MAT-468: Sesión 6, Mínimos cuadrados no lineales

Felipe Osorio

http://fosorios.mat.utfsm.cl

Departamento de Matemática, UTFSM

Mínimos cuadrados no lineales: Método Gauss-Newton

Considere el conjunto de ecuaciones de regresión:

$$Y_i = f(\boldsymbol{x}_i; \boldsymbol{\theta}) + \epsilon_i, \qquad i = 1, \ldots, n,$$

que pueden ser escritas de forma conveniente como:

$$\boldsymbol{Y} = \boldsymbol{f}(\boldsymbol{\theta}) + \boldsymbol{\epsilon},$$

donde

$$\boldsymbol{Y} = \begin{pmatrix} Y_1 \\ \vdots \\ Y_n \end{pmatrix}, \qquad \boldsymbol{f}(\boldsymbol{\theta}) = \begin{pmatrix} f(\boldsymbol{x}_1; \boldsymbol{\theta}) \\ \vdots \\ f(\boldsymbol{x}_n; \boldsymbol{\theta}) \end{pmatrix}, \qquad \boldsymbol{\epsilon} = \begin{pmatrix} \epsilon_1 \\ \vdots \\ \epsilon_n \end{pmatrix}.$$

Mínimos cuadrados no lineales: Método Gauss-Newton

Además, consideraremos los siguientes supuestos de momentos:

$$\mathsf{E}(\boldsymbol{\epsilon}) = \mathbf{0}, \qquad \mathsf{Cov}(\boldsymbol{\epsilon}) = \sigma^2 \boldsymbol{I}_n,$$

y defina:

$$S(\boldsymbol{\theta}) = \sum_{i=1}^{n} \{Y_i - f(\boldsymbol{x}_i; \boldsymbol{\theta})\}^2 = \|\boldsymbol{Y} - \boldsymbol{f}(\boldsymbol{\theta})\|^2.$$

El objetivo de la siguiente sección será introducir un procedimiento para obtener $\widehat{\boldsymbol{\theta}}$ como:

$$\widehat{\boldsymbol{\theta}} := \arg\min_{\theta} S(\boldsymbol{\theta})$$

Observación: Una aplicación importante de este problema en estadística es regresión no lineal.

Mínimos cuadrados no lineales: Método Gauss-Newton

Estimadores usados en regresión no lineal pueden ser caracterizados como formas lineales y cuadráticas bastante similares a las que surgen en regresión lineal.

En efecto, sea

$$\boldsymbol{F}(\boldsymbol{\theta}) = \frac{\partial \boldsymbol{f}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}^\top} = \Big(\frac{\partial f(\boldsymbol{x}_i;\boldsymbol{\theta})}{\partial \theta_j}\Big).$$

Note por otro lado que:

$$\begin{aligned}\frac{\partial S(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} &= \frac{\partial}{\partial \boldsymbol{\theta}}(\boldsymbol{Y}-\boldsymbol{f}(\boldsymbol{\theta}))^\top(\boldsymbol{Y}-\boldsymbol{f}(\boldsymbol{\theta})) = 2\Big(\frac{\partial}{\partial \boldsymbol{\theta}}(\boldsymbol{Y}-\boldsymbol{f}(\boldsymbol{\theta})\Big)^\top(\boldsymbol{Y}-\boldsymbol{f}(\boldsymbol{\theta})) \\ &= -2\boldsymbol{F}^\top(\boldsymbol{\theta})(\boldsymbol{Y}-\boldsymbol{f}(\boldsymbol{\theta})).\end{aligned}$$

De modo que, $\widehat{\boldsymbol{\theta}}$ es solución de la ecuación de estimación:

$$\boldsymbol{\Psi}(\boldsymbol{\theta}) = \boldsymbol{F}^\top(\boldsymbol{\theta})(\boldsymbol{Y}-\boldsymbol{f}(\boldsymbol{\theta})).$$

Mínimos cuadrados no lineales: Método Gauss-Newton

Considere una estimación inicial $\boldsymbol{\theta}^{(0)}$ y considere la aproximación de primer orden:

$$\boldsymbol{f}(\boldsymbol{\theta}) \approx \boldsymbol{f}(\boldsymbol{\theta}^{(0)}) + \boldsymbol{F}(\boldsymbol{\theta}^{(0)})(\boldsymbol{\theta} - \boldsymbol{\theta}^{(0)}),$$

de este modo,

$$\begin{aligned} S(\boldsymbol{\theta}) &= \|\boldsymbol{Y} - \boldsymbol{f}(\boldsymbol{\theta})\|^2 \\ &\approx \widetilde{S}(\boldsymbol{\theta}) \\ &= \|\boldsymbol{Y} - \boldsymbol{f}(\boldsymbol{\theta}^{(0)}) - \boldsymbol{F}(\boldsymbol{\theta}^{(0)})(\boldsymbol{\theta} - \boldsymbol{\theta}^{(0)})\|^2 \\ &= \|\boldsymbol{e}^{(0)} - \boldsymbol{F}^{(0)}(\boldsymbol{\theta} - \boldsymbol{\theta}^{(0)})\|^2, \end{aligned}$$

donde $\boldsymbol{e}^{(0)} = \boldsymbol{Y} - \boldsymbol{f}(\boldsymbol{\theta}^{(0)})$ y $\boldsymbol{F}^{(0)} = \boldsymbol{F}(\boldsymbol{\theta}^{(0)})$.

Observación: $\widetilde{S}(\boldsymbol{\theta})$ es una función cuadrática en $\boldsymbol{\theta}$.

Mínimos cuadrados no lineales: Método Gauss-Newton

Diferenciando $\widetilde{S}(\boldsymbol{\theta})$ con relación a $\boldsymbol{\theta}$ obtenemos,

$$\begin{aligned}\frac{\partial \widetilde{S}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} &= 2\Big(\frac{\partial}{\partial \boldsymbol{\theta}}(\boldsymbol{e}^{(0)} - \boldsymbol{F}^{(0)}(\boldsymbol{\theta} - \boldsymbol{\theta}^{(0)})\Big)^{\top}(\boldsymbol{e}^{(0)} - \boldsymbol{F}^{(0)}(\boldsymbol{\theta} - \boldsymbol{\theta}^{(0)})) \\ &= -2\boldsymbol{F}^{(0)^{\top}}(\boldsymbol{e}^{(0)} - \boldsymbol{F}^{(0)}(\boldsymbol{\theta} - \boldsymbol{\theta}^{(0)})),\end{aligned}$$

igualando a cero y resolviendo, obtenemos que para $\boldsymbol{F}^{(0)}$ matriz de rango completo, $\widetilde{S}(\boldsymbol{\theta})$ es mínimo para

$$\boldsymbol{F}^{(0)^{\top}}\boldsymbol{F}^{(0)}(\boldsymbol{\theta} - \boldsymbol{\theta}^{(0)}) = \boldsymbol{F}^{(0)^{\top}}\boldsymbol{e}^{(0)},$$

es decir

$$\widehat{\boldsymbol{\theta}} = \boldsymbol{\theta}^{(0)} + (\boldsymbol{F}^{(0)^{\top}}\boldsymbol{F}^{(0)})^{-1}\boldsymbol{F}^{(0)^{\top}}\boldsymbol{e}^{(0)}.$$

Mínimos cuadrados no lineales: Método Gauss-Newton

Diferenciando $\widetilde{S}(\boldsymbol{\theta})$ con relación a $\boldsymbol{\theta}$ obtenemos,

$$\begin{aligned}\frac{\partial \widetilde{S}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} &= 2\Big(\frac{\partial}{\partial \boldsymbol{\theta}}(\boldsymbol{e}^{(0)} - \boldsymbol{F}^{(0)}(\boldsymbol{\theta} - \boldsymbol{\theta}^{(0)})\Big)^{\top}(\boldsymbol{e}^{(0)} - \boldsymbol{F}^{(0)}(\boldsymbol{\theta} - \boldsymbol{\theta}^{(0)})) \\ &= -2\boldsymbol{F}^{(0)^{\top}}(\boldsymbol{e}^{(0)} - \boldsymbol{F}^{(0)}(\boldsymbol{\theta} - \boldsymbol{\theta}^{(0)})),\end{aligned}$$

igualando a cero y resolviendo, obtenemos que para $\boldsymbol{F}^{(0)}$ matriz de rango completo, $\widetilde{S}(\boldsymbol{\theta})$ es mínimo para

$$\boldsymbol{F}^{(0)^{\top}}\boldsymbol{F}^{(0)}(\boldsymbol{\theta} - \boldsymbol{\theta}^{(0)}) = \boldsymbol{F}^{(0)^{\top}}\boldsymbol{e}^{(0)},$$

es decir

$$\widehat{\boldsymbol{\theta}} = \boldsymbol{\theta}^{(0)} + (\boldsymbol{F}^{(0)^{\top}}\boldsymbol{F}^{(0)})^{-1}\boldsymbol{F}^{(0)^{\top}}\boldsymbol{e}^{(0)}.$$

Mínimos cuadrados no lineales: Método Gauss-Newton

Finalmente obtenemos la iteración

$$\boldsymbol{\theta}^{(k+1)} = \boldsymbol{\theta}^{(k)} + \boldsymbol{\delta}^{(k)}, \qquad k = 0, 1, \ldots$$

donde el incremento $\boldsymbol{\delta}^{(k)}$ es solución del sistema de ecuaciones lineales (sobredeterminadas)

$$\boldsymbol{F}^{(k)} \boldsymbol{\delta}^{(k)} = \boldsymbol{e}^{(k)}.$$

algoritmo que es conocido como método Gauss-Newton.

Observación: Note que $\boldsymbol{\delta}^{(k)}$ corresponde a un problema LS con solución:

$$\boldsymbol{\delta}^{(k)} = (\boldsymbol{F}^{(k)^\top} \boldsymbol{F}^{(k)})^{-1} \boldsymbol{F}^{(k)^\top} \boldsymbol{e}^{(k)}.$$

Mínimos cuadrados no lineales: Método Gauss-Newton

Para asegurar que el incremento de Gauss, $\boldsymbol{\delta}^{(k)}$ produce direcciones de descenso, se puede realizar una reducción de paso (step-halving) tal que

$$S(\boldsymbol{\theta}^{(k)} + \lambda\boldsymbol{\delta}^{(k)}) < S(\boldsymbol{\theta}^{(k)}),$$

donde $\lambda \in (0, 1]$ es el largo de paso.

Datos de Puromycin (Treolar, 1974)

Modelo Michaelis-Menten: usado para el estudio de cinética de enzimas.

Permite estudiar la relación entre *velocidad inicial* de una reacción enzimática a la concentración de un substrato x a través de la ecuación:

$$f(x, \boldsymbol{\theta}) = \frac{V_m x}{K + x}, \qquad \boldsymbol{\theta} = (V_m, K)^\top.$$

Diferenciando f con relación a V_m y K, obtenemos

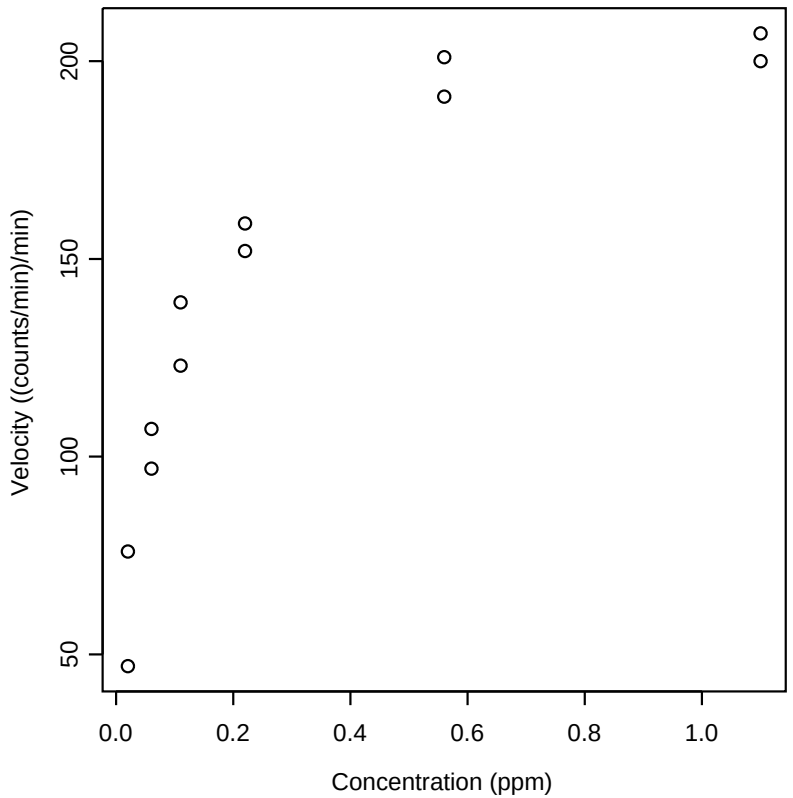
$$\frac{\partial f}{\partial V_m} = \frac{x}{K + x}, \qquad \frac{\partial f}{\partial K} = -\frac{V_m x}{(K + x)^2}$$

Note que el modelo Michaelis-Menten puede ser transformado en un modelo lineal, como

$$\frac{1}{f} = \frac{1}{V_m} + \frac{K}{V_m}\frac{1}{x} = \beta_1 + \beta_2 z.$$

Datos de Puromycin (Treolar, 1974)

Velocidad de reacción versus concentración del substrato para un experimento de Puromycin



Ventajas del modelo de regresión no lineal

- Más flexibilidad en la construcción/elección del modelo.
- la función de modelo se basa en la teoría respecto del mecanismo que produce la respuesta (no es un modelo empírico).
- predicciones fuera del rango de observación puede ser realizadas con *mayor confianza*.
- Parámetros tiene un significado físico y por tanto son de interés primário.

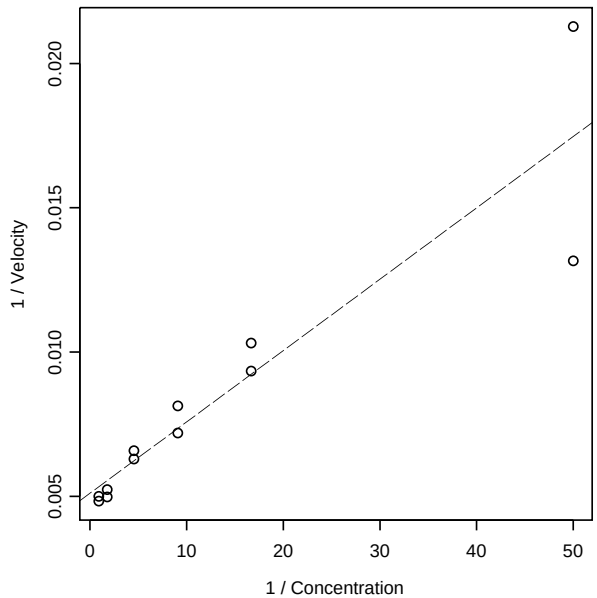
Desventajas del modelo de regresión no lineal

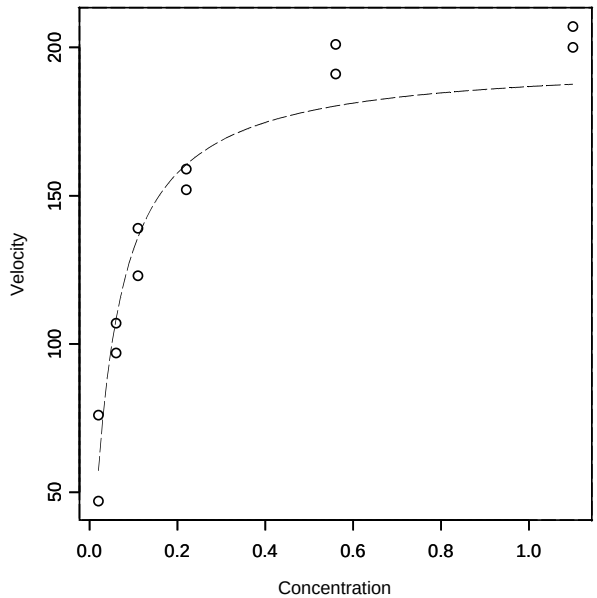
- Estimación de parámetros no tiene forma explícita. Se debe utilizar algoritmos iterativos.
- Necesidad de proveer estimaciones iniciales para el procedimiento de estimación.
- Errores estándar, intervalos de confianza, sólo son aproximados. Mayor precisión en la variabilidad de estimaciones requiere de mayor esfuerzo computacional.
- Posibilidad de óptimos múltiples y/o falsa convergencia.

Observaciones

- Transformación de los datos también involucra transformar los disturbios aleatorios.
- Parámetros transformados carecen de interpretación.
- Por supuesto, una gran cantidad de modelos de interés no son linealizables.

Datos de Puromycin





Datos de Puromycin

Realizando el ajuste en el modelo linealizado (usando mínimos cuadrados), obtenemos

$$\widehat{\boldsymbol{\beta}} = (0.005107, 0.000247)^\top,$$

luego

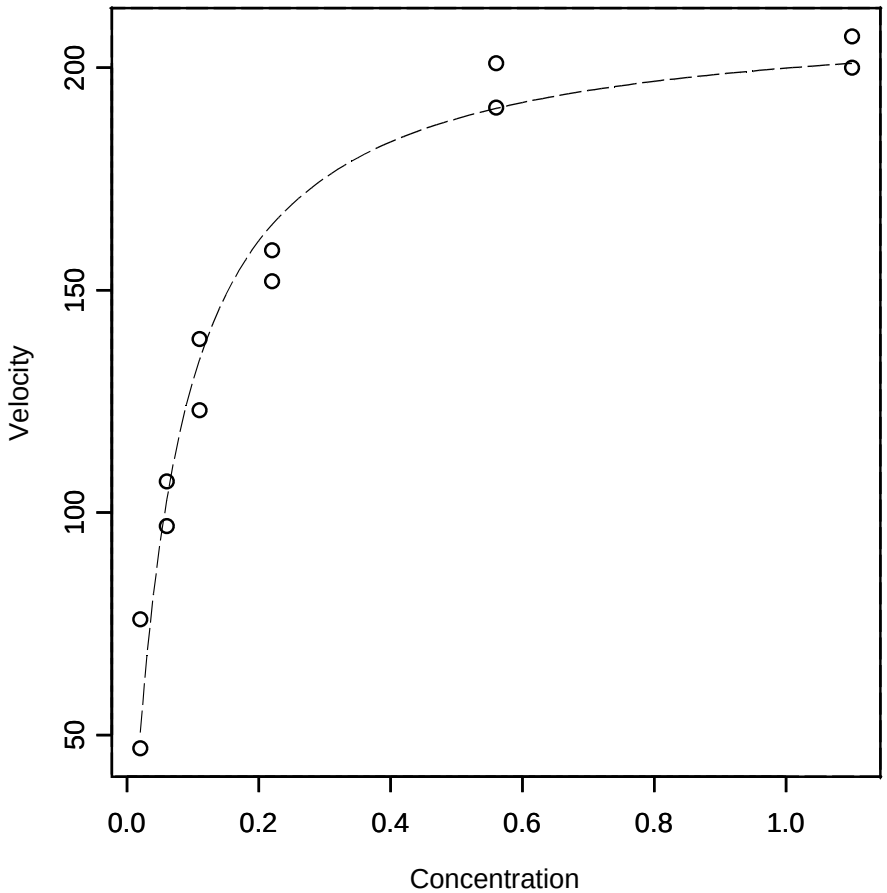
$$\widehat{\boldsymbol{\theta}} = (195.80, 0.04841)^\top,$$

Mientras que, usando minimos cuadrados no lineales obtuvimos[1]

$$\widehat{\boldsymbol{\theta}} = (212.68, 0.06412)^\top.$$

[1] Se consideró $\widehat{\boldsymbol{\theta}}^{(0)} = (205, 0.08)^\top$

Datos de Puromycin



Datos de Puromycin (Treolar, 1974)

Conjunto de datos

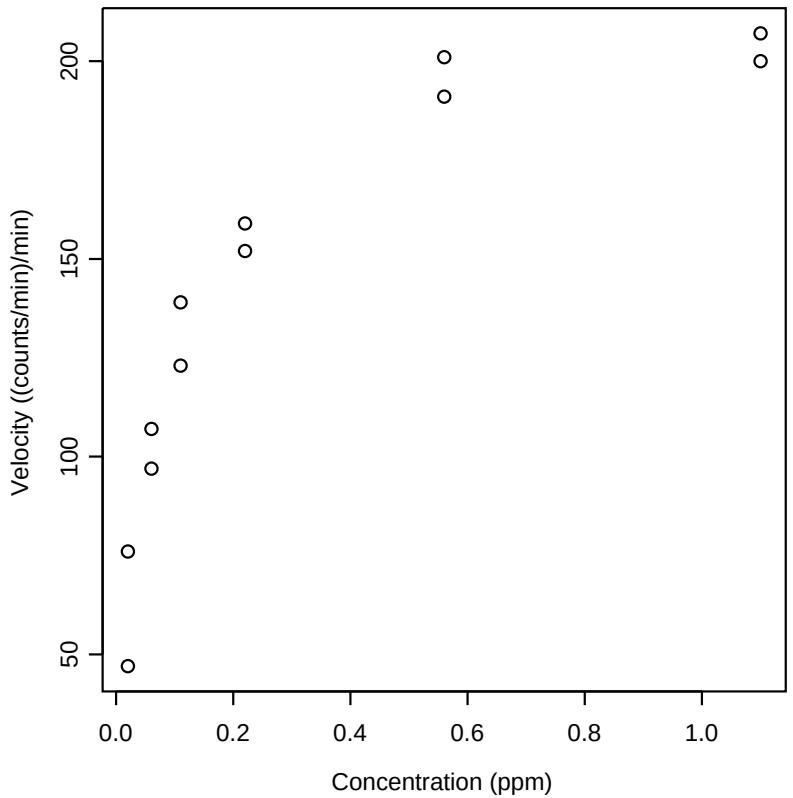
```
> puromycin <- list(
+ conc = c(.02, .02, .06, .06, .11, .11, .22, .22, .56, .56, 1.10, 1.10),
+ vel = c(76, 47, 97, 107, 123, 139, 159, 152, 191, 201, 207, 200))

> puromycin <- as.data.frame(puromycin)
```

Exploramos el conjunto de datos por medio del gráfico:

```
> plot(vel ~ conc, data = puromycin,
+       xlab = "Concentration (ppm)",
+       ylab = "Velocity ((counts/min)/min)")
```

Datos de Puromycin (Treolar, 1974)

Velocidad de reacción versus concentración del substrato para un experimento de Puromycin



Ajuste de los datos de Puromycin

Ajuste usando la función `nls()`:

```
> fm1 <- nls(vel ~ Vm * conc / (K + conc), data = puromycin,
+       start = list(K = 0.08, Vm = 205), trace = TRUE)
3155.004 :     0.08 205.00
1205.662 :     0.06289222 213.02889453
1195.477 :     0.06398775 212.60337573
1195.449 :     0.06410830 212.67543440
1195.449 :     0.06412003 212.68293989
1195.449 :     0.06412116 212.68366582

> fm1
Nonlinear regression model
model:  vel ~ Vm * conc/(K + conc)
data:  puromycin
       K         Vm
0.06412 212.68367
residual sum-of-squares: 1195

Number of iterations to convergence: 5
Achieved convergence tolerance: 4.166e-06
```

Resultados de estimación

Sabemos que la función de log-verosimilitud está dada por

$$\ell(\boldsymbol{\psi}) = -\frac{n}{2}\log 2\pi\sigma^2 - \frac{1}{2\sigma^2}S(\boldsymbol{\theta}), \qquad \boldsymbol{\psi} = (\boldsymbol{\theta}^\top, \sigma^2)^\top$$

donde la suma de cuadrados residual es $S(\boldsymbol{\theta}) = \|\boldsymbol{Y} - \boldsymbol{f}(\boldsymbol{\theta})\|^2$.

Para nuestros datos, obtenemos $S(\widehat{\boldsymbol{\theta}})$, $\ell(\widehat{\boldsymbol{\psi}})$ y $\widehat{\boldsymbol{\theta}}$ usando

```
> deviance(fm1)
[1] 1195.449

> logLik(fm1)
'log Lik.' -44.63548 (df=3)

> coef(fm1)
         K             Vm
0.06412116 212.68366582
```

Resultados de estimación

Detalles sobre la estimación pueden ser obtenidos usando la función `summary()`

```
> summary(fm1)

Formula: vel ~ Vm * conc/(K + conc)

Parameters:
    Estimate Std. Error t value Pr(>|t|)
K   6.412e-02  8.281e-03   7.743 1.57e-05
Vm  2.127e+02  6.947e+00  30.615 3.24e-11

Residual standard error: 10.93 on 10 degrees of freedom

Number of iterations to convergence: 5
Achieved convergence tolerance: 4.166e-06
```

Regiones de verosimilitud

La región de verosimilitud conjunta aproximada con un nivel de confianza $1-\alpha$ para $\boldsymbol{\theta}$ es

$$S(\boldsymbol{\theta}) \leq S(\widehat{\boldsymbol{\theta}})\left\{1+\frac{p}{n-p}F_{1-\alpha}(p,n-p)\right\},$$

en nuestro caso,

$$S(\boldsymbol{\theta}) = \sum_{i=1}^{12}\left(\text{velocidad}_i - \frac{V_m \text{concentracion}_i}{K+\text{concentracion}_i}\right)^2$$

Resultados de estimación

Gráfico del modelo ajustado:

```
> plot(vel ~ conc, data = puromycin, ylim = c(45, 215),
+      xlab = "Concentration (ppm)",
+      ylab = "Velocity ((counts/min)/min)")

> concVal <- with(puromycin, seq(min(conc), max(conc), length.out = 100))
> lines(concVal, predict(fm1, newdata = data.frame(conc = concVal)))
> abline(h = coef(fm1)[2], lty = 8)
```

Gráfico del elipsoide de confianza del 95% para $\boldsymbol{\theta} = (Vm, K)^\top$

```
> fm1.con <- nlsContourRSS(fm1)
100%
RSS contour surface array returned

> plot(fm1.con, nlev = 10)
```

Gráfico de ajuste para los datos de Puromycin

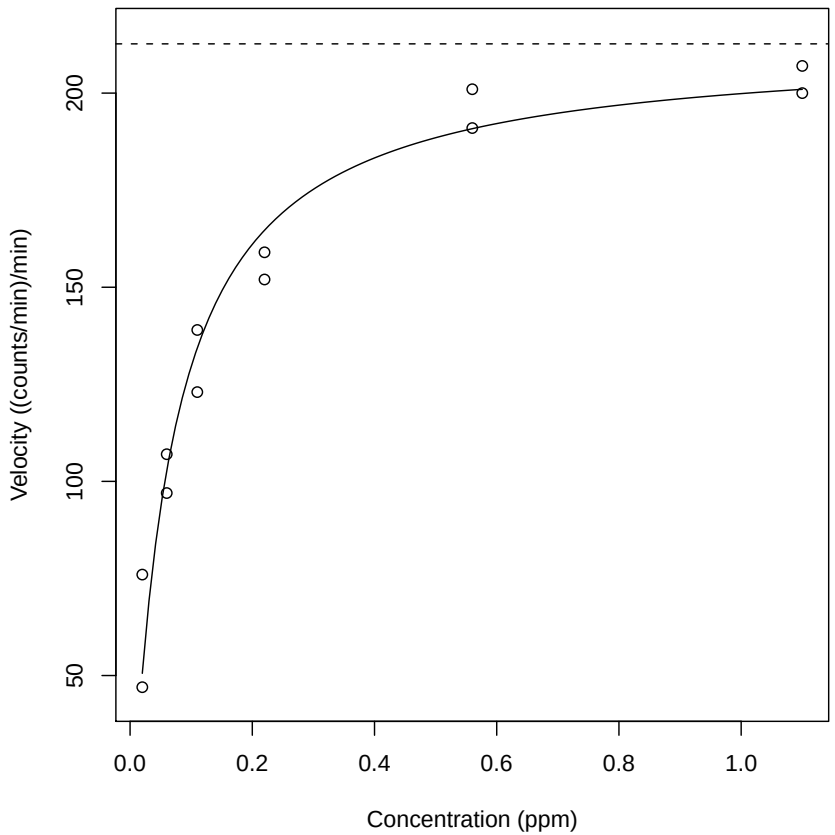


Gráfico de contornos de la suma de cuadrados residual

La región indicada en la línea punteada representa un elipsoide de confianza del 95%.

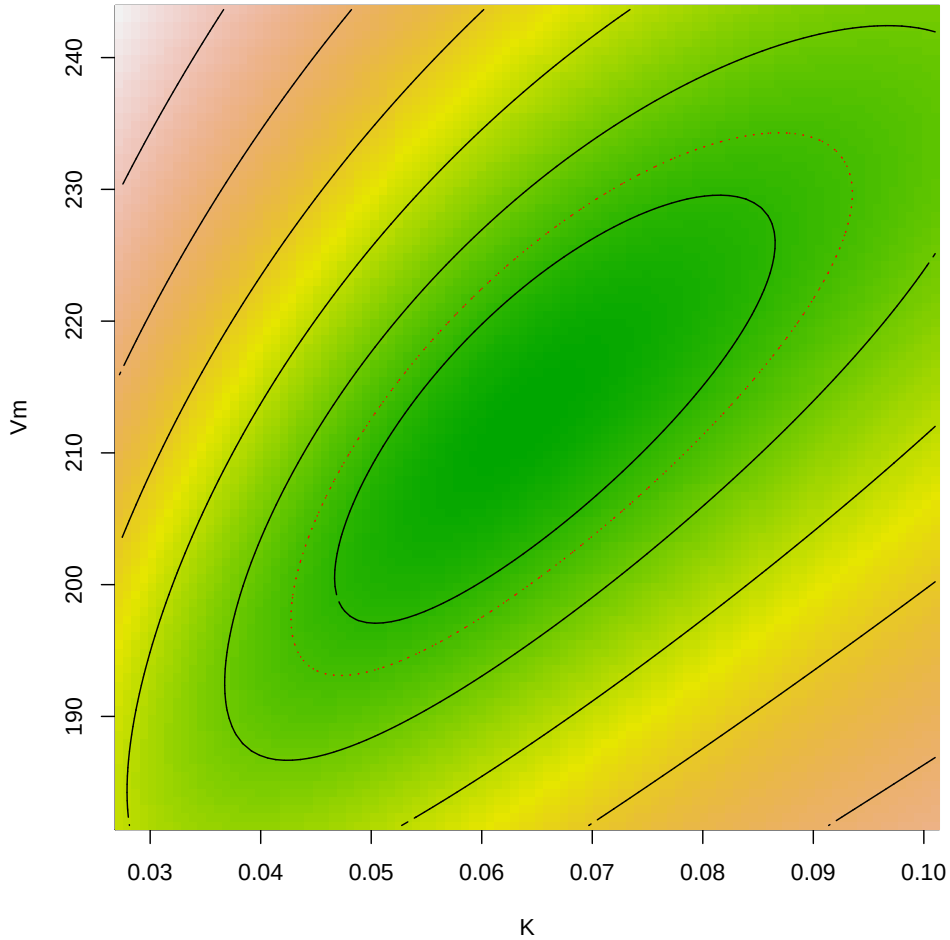
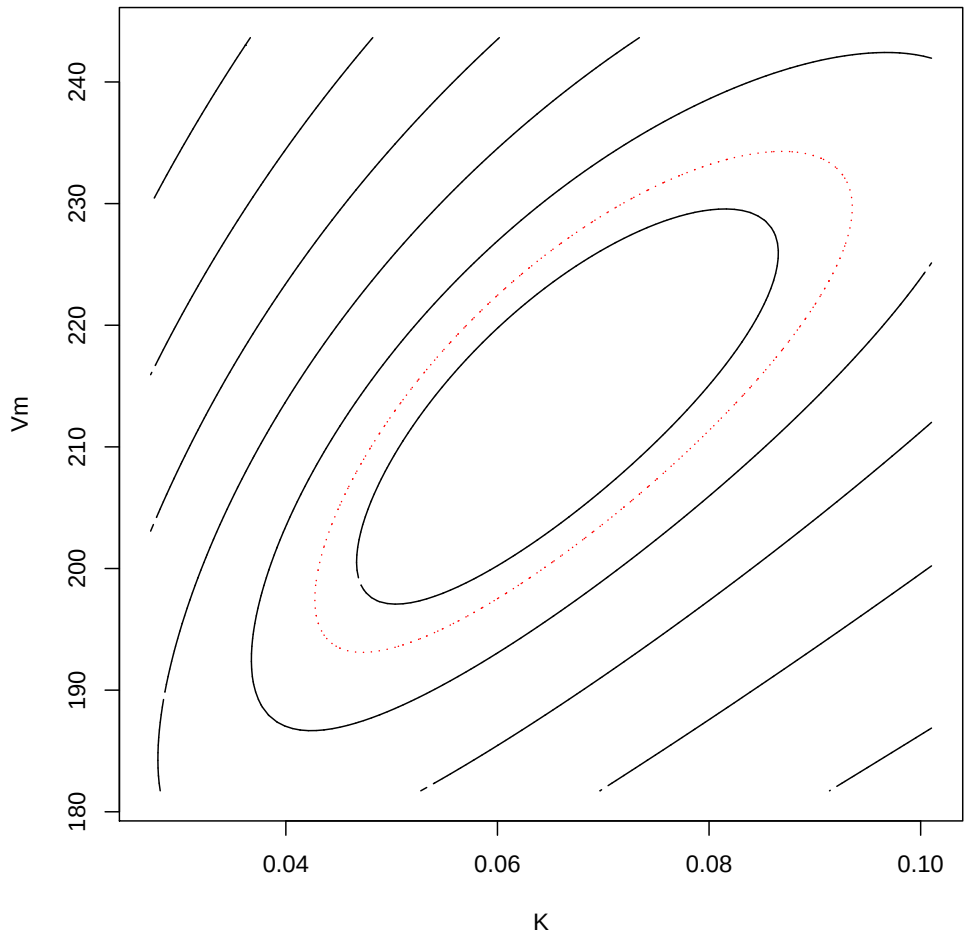


Gráfico de contornos de la suma de cuadrados residual

La región indicada en la línea punteada representa un elipsoide de confianza del 95%.



Self-Starting models

Bates y Watts (1988) describen técnicas para determinar estimaciones iniciales en un modelo de regresión no lineal. Algunas de estas estrategias son:

- Tomar provecho de modelos parcialmente lineales, sólo son necesarios valores iniciales para aquellos parámetros no lineales.
- Escoger estimaciones iniciales que tengan alguna interpretación significativa.
- Refinar estimaciones de algunos parámetros por iterar sobre ellos, mientras que los otros parámetros se mantienen fijados

Modelos transformables

Recuerde que, el modelo Michaelis-Menten,

$$f(x, \boldsymbol{\theta}) = \frac{V_m x}{K + x}, \qquad \boldsymbol{\theta} = (V_m, K)^\top.$$

Es un modelo intrinsecamente lineal, pues

$$\frac{1}{f} = \frac{1}{V_m}\left(1 + \frac{K}{x}\right) = \frac{1}{V_m} + \frac{K}{V_m}\frac{1}{x} = \beta_1 + \beta_2 z.$$

Esta estratégia es implementada en la función `SSmicmen()`.

Función self-start: `SSmicmen`

En particular, la función `SSmicmen()`, realiza los siguientes cálculos:

```
# ajusta un modelo lineal
lin <- lm(1/vel ~ I(1/conc), data = puromycin)
pars <- coef(lin)

# refina estimación inicial via iteración
obj <- nls(y ~ x/(K + x), data = puromycin,
+        start = list(K = abs(pars[2]/pars[1])),
+        algorithm = "plinear")

# retorna 'valores iniciales'
start <- c(obj[2], obj[1])
```

Usando nls con SSmicmen

La función SSmicmen() está documentada como parte de la ayuda de nls. Note que no se requiere que el usuario defina valores iniciales.

```
> fm2 <- nls(vel ~ SSmicmen(conc, Vm, K), data = puromycin, trace = TRUE)
1195.449 :  212.68370749   0.06412123

> fm2
Nonlinear regression model
model:  vel ~ SSmicmen(conc, Vm, K)
 data:  puromycin
     Vm          K
212.68371   0.06412
residual sum-of-squares: 1195

Number of iterations to convergence: 0
Achieved convergence tolerance: 1.917e-06
```

Sin embargo, las funciones Self-starting están disponibles sólo para algunos tipos de funciones no lineales

Modelos condicionalmente lineales

Note que, el modelo Michaelis-Menten es condicionalmente lineal

$$f(x, \boldsymbol{\theta}) = \frac{V_m x}{K + x} = V_m \Big(\frac{x}{K + x} \Big),$$

podemos usar el algoritmo de Golub-Pereyra (1973)

```
> fm3 <- nls(vel ~ conc / (K + conc), data = puromycin,
+     algorithm = "plinear",
+     start = list(K = 0.08), trace = TRUE)
1524.682 :   0.0800 222.2759
1195.471 :   0.06423932 212.75947067
1195.449 :   0.06413257 212.69098858
1195.449 :   0.06412237 212.68444089
1195.449 :   0.06412139 212.68381040
```

Resumen de estimación

Detalles de la estimación en el modelo condicionalmente lineal:

```
> summary(fm3)

Formula: vel ~ conc/(K + conc)

Parameters:
     Estimate Std. Error t value Pr(>|t|)
K    6.412e-02  8.281e-03   7.743 1.57e-05 ***
.lin 2.127e+02  6.947e+00  30.615 3.24e-11 ***

Residual standard error: 10.93 on 10 degrees of freedom

Number of iterations to convergence: 4
Achieved convergence tolerance: 8.037e-06

> deviance(fm3)
[1] 1195.449

> logLik(fm3)
'log Lik.' -44.63548 (df=3)
```

Usando nl2sol desde PORT

```
> fm4 <- nls(vel ~ Vm * conc / (K + conc), data = puromycin,
+     algorithm = "port",
+     start = list(K = 0.08), trace = TRUE)
0:      1577.5021: 0.0800000  205.000
1:      1390.1712: 0.0784593  205.959
2:      600.78114: 0.0632716  213.046
3:      597.73075: 0.0640318  212.629
4:      597.72447: 0.0641126  212.678
5:      597.72441: 0.0641204  212.683
6:      597.72441: 0.0641212  212.684
7:      597.72441: 0.0641213  212.684

> fm4
Nonlinear regression model
model:  vel ~ Vm * conc/(K + conc)
 data:  puromycin
      K         Vm
0.06412 212.68374
residual sum-of-squares: 1195

Algorithm "port", convergence message: relative convergence (4)

> deviance(fm4)
[1] 1195.449

> logLik(fm4)
'log Lik.' -44.63548 (df=3)
```

Supliendo información del gradiente

Pasamos la información del gradiente (y la propia función f) usando, por ejemplo:

```
MMgrad <- function(conc, Vm, K) {
  numer <- Vm * conc
  denom <- K + conc
  mean <- numer / denom
  partialVm <- mean / Vm
  partialK <- -numer / (denom^2)
  attr(mean, "gradient") <- cbind(partialVm, partialK)
  return(mean)
}
```

Resumen de estimación

```
> fm5 <- nls(vel ~ MMgrad(conc, Vm, K), data = puromycin, trace = TRUE,
+     start = list(Vm = 205, K = 0.08))
3155.004 :  205.00   0.08
1205.662 :  213.02889449   0.06289222
1195.477 :  212.60337549   0.06398775
1195.449 :  212.67543430   0.06410830
1195.449 :  212.68294013   0.06412003
1195.449 :  212.68366575   0.06412116

> fm5
Nonlinear regression model
model:  vel ~ MMgrad(conc, Vm, K)
 data:  puromycin
     Vm          K
212.68367   0.06412
residual sum-of-squares: 1195

Number of iterations to convergence: 5
Achieved convergence tolerance: 4.161e-06
```

Datos de Puromycin

Resumen de estimación en el modelo Michaelis-Menten:

$$f(x, \boldsymbol{\theta}) = \frac{V_m x}{K + x}, \qquad \boldsymbol{\theta} = (V_m, K)^\top.$$

Método	$\widehat{V}_m$	$\widehat{K}$	iteraciones	$S(\widehat{\boldsymbol{\theta}})$
Gauss-Newton	212.68367	0.06412	5	1195.449
selfStart	212.68371	0.06412	0	1195.449
Golub-Pereira	212.68381	0.06412	4	1195.449
nl2sol	212.68374	0.06412	7	1195.449
G-N gradiente	212.68367	0.06412	5	1195.449

Se usó (según corresponda) $\boldsymbol{\theta}^{(0)} = (205, 0.08)^\top$.

Funciones Self-starting disponibles

Repertorio de funciones Self-starting disponibles para nls (y extensiones):

SSasymp	Modelo de regresión asintótico.
SSasympOff	Modelo de regresión asintótico con un offset.
SSasympOrig	Modelo de regresión asintótico a través del origen.
SSbiexp	Modelo biexponencial.
SSfol	Modelo de un compartimento de primer orden.
SSfpl	Modelo logístico de cuatro parámetros.
SSgompertz	Modelo de crecimiento de Gompertz.
SSlogis	Modelo logístico.
SSmicmen	Modelo Michaelis-Menten.
SSweibull	Modelo de curva de crecimiento Weibull.
selfStart	Constructor de modelos nolineales Self-starting.

Estas funciones han sido usadan en contextos más generales como en la función nlme para ajustar modelos nolineales con efectos mixtos.

Modelos parcialmente lineales: Método de Golub-Pereyra

A seguir revisamos algunos aspectos sobre la estimación en modelos parcialmente lineales. Mayores detalles en Golub y Pereyra (1973)[2]

Golub y Pereyra (1973) consideran el siguiente modelo no lineal:

$$f(\boldsymbol{x};\boldsymbol{\alpha},\boldsymbol{\beta}) = \sum_{j=1}^{p} \beta_j g_j(\boldsymbol{x};\boldsymbol{\alpha}), \qquad \boldsymbol{\alpha} \in \mathbb{R}^k, \boldsymbol{\beta} \in \mathbb{R}^p.$$

Ellos proponen minimizar la función:

$$S(\boldsymbol{\alpha},\boldsymbol{\beta}) = \|\boldsymbol{Y} - \boldsymbol{f}(\boldsymbol{\alpha},\boldsymbol{\beta})\|^2 = \|\boldsymbol{Y} - \boldsymbol{G}(\boldsymbol{\alpha})\boldsymbol{\beta}\|^2. \tag{1}$$

[2] SIAM J. Numer. Anal. **10**, 413-432

Modelos parcialmente lineales: Método de Golub-Pereyra

- Golub y Pereyra (1973) proponen dos algoritmos para llevar a cabo la minimización de (1).
- Una contribución importante de ese trabajo fue obtener una expresión para la derivada de una inversa generalizada (así como de una matriz de proyección).
- Simplificaremos la exposición asumiendo que $\boldsymbol{G}(\boldsymbol{\alpha})$ tiene rango columna completo (para cualquier $\boldsymbol{\alpha}$).
- A la fecha este algoritmo es uno de los más eficientes para resolver problemas separables (en la media).

Modelos parcialmente lineales: Método de Golub-Pereyra

Suponiendo que $\boldsymbol{\alpha}$ es fijado, podemos optimizar (1) con relación a $\boldsymbol{\beta}$, obteniendo

$$\widehat{\boldsymbol{\beta}}(\boldsymbol{\alpha}) = (\boldsymbol{G}^\top \boldsymbol{G})^{-1} \boldsymbol{G}^\top \boldsymbol{Y}, \qquad \boldsymbol{G} = \boldsymbol{G}(\boldsymbol{\alpha}).$$

Así, perfilando $S(\boldsymbol{\alpha}, \boldsymbol{\beta})$ resulta,

$$\begin{aligned} S_*(\boldsymbol{\alpha}) &= S(\boldsymbol{\alpha}, \widehat{\boldsymbol{\beta}}(\boldsymbol{\alpha})) = \|\boldsymbol{Y} - \boldsymbol{G}(\boldsymbol{\alpha})\widehat{\boldsymbol{\beta}}(\boldsymbol{\alpha})\|^2 \\ &= \|\boldsymbol{Y} - \boldsymbol{H}(\boldsymbol{\alpha})\boldsymbol{Y}\|^2 = \|\{\boldsymbol{I} - \boldsymbol{H}(\boldsymbol{\alpha})\}\boldsymbol{Y}\|^2, \end{aligned}$$

donde

$$\boldsymbol{H}(\boldsymbol{\alpha}) = \boldsymbol{G}(\boldsymbol{G}^\top \boldsymbol{G})^{-1} \boldsymbol{G}^\top, \qquad \boldsymbol{G} = \boldsymbol{G}(\boldsymbol{\alpha}).$$

Como $\boldsymbol{H}(\boldsymbol{\alpha})$ es matriz de proyección, tenemos que

$$S_*(\boldsymbol{\alpha}) = \boldsymbol{Y}^\top \{\boldsymbol{I} - \boldsymbol{H}(\boldsymbol{\alpha})\}\boldsymbol{Y}.$$

Modelos parcialmente lineales: Método de Golub-Pereyra

Suponiendo que $\boldsymbol{\alpha}$ es fijado, podemos optimizar (1) con relación a $\boldsymbol{\beta}$, obteniendo

$$\widehat{\boldsymbol{\beta}}(\boldsymbol{\alpha}) = (\boldsymbol{G}^\top \boldsymbol{G})^{-1} \boldsymbol{G}^\top \boldsymbol{Y}, \qquad \boldsymbol{G} = \boldsymbol{G}(\boldsymbol{\alpha}).$$

Así, perfilando $S(\boldsymbol{\alpha}, \boldsymbol{\beta})$ resulta,

$$\begin{aligned} S_*(\boldsymbol{\alpha}) &= S(\boldsymbol{\alpha}, \widehat{\boldsymbol{\beta}}(\boldsymbol{\alpha})) = \|\boldsymbol{Y} - \boldsymbol{G}(\boldsymbol{\alpha})\widehat{\boldsymbol{\beta}}(\boldsymbol{\alpha})\|^2 \\ &= \|\boldsymbol{Y} - \boldsymbol{H}(\boldsymbol{\alpha})\boldsymbol{Y}\|^2 = \|\{\boldsymbol{I} - \boldsymbol{H}(\boldsymbol{\alpha})\}\boldsymbol{Y}\|^2, \end{aligned}$$

donde

$$\boldsymbol{H}(\boldsymbol{\alpha}) = \boldsymbol{G}(\boldsymbol{G}^\top \boldsymbol{G})^{-1} \boldsymbol{G}^\top, \qquad \boldsymbol{G} = \boldsymbol{G}(\boldsymbol{\alpha}).$$

Como $\boldsymbol{H}(\boldsymbol{\alpha})$ es matriz de proyección, tenemos que

$$S_*(\boldsymbol{\alpha}) = \boldsymbol{Y}^\top \{\boldsymbol{I} - \boldsymbol{H}(\boldsymbol{\alpha})\}\boldsymbol{Y}.$$

Modelos parcialmente lineales: Método de Golub-Pereyra

Suponiendo que $\boldsymbol{\alpha}$ es fijado, podemos optimizar (1) con relación a $\boldsymbol{\beta}$, obteniendo

$$\widehat{\boldsymbol{\beta}}(\boldsymbol{\alpha}) = (\boldsymbol{G}^\top \boldsymbol{G})^{-1} \boldsymbol{G}^\top \boldsymbol{Y}, \qquad \boldsymbol{G} = \boldsymbol{G}(\boldsymbol{\alpha}).$$

Así, perfilando $S(\boldsymbol{\alpha}, \boldsymbol{\beta})$ resulta,

$$\begin{aligned} S_*(\boldsymbol{\alpha}) &= S(\boldsymbol{\alpha}, \widehat{\boldsymbol{\beta}}(\boldsymbol{\alpha})) = \|\boldsymbol{Y} - \boldsymbol{G}(\boldsymbol{\alpha})\widehat{\boldsymbol{\beta}}(\boldsymbol{\alpha})\|^2 \\ &= \|\boldsymbol{Y} - \boldsymbol{H}(\boldsymbol{\alpha})\boldsymbol{Y}\|^2 = \|\{\boldsymbol{I} - \boldsymbol{H}(\boldsymbol{\alpha})\}\boldsymbol{Y}\|^2, \end{aligned}$$

donde

$$\boldsymbol{H}(\boldsymbol{\alpha}) = \boldsymbol{G}(\boldsymbol{G}^\top \boldsymbol{G})^{-1} \boldsymbol{G}^\top, \qquad \boldsymbol{G} = \boldsymbol{G}(\boldsymbol{\alpha}).$$

Como $\boldsymbol{H}(\boldsymbol{\alpha})$ es matriz de proyección, tenemos que

$$S_*(\boldsymbol{\alpha}) = \boldsymbol{Y}^\top \{\boldsymbol{I} - \boldsymbol{H}(\boldsymbol{\alpha})\}\boldsymbol{Y}.$$

Modelos parcialmente lineales: Método de Golub-Pereyra

Es decir, para implementar un algoritmo Gauss-Newton, asociado a la minimización de $S_*(\boldsymbol{\alpha})$, debemos obtener

$$\frac{\partial S_*(\boldsymbol{\alpha})}{\partial \boldsymbol{\alpha}} \qquad \text{o bien} \qquad \frac{\partial \boldsymbol{f}_*(\boldsymbol{\alpha})}{\partial \boldsymbol{\alpha}^\top},$$

donde $\boldsymbol{f}_*(\boldsymbol{\alpha}) = \boldsymbol{G}(\boldsymbol{\alpha})\widehat{\boldsymbol{\beta}}(\boldsymbol{\alpha}) = \boldsymbol{H}(\boldsymbol{\alpha})\boldsymbol{Y}$. De este modo, debemos calcular

$$\frac{\partial \boldsymbol{H}(\boldsymbol{\alpha})}{\partial \boldsymbol{\alpha}^\top},$$

lo que no es una tarea trivial.

En efecto, el algoritmo 2 propuesto por Golub y Pereyra (1973) adopta la forma:

$$\boldsymbol{\alpha}^{(k+1)} = \boldsymbol{\alpha}^{(k)} - \left\{ \frac{\partial \boldsymbol{f}_*(\boldsymbol{\alpha})}{\partial \boldsymbol{\alpha}^\top} \right\}^{-1} \boldsymbol{f}_*(\boldsymbol{\alpha}) \Big|_{\alpha=\alpha^{(k)}}.$$

Modelos parcialmente lineales: Método de Golub-Pereyra

Para implementar el método, considere $\boldsymbol{G}^- = (\boldsymbol{G}^\top \boldsymbol{G})^{-1}\boldsymbol{G}^\top$ una inversa generalizada de $\boldsymbol{G}$. Así

$$\mathrm{d}\,\boldsymbol{H} = \mathrm{d}\,\boldsymbol{G}\boldsymbol{G}^- = (\mathrm{d}\,\boldsymbol{G})\boldsymbol{G}^- + \boldsymbol{G}\,\mathrm{d}\,\boldsymbol{G}^-$$

Aprovechando la estructura de $\boldsymbol{G}^-$, tenemos[3]

$$\begin{aligned}\mathrm{d}\,\boldsymbol{H} &= (\mathrm{d}\,\boldsymbol{G})\boldsymbol{G}^- - \{(\mathrm{d}\,\boldsymbol{G})\boldsymbol{G}^-\}^\top \boldsymbol{G}\boldsymbol{G}^- - (\boldsymbol{G}\boldsymbol{G}^-)^\top(\mathrm{d}\,\boldsymbol{G})\boldsymbol{G}^- + \{(\mathrm{d}\,\boldsymbol{G})\boldsymbol{G}^-\}^\top \\ &= 2(\boldsymbol{I} - \boldsymbol{H})\boldsymbol{G}^-\,\mathrm{d}\,\boldsymbol{G}.\end{aligned}$$

Luego, se debe vectorizar para obtener $\mathrm{D}\,\mathrm{vec}\,\boldsymbol{G} = \partial\,\mathrm{vec}\,\boldsymbol{G}/\partial\boldsymbol{\alpha}^\top$.

[3] Golub y Pereyra solo usan las propiedades de una inversa generalizada.