3. Resuelva el sistema triangular inferior $\mathbf{G} \mathbf{z} = \mathbf{y}$.
4. Resuelva el sistema triangular superior $\mathbf{G}^T \mathbf{x} = \mathbf{z}$.

El principal inconveniente de los sistemas normales para resolver problemas de mínimos cuadrados es la exactitud porque **habitualmente aparecen números de condición muy altos**.

Ejemplo 3.2. Considere el sistema $\mathbf{A} \mathbf{x} = \mathbf{b}$ con

$$\mathbf{A} = \begin{pmatrix} 1 & 0 & 1 \\ 2 & 3 & 5 \\ 5 & 3 & -2 \\ 3 & 5 & 4 \\ -1 & 6 & 3 \end{pmatrix}, \quad \mathbf{b} = \begin{pmatrix} 4 \\ -2 \\ 5 \\ -2 \\ 1 \end{pmatrix}.$$

La matriz del sistema normal es

$$\mathbf{B} = \mathbf{A}^T \mathbf{A} = \begin{pmatrix} 40 & 30 & 10 \\ 30 & 79 & 47 \\ 10 & 47 & 55 \end{pmatrix}, \quad \mathbf{y} = \mathbf{A}^T \mathbf{b} = \begin{pmatrix} 18 \\ 5 \\ -21 \end{pmatrix}.$$

Entonces la solución del sistema $\mathbf{B} \mathbf{x} = \mathbf{y}$ es $\mathbf{x} = (0,3472, 0,3990, -0,7859)^T$ con un residuo considerable

$$\mathbf{r} = \mathbf{b} - \mathbf{A} \mathbf{x} = (4,4387, 0,0381, 0,495, -1,893, 1,131)^T.$$

4. DESCOMPOSICIÓN VALORES SINGULARES DE UNA MATRIZ

La descomposición valores singulares de una matriz está basada en el hecho de que para cualquier matriz $\mathbf{A} \in \mathbb{C}^{m \times n}$, es posible encontrar un conjunto de valores no negativos σ_i y vectores $\mathbf{u}_i, \mathbf{v}_i$ tales

$$\mathbf{A}\mathbf{v}_i = \sigma_i\mathbf{u}_i.$$

Los valores σ_i son los **valores singulares** de la matriz y los vectores $\mathbf{u}_i \in \mathbb{C}^m$ y $\mathbf{v}_i \in \mathbb{C}^n$ son los vectores singulares por la izquierda y derecha respectivamente que siempre se puede elegir ortogonales. Es conveniente suponer que los valores singulares están ordenados en sentido descendente, es decir, $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_p > 0$. Para una matriz $\mathbf{A} \in \mathbb{R}^{m \times n}$, sus valores singulares representan los semiejes principales de la hiperelipse imagen de la esfera unidad por la aplicación que representa dicha matriz (ver [8] página 25).

Definición 4.1. Dada una matriz $\mathbf{A} \in \mathbb{C}^{m \times n}$, una factorización

$$\mathbf{A} = \mathbf{U} \cdot \mathbf{\Sigma} \cdot \mathbf{V}^*,$$

es denominada **descomposición en valores singular (SVD)** (singular values decomposition) de \mathbf{A} , si $\mathbf{U} \in \mathbb{C}^{m \times m}$, $\mathbf{V} \in \mathbb{C}^{n \times n}$ son matrices unitarias cuyas columnas son los vectores principales por la izquierda y derecha respectivamente, y la matriz $\mathbf{\Sigma} = \text{diag}(\sigma_1, \dots, \sigma_p) \in \mathbb{R}^{m \times n}$, donde los σ_i son los valores singulares de la matriz con $p = \min(m, n)$.

Para $m > n$ esta descomposición se puede representar de la siguiente manera:

$$\begin{pmatrix} \mathbf{A} \\ m \times n \end{pmatrix} = \begin{pmatrix} \mathbf{U} \\ m \times m \end{pmatrix} \begin{pmatrix} \mathbf{\Sigma} \\ m \times n \end{pmatrix} \begin{pmatrix} \mathbf{V}^* \\ n \times n \end{pmatrix}.$$

Observe que las últimas $m-n$ columnas de \mathbf{U} no contribuyen a la descomposición $\mathbf{A} = \mathbf{U} \cdot \mathbf{\Sigma} \cdot \mathbf{V}^*$, es decir, $\mathbf{A} = \hat{\mathbf{U}} \cdot \hat{\mathbf{\Sigma}} \cdot \mathbf{V}^*$ donde $\hat{\mathbf{U}} \in \mathbb{C}^{m \times n}$ son las n primeras columnas de \mathbf{U} , y $\hat{\mathbf{\Sigma}} \in \mathbb{C}^{n \times n}$ son las n primeras filas de $\mathbf{\Sigma}$, dicha nueva descomposición es conocida como **SVD reducida**.

$$\begin{pmatrix} \mathbf{A} \\ m \times n \end{pmatrix} = \begin{pmatrix} \hat{\mathbf{U}} \\ m \times n \end{pmatrix} \begin{pmatrix} \hat{\mathbf{\Sigma}} \\ n \times n \\ \mathbf{0} \\ n \times n \end{pmatrix} \begin{pmatrix} \mathbf{V}^* \\ n \times n \end{pmatrix}.$$

Teorema 4.2. *Existencia y unicidad de la SVD* Toda matriz $\mathbf{A} \in \mathbb{C}^{m \times n}$ tiene una descomposición en valores singulares y además sus valores singulares son únicos.

Demostración. Ver [8] página 29. \square

El siguiente teorema muestra algunas propiedades interesantes de los valores singulares de una matriz

Teorema 4.3. Sean $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_r > 0 = \dots = 0$ los valores singulares de la matriz \mathbf{A} , entonces

1. El rango de A es r , el número de valores singulares no nulos.
2. $\|\mathbf{A}\|_2 = \sigma_1$.
3. Los valores singulares no nulos de \mathbf{A} son las raíces cuadradas de los autovalores no nulos de $\mathbf{A}^* \mathbf{A}$.
4. Si $\mathbf{A} = \mathbf{A}^*$ sus valores singulares son los valores absolutos de los autovalores de la matriz \mathbf{A} .
5. Si $\mathbf{A} \in \mathbb{C}^{m \times m}$, entonces $|\det(\mathbf{A})| = \prod_{j=1}^m \sigma_j$.

Demostración. Ver lección 5 de [8]. □

La SVD de una matriz sirve para calcular su pseudoinversa utilizando el siguiente teorema

Teorema 4.4. Si en la descomposición valores singulares de la matriz A

$$\Sigma = \begin{pmatrix} D \\ 0 \end{pmatrix}, \quad D = \text{diag}(\sigma_1, \dots, \sigma_n), \quad \sigma_1 \geq \dots \geq \sigma_n > 0,$$

entonces

$$\mathbf{A}^\dagger = V \Sigma^+ U^*, \quad \text{donde } \Sigma^+ = \begin{pmatrix} D^{-1} & 0 \end{pmatrix}.$$

Volviendo al problema de mínimos cuadrados y su sistema normal (3.1), la solución se expresaba en términos de la matriz pseudoinversa (3.2) para la que se define su número de condición

Definición 4.5. Dada una matriz $\mathbf{A} \in \mathbb{C}^{m \times n}$ con $m \geq n$ y los valores singulares $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_r > 0$, el número de condición en la norma $\|\cdot\|$ es

$$\kappa(\mathbf{A}) = \begin{cases} \|\mathbf{A}\| \cdot \|\mathbf{A}^\dagger\|, & \text{si } \text{rand}(\mathbf{A}) = n, \\ +\infty, & \text{otros.} \end{cases}$$

Como \mathbf{A}^\dagger recuerda a la matriz inversa, esta definición es consistente con la previa para matrices cuadradas.

Lema 4.6. Dada una matriz $\mathbf{A} \in \mathbb{C}^{m \times n}$ con $m \geq n$ y los valores singulares $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_n > 0$, el número de condición para la norma $\|\cdot\|_2$ es

$$\kappa_2(\mathbf{A}) = \frac{\sigma_1}{\sigma_n}.$$

Como consecuencia de este lema la matriz del sistema normal tiene un número de condición

$$\kappa_2(\mathbf{A}^T \mathbf{A}) = \frac{\lambda_1}{\lambda_n} = \frac{\sigma_1^2}{\sigma_n^2} = \kappa_2(\mathbf{A})^2,$$

es decir, el número de condición del sistema normal es el cuadrado del número de condición de la matriz original, lo que resulta un grave problema que se agudiza en caso como el ejemplo 3.2. En general: **no es aconsejable resolver los problemas de mínimos cuadrados calculando las soluciones de sus sistemas normales, sobretodo cuando σ_n es pequeño, porque son sistemas mal condicionados.** Por ejemplo, MATLAB evita las ecuaciones normales, de hecho el **backslash**, es decir $\mathbf{A} \setminus \mathbf{b}$ calcula la solución del sistema $\mathbf{A} \mathbf{x} = \mathbf{b}$ pero no utiliza las ecuaciones normales.

Los algoritmos que calculan la descomposición valores singulares de una matriz son muy técnicos y por ello superan el contenido de estas notas, en las referencias [3], [2], [9], [8] y [1] se pueden consultar los detalles. De los tres métodos más habituales para resolver problemas de mínimos cuadrados: sistema normal, descomposición valores singulares y factorización QR que se estudiará a continuación, el método de la pseudoinversa es el que tiene mejores propiedades de estabilidad para problemas con rango no máximo. En MATLAB `[U,S,V]=svd(A)` calcula la descomposición valores singulares de la matriz \mathbf{A} , y el comando `pinv(A)` computa la matriz pseudoinversa.

5. LA FACTORIZACIÓN QR DE UNA MATRIZ

Unos de los métodos clásicos para resolver sistemas sobre-determinados utiliza la factorización QR de la matriz del sistema. Dada una matriz \mathbf{A} de tamaño $m \times n$ de rango completo, la idea consiste en factorizar

$$\mathbf{A} = \mathbf{Q} \begin{pmatrix} \mathbf{R} \\ \mathbf{0} \end{pmatrix},$$

donde \mathbf{Q} es una matriz ortogonal $m \times m$ y \mathbf{R} triangular superior de tamaño $n \times n$. Es fácil comprobar que si $\text{rank}(\mathbf{A}) = n$ entonces \mathbf{R} es no singular.

Suponiendo conocida una factorización QR de la \mathbf{A} , entonces se debe minimizar

$$\|\mathbf{r}\|_2 = \|\mathbf{b} - \mathbf{Ax}\|_2 = \left\| \mathbf{b} - \mathbf{Q} \begin{pmatrix} \mathbf{R} \\ \mathbf{0} \end{pmatrix} \mathbf{x} \right\|_2 = \left\| \mathbf{Q}^T \mathbf{b} - \begin{pmatrix} \mathbf{R} \\ \mathbf{0} \end{pmatrix} \mathbf{x} \right\|_2,$$

si ahora se escribe $\mathbf{Q}^T \mathbf{b} = \begin{pmatrix} \mathbf{c} \\ \mathbf{d} \end{pmatrix}$, con n componente para \mathbf{c} y las restante $m - n$ para \mathbf{d} , entonces

$$\|\mathbf{r}\|_2^2 = \|\mathbf{b} - \mathbf{Ax}\|_2^2 = \|\mathbf{c} - \mathbf{Rx}\|_2^2 + \|\mathbf{d}\|_2^2,$$

y por tanto como el valor de \mathbf{x} que lo minimiza es cuando $\mathbf{Rx} = \mathbf{c}$, valor mínimo que vale $\|\mathbf{d}\|$, y en consecuencia para resolver el problema de mínimos cuadrados se tiene el siguiente algoritmo

Algoritmo 5.1. Dadas la matriz \mathbf{A} y el vector \mathbf{b} .

1. Calcule la factorización QR de la matriz $\mathbf{A} = \mathbf{Q} \begin{pmatrix} \mathbf{R} \\ \mathbf{0} \end{pmatrix}$.
2. Compute el vector $\begin{pmatrix} \mathbf{c} \\ \mathbf{d} \end{pmatrix} = \mathbf{Q}^T \mathbf{b}$
3. Resuelva el sistema triangular superior $\mathbf{Rx} = \mathbf{c}$.

De este algoritmo todavía no se conoce los métodos y algoritmos para calcular factorizaciones QR de matrices que es el tema que se trata en la sección siguiente.

6. MÉTODOS PARA CALCULAR FACTORIZACIONES QR DE MATRICES

6.1. Método de Gram-Schmidt. Los textos clásicos de algebra lineal siempre explican el método de Gram-Schmidt para calcular una base ortonormal de un subespacio (vera por ejemplo [4] o el clásico [5]). Dicho método aplicado a las columnas de la matriz $\mathbf{A} \in \mathbb{C}^{m \times n}$ con $m \geq n$ sería el siguiente algoritmo

Algoritmo 6.1. Dadas la matriz $\mathbf{A} = (\mathbf{a}_1, \dots, \mathbf{a}_n)$.

```

 $\mathbf{q}_1 = \mathbf{a}_1 / \|\mathbf{a}_1\|$ 
for  $j = 2 : n$ 
     $\mathbf{q}_j = \mathbf{a}_j$ 
    for  $i = 1 : j - 1$ 
         $r_{ij} = \langle \mathbf{q}_j, \mathbf{q}_i \rangle$ 
         $\mathbf{q}_j = \mathbf{q}_j - r_{ij} \mathbf{q}_i$ 
    end
     $r_{jj} = \|\mathbf{q}_j\|$ 
     $\mathbf{q}_j = \mathbf{q}_j / r_{jj}$ 
end

```

Después elegir $\mathbf{q}_{n+1}, \dots, \mathbf{q}_m$ para que $\mathbf{q}_1, \dots, \mathbf{q}_m$ sea una base ortonormal $\mathbf{Q} = (\mathbf{q}_1, \dots, \mathbf{q}_m)$ y $(\mathbf{R})_{ij} = r_{ij}$ para $i \leq j$ y ceros en los otros.

Aunque en realidad las $m - n$ últimas columnas de \mathbf{Q} no se utilizan porque

$$\begin{pmatrix} & & & \\ & \mathbf{A} & & \\ & & & \\ m \times n & & & \end{pmatrix} = \begin{pmatrix} & & & \\ & \hat{\mathbf{Q}} & & \\ & & & \\ m \times n & & & \end{pmatrix} \begin{pmatrix} \hat{\mathbf{R}} \\ n \times n \\ \mathbf{0} \end{pmatrix}$$

con $\hat{\mathbf{Q}} = (\mathbf{q}_1, \dots, \mathbf{q}_n)$. Es decir que $\mathbf{A} = \hat{\mathbf{Q}}\hat{\mathbf{R}}$ en lo que se denomina la **factorización reducida**.

Este algoritmo permite demostrar el siguiente teorema de existencia de factorización QR

Teorema 6.2. *Cualquier matriz $\mathbf{A} \in \mathbb{R}^{m \times n}$ con $m \geq n$ se puede factorizar de la forma $\mathbf{A} = \mathbf{QR}$ donde $\mathbf{Q} \in \mathbb{R}^{m \times m}$ es ortogonal y $\mathbf{R} \in \mathbb{R}^{m \times n}$ triangular superior.*

El algoritmo de orto-normalización de Gram-Schmidt es numéricamente inestable y por tanto no es aconsejable para calcular factorizaciones QR (Lecture 9 de [8]).

6.2. Reflectores de Householder. Los reflectores de Householder son transformaciones matricial que se utilizan en algunos de lo más importantes algoritmos numéricos del Análisis Numérico Matricial. Aquí se usan para resolver problemas de mínimos cuadrados, pero también son importantes en los problemas de autovalores y valores singulares de matrices.

Formalmente, un reflector de Householder es una matriz de la forma

$$\mathbf{H} = \mathbf{I} - 2\mathbf{u}\mathbf{u}^T,$$

donde el vector $\mathbf{u} \in \mathbb{R}^n$ es unitario ($\|\mathbf{u}\|_2^2 = \mathbf{u}^T \mathbf{u} = 1$) e \mathbf{I} es la matriz identidad. A continuación se listan algunas propiedades de estas matrices:

1. Los reflectores de Householder son matrices simétricas: $\mathbf{H}^T = \mathbf{H}$.

2. Los reflectores de Householder son matrices ortogonales: $\mathbf{H}^{-1} = \mathbf{H}^T$ y por tanto $\mathbf{H}^2 = \mathbf{I}$.
3. $\mathbf{H}\mathbf{u} = \mathbf{u} - 2\mathbf{u}\mathbf{u}^T\mathbf{u} = -\mathbf{u}$.
4. Si \mathbf{v} es un vector ortogonal al vector \mathbf{u} , entonces $\mathbf{H}\mathbf{v} = \mathbf{v} - 2\mathbf{u}\mathbf{u}^T\mathbf{v} = \mathbf{v}$.

Geoméricamente es fácil observar que el reflector \mathbf{H} a cada vector de \mathbb{R}^n le lleva a su reflexión respecto del hiperplano ortogonal al vector \mathbf{u} que lo define.

La propiedad que se utiliza en el posterior algoritmo se recoge en el siguiente teorema:

Teorema 6.3. *Para todo par de vectores $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$ con las misma norma euclídea, el reflector construido con el vector $\mathbf{u} = \frac{\mathbf{x} - \mathbf{y}}{\|\mathbf{x} - \mathbf{y}\|_2}$ lleva un vector en el otro.*

Algoritmo 6.4. Dadas la matriz $\mathbf{A} \in \mathbb{R}^{m \times n}$ con $m \geq n$.

```

Q = I y R = A
for  $k = 1 : n$ 
     $\mathbf{u} = (r_{kk}, \dots, r_{mk}) \in \mathbb{R}^{m-k+1}$ 
     $\bar{\mathbf{v}} = \text{sign}(u_1)\|\mathbf{u}\|_2\mathbf{e}_1 + \mathbf{u}$  donde  $\mathbf{e}_1 = (1, 0, \dots, 0)^T \in \mathbb{R}^{m-k+1}$ 
     $\mathbf{v} = \bar{\mathbf{v}}/\|\bar{\mathbf{v}}\|_2$ 
     $\mathbf{H}_k = (\mathbf{I}_{m-k+1} - 2\mathbf{v}\mathbf{v}^T) \in \mathbb{R}^{(m-k+1) \times (m-k+1)}$ 
     $\mathbf{Q}_k = \begin{pmatrix} \mathbf{I}_{k-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{H}_k \end{pmatrix}$ 
    R =  $\mathbf{Q}_k$ R
    Q =  $\mathbf{Q}\mathbf{Q}_k$ 
end

```

Comentarios importantes:

1. Este algoritmo calcula las matrices ortogonales \mathbf{Q}_k y para obtener finalmente las matrices

$$\mathbf{Q} = \mathbf{Q}_1 \cdots \mathbf{Q}_{n-1}, \quad \mathbf{R} = \mathbf{Q}_{n-1} \cdots \mathbf{Q}_1 \mathbf{A}.$$

2. En el algoritmo 5.1 no se necesita explícitamente calcular la matriz \mathbf{Q} pero si calcular el producto de $\mathbf{Q}^T\mathbf{b}$, entonces en el algoritmo se puede reemplazar el calculo de $\mathbf{Q} = \mathbf{Q}\mathbf{Q}_k$ por $\mathbf{b} = \mathbf{Q}_k\mathbf{b}$ para llegar a $\mathbf{Q}_{n-1} \cdots \mathbf{Q}_1\mathbf{b} = (\mathbf{Q}_1 \cdots \mathbf{Q}_{n-1})^T\mathbf{b} = \mathbf{Q}^T\mathbf{b}$.
3. El coste computacionales de este algoritmo es aproximadamente $2mn^2 - \frac{2}{3}n^3$.
4. **Es un algoritmo numéricamente estable** (Lecture 16 de [8]).

En MATLAB el comando $[\mathbf{Q}, \mathbf{R}] = \text{qr}(\mathbf{A})$ calcula la factorización QR de la matriz \mathbf{A} y si se quiere ordenan los término de la matriz \mathbf{R} de la forma decreciente, es decir, $|r_{11}| \geq |r_{22}| \geq \dots \geq |r_{nn}|$, basta escribir $[\mathbf{Q}, \mathbf{R}, \mathbf{P}] = \text{qr}(\mathbf{A})$ tal que $\mathbf{A}\mathbf{P} = \mathbf{Q}\mathbf{R}$, esta opción es interesante para obtener fácilmente el rango del problema.

6.3. Computación de los valores singulares de una matriz. Finalizaremos este capítulo con una breve descripción del algoritmo de Golub y Reinsch de 1971 para computar los valores singulares de una matriz \mathbf{A} . Dicho método tiene dos etapa:

1. Utilizando reflectores de Householder se reduce la matriz \mathbf{A} a una forma bi-diagonal de la forma siguiente.

$$\mathbf{B} = \begin{pmatrix} * & * & & & \\ & * & * & & \\ & & \ddots & \ddots & \\ & & & * & * \\ & & & & * \end{pmatrix}.$$

Suponiendo que la matriz fuera de 5×4

$$\mathbf{A} = \begin{pmatrix} * & * & * & * \\ * & * & * & * \\ * & * & * & * \\ * & * & * & * \\ * & * & * & * \end{pmatrix} \rightarrow \mathbf{U}_1^T \mathbf{A} = \begin{pmatrix} * & * & * & * \\ 0 & * & * & * \\ 0 & * & * & * \\ 0 & * & * & * \\ 0 & * & * & * \end{pmatrix},$$

después se aplica otra transformación ortogonal por la derecha de la forma

$$\mathbf{U}_1^T \mathbf{A} \mathbf{V}_1 = \begin{pmatrix} * & * & 0 & 0 \\ 0 & * & * & * \\ 0 & * & * & * \\ 0 & * & * & * \\ 0 & * & * & * \end{pmatrix}.$$

En el siguiente paso

$$\mathbf{U}_2^T \mathbf{U}_1^T \mathbf{A} \mathbf{V}_1 \mathbf{V}_2 = \begin{pmatrix} * & * & 0 & 0 \\ 0 & * & * & 0 \\ 0 & 0 & * & * \\ 0 & 0 & * & * \\ 0 & 0 & * & * \end{pmatrix},$$

y así hasta llegar a la forma bi-diagonal \mathbf{B} . Evidentemente, el número de transformaciones ortogonales por la derecha e izquierda no necesariamente serán la misma.

2. Después de obtener la forma bi-diagonal \mathbf{B} , hay varias maneras de proceder. La más sencilla es calcular los autovalores de la matriz tridiagonal $\mathbf{B}^T \mathbf{B}$ porque son los cuadrados de los valores singulares de \mathbf{A} (ver [3], [2], [9], [8], [1], [8]).

En MATLAB `[U,S,V]=svd(A)` devuelve la descomposición en valores singulares de la matriz \mathbf{A} , y el comando `pinv(A)` computa su pseudoinversa.

REFERENCIAS

1. R. Bulirsch and J. Stoer, *Introduction to Numerical Analysis*, Springer, 1980.
2. G.H. Golub and C.F. Van Loan, *Matrix Computations*, The Johns Hopkins University Press, 1989.
3. N.J. Higham, *Accuracy and Stability of Numerical Algorithms*, SIAM, 1996.
4. B. Korman, *Álgebra lineal con aplicaciones y Matlab*, Pearson, 1999.
5. S. Lang, *Introduction to Linear Algebra (Undergraduate Texts in Mathematics) 2nd edition*, Springer, 1997.
6. J.F. Mathews and K.D. Fink, *Métodos Numéricos con MATLAB. Tercera edición*, Prentice Hall, 1999.
7. C.B. Moler, *Numerical Computing with MATLAB*, SIAM, 2004.
8. L.N. Trefethen and D. Bau, *Numerical Linear Algebra*, SIAM, 1997.
9. D.S. Watkins, *Fundamentals of Matrix Computations*, John Wiley, 1991.