Orden del Tema

Introducción: Algoritmo de Newton

- El método de Newton es una algoritmo iterativo que permite obtener el óptimo $\boldsymbol{x}^*$ de una función 2 veces continuamente diferenciable $f : \mathbb{R}^n \rightarrow \mathbb{R}$.
- Es un algoritmo de optimización sin restricciones
- A partir de un punto inicial $\boldsymbol{x}_0$ se genera una secuencia $\{\boldsymbol{x}_k\}$

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - (\nabla^2 f_k)^{-1} \nabla f_k$$

donde $\nabla^2 f_k = \nabla^2 f(\boldsymbol{x}_k)$ es el Hessiano y $\nabla f_k = \nabla f(\boldsymbol{x}_k)$ es el gradiente

Introducción

- Para encontrar el nuevo punto $\boldsymbol{x}_{k+1}$ a partir de $\boldsymbol{x}_k$, se define $\boldsymbol{d} = \boldsymbol{x} - \boldsymbol{x}_k$ y se usa una aproximación de Taylor de segundo orden de la función

$$m_k(\boldsymbol{d}) = f_k + \nabla^T f_k \boldsymbol{d} + \frac{1}{2}\boldsymbol{d}^T \nabla^2 f_k \boldsymbol{d}$$

- El tamaño de paso p se obtiene calculando el gradiente, igualando a cero y resolviendo para p

$$\begin{aligned} \nabla f_k + \nabla^2 f_k \boldsymbol{d} &= 0 \\ \boldsymbol{d} &= -(\nabla^2 f_k)^{-1} \nabla f_k \\ \boldsymbol{x}_{k+1} &= \boldsymbol{x}_k - \alpha_k (\nabla^2 f_k)^{-1} \nabla f_k, \ \alpha_k = 1 \end{aligned}$$

Introducción

- Para encontrar el nuevo punto $\boldsymbol{x}_{k+1}$ a partir de $\boldsymbol{x}_k$, se define $\boldsymbol{d} = \boldsymbol{x} - \boldsymbol{x}_k$ y se usa una aproximación de Taylor de segundo orden de la función

$$m_k(\boldsymbol{d}) = f_k + \nabla^T f_k \boldsymbol{d} + \frac{1}{2}\boldsymbol{d}^T \nabla^2 f_k \boldsymbol{d}$$

- El tamaño de paso $\boldsymbol{p}$ se obtiene calculando el gradiente, igualando a cero y resolviendo para $\boldsymbol{p}$

$$\begin{aligned} \nabla f_k + \nabla^2 f_k \boldsymbol{d} &= 0 \\ \boldsymbol{d} &= -(\nabla^2 f_k)^{-1} \nabla f_k \\ \boldsymbol{x}_{k+1} &= \boldsymbol{x}_k - \alpha_k (\nabla^2 f_k)^{-1} \nabla f_k, \;\; \alpha_k = 1 \end{aligned}$$

Introducción

- Una de la problemáticas del método de Newton es que requiere calcular el Hessiano $\nabla^2 f_k$ el cual pueder ser muy costoso
- Más aún, requiere del cálculo de la inversa del Hessiano

$$\boldsymbol{d}_k = -(\nabla^2 f_k)^{-1} \nabla f_k$$

- Otro problema es que no se puede garantizar que $\boldsymbol{d}_k$ sea una dirección de descenso, i.e, puede suceder que $\boldsymbol{d}_k^T \nabla f_k = -\nabla^T f_k (\nabla^2 f_k)^{-1} \nabla f_k > 0$ en alguna o varias iteraciones ..., por ejemplo, si $\nabla^2 f_k$ es definida negativa.

Introducción

- Una alternativa de solución, cuando $\boldsymbol{d}_k$ no es una dirección de descenso, es modificar el Hessiano añadiendo una matrix $\mathbf{E}_k$ de modo que la nueva matriz $\mathbf{B}_k = \nabla^2 f_k + \mathbf{E}_k$ sea definida positiva. En ejemplo trivial, sería hacer $\mathbf{E}_k$ un múltiplo de la matriz idéntica $\mathbf{E}_k = \tau\mathbf{I}$. La nueva dirección de descenso sería

$$\boldsymbol{d} = -\mathbf{B}_k^{-1}\nabla f_k$$

- La alternativa anterior garantiza que la dirección d_k sea de descenso, sin embargo, requiere del cálculo de Hessiano. Además necesita de un algoritmo que garantice que la nueva matriz $\mathbf{B}_k$ sea definida positiva (p.e. Cholesky).
- Existe alguna otra alternativa??

Introducción

- Una alternativa de solución, cuando $\boldsymbol{d}_k$ no es una dirección de descenso, es modificar el Hessiano añadiendo una matrix $\mathbf{E}_k$ de modo que la nueva matriz $\mathbf{B}_k = \nabla^2 f_k + \mathbf{E}_k$ sea definida positiva. En ejemplo trivial, sería hacer $\mathbf{E}_k$ un múltiplo de la matriz idéntica $\mathbf{E}_k = \tau\mathbf{I}$. La nueva dirección de descenso sería

$$\boldsymbol{d} = -\mathbf{B}_k^{-1}\nabla f_k$$

- La alternativa anterior garantiza que la dirección $\boldsymbol{d}_k$ sea de descenso, sin embargo, requiere del cálculo de Hessiano. Además necesita de un algoritmo que garantice que la nueva matriz $\mathbf{B}_k$ sea definida positiva (p.e. Cholesky).
- Existe alguna otra alternativa??

Introducción

- Una alternativa de solución, cuando $\boldsymbol{d}_k$ no es una dirección de descenso, es modificar el Hessiano añadiendo una matrix $\mathbf{E}_k$ de modo que la nueva matriz $\mathbf{B}_k = \nabla^2 f_k + \mathbf{E}_k$ sea definida positiva. En ejemplo trivial, sería hacer $\mathbf{E}_k$ un múltiplo de la matriz idéntica $\mathbf{E}_k = \tau \mathbf{I}$. La nueva dirección de descenso sería

$$\boldsymbol{d} = -\mathbf{B}_k^{-1} \nabla f_k$$

- La alternativa anterior garantiza que la dirección $\boldsymbol{d}_k$ sea de descenso, sin embargo, requiere del cálculo de Hessiano. Además necesita de un algoritmo que garantice que la nueva matriz $\mathbf{B}_k$ sea definida positiva (p.e. Cholesky).
- Existe alguna otra alternativa??

Introducción

- Los métodos cuasi-newton construyen un modelo que se basa en medir los cambios del gradiente.
- Solo requieren del cálculo del gradiente, similar al algoritmo de máximo descenso.
- Su comportamiento es superior al algoritmo de máximo descenso.
- En lugar de calcular el Hessiano en cada iteración se propone un método que permite calcular una secuencia $\mathbf{B}_k$ usando la curvatura medida en el paso actual.

Introducción

- El primer algoritmo Cuasi-Newton fue propuesto por Davidson en 1959, luego fue estudiado por Fletcher and Powell, y es conocido como el *Algoritmo DFP* en honor a sus trabajos.
- Una mejora del algoritmo *DFP* se obtuvo independientemente por Broyden, Fletcher, Goldfarb y Shanno por eso es conocido como el *Algoritmo BFGS*

Orden del Tema

Recordatorio

En el problema cuadratico

$$\min f(\boldsymbol{x}) \quad = \quad \frac{1}{2}\boldsymbol{x}^T \mathbf{Q}\boldsymbol{x} - \boldsymbol{b}^T \boldsymbol{x}$$

donde $\mathbf{Q}^T = \mathbf{Q}$, al usar la actualización de descenso

$$\boldsymbol{x}_{k+1} \quad = \quad \boldsymbol{x}_k + \alpha \boldsymbol{d}_k$$

Por otro lado

$$\begin{aligned} \boldsymbol{g}_{k+1} &= \mathbf{Q}\boldsymbol{x}_{k+1} - \boldsymbol{b} \\ \boldsymbol{g}_k &= \mathbf{Q}\boldsymbol{x}_k - \boldsymbol{b} \end{aligned}$$

se llega a la conclusión

$$\mathbf{Q}(\boldsymbol{x}_{k+1} - \boldsymbol{x}_k) \quad = \quad \boldsymbol{g}_{k+1} - \boldsymbol{g}_k$$

Recordatorio

Definiendo $\boldsymbol{y}_k = \boldsymbol{g}_{k+1} - \boldsymbol{g}_k$ y $\boldsymbol{s}_k = \boldsymbol{x}_{k+1} - \boldsymbol{x}_k$ se tiene la relación

$$\mathbf{Q}\boldsymbol{s}_k = \boldsymbol{y}_k$$

que se conoce como **ecuación de la secante.** Podemos escribir tambien

$$\mathbf{Q}^{-1}\boldsymbol{y}_k = \boldsymbol{s}_k$$

Idea General

Para el problema general

$$\min f(\boldsymbol{x})$$

La idea de lo metodos Cuasi-Newton es usar una aproximación del metodo de Newton

$$\begin{aligned}\boldsymbol{x}_{k+1} &= \boldsymbol{x}_k - \mathbf{B}_k^{-1}\boldsymbol{g}_k \\ \boldsymbol{x}_{k+1} &= \boldsymbol{x}_k - \mathbf{H}_k\boldsymbol{g}_k\end{aligned}$$

donde $\mathbf{B}_k \approx \nabla^2 f_k$ o $\mathbf{H}_k \approx \nabla^2 f_k^{-1}$.

Idea General

Que se obtienen de minimizar alguno de los modelos

$$
\begin{aligned}
m_{k+1}(\boldsymbol{d}) &= f_{k+1} + \nabla^T f_{k+1}\boldsymbol{d} + \frac{1}{2}\boldsymbol{d}^T \mathbf{B}_{k+1}\boldsymbol{d} \\
m_{k+1}(\boldsymbol{d}) &= f_{k+1} + \nabla^T f_{k+1}\boldsymbol{d} + \frac{1}{2}\boldsymbol{d}^T \mathbf{H}_{k+1}^{-1}\boldsymbol{d}
\end{aligned}
$$

donde $\mathbf{B}_k \approx \nabla^2 f_k$ o $\mathbf{H}_k \approx \nabla^2 f_k^{-1}$.

Idea General

Para determinar las matrices anteriores se imponen algunas condiciones a las matrices $\mathbf{B}_{k+1}$ o $\mathbf{H}_{k+1}$.
Por ejemplo se pide que sean simétricas y satisfagan la ecuación de la secante, ie

$$\begin{aligned} \mathbf{B}_{k+1}\boldsymbol{s}_k &= \boldsymbol{y}_k \\ \mathbf{H}_{k+1}\boldsymbol{y}_k &= \boldsymbol{s}_k \end{aligned}$$

Symmetric rank 1

A partir de ahora vamos a aproximar la inversa del Hessiano, ie, $\mathbf{H}_k \approx \nabla^2 f_k^{-1}$.
La idea es construir una secuencia $\mathbf{H}_0, \mathbf{H}_1, \cdots$ que satisfaga,

$$\mathbf{H}_{k+1} \boldsymbol{y}_i = \boldsymbol{s}_i, \; i = 0, 1, \cdots k$$

Symmetric rank 1

La corrección de rango 1 se puede escribir como sigue

$$\mathbf{H}_{k+1} = \mathbf{H}_k + \alpha \boldsymbol{x}\boldsymbol{x}^T$$

Symmetric rank 1

Luego

$$\begin{aligned} \mathbf{H}_{k+1}\boldsymbol{y}_k &= \mathbf{H}_k\boldsymbol{y}_k + \alpha\boldsymbol{x}\boldsymbol{x}^T\boldsymbol{y}_k \\ \alpha\boldsymbol{x}(\boldsymbol{x}^T\boldsymbol{y}_k) &= \boldsymbol{s}_k - \mathbf{H}_k\boldsymbol{y}_k \\ \boldsymbol{x} &= \frac{\boldsymbol{s}_k - \mathbf{H}_k\boldsymbol{y}_k}{\alpha\boldsymbol{x}^T\boldsymbol{y}_k} \end{aligned}$$

y por tanto

$$\mathbf{H}_{k+1} = \mathbf{H}_k + \frac{(\boldsymbol{s}_k - \mathbf{H}_k\boldsymbol{y}_k)(\boldsymbol{s}_k - \mathbf{H}_k\boldsymbol{y}_k)^T}{\alpha(\boldsymbol{x}^T\boldsymbol{y}_k)^2}$$

cuanto es $\alpha(\boldsymbol{x}^T\boldsymbol{y}_k)^2$?

Symmetric rank 1

De la igualdad $\alpha \boldsymbol{x}(\boldsymbol{x}^T \boldsymbol{y}_k) = \boldsymbol{s}_k - \mathbf{H}_k \boldsymbol{y}_k$ se tiene

$$\begin{aligned}
\alpha \boldsymbol{x}(\boldsymbol{x}^T \boldsymbol{y}_k) &= \boldsymbol{s}_k - \mathbf{H}_k \boldsymbol{y}_k \\
\alpha \boldsymbol{y}_k^T \boldsymbol{x}(\boldsymbol{x}^T \boldsymbol{y}_k) &= \boldsymbol{y}_k^T (\boldsymbol{s}_k - \mathbf{H}_k \boldsymbol{y}_k) \\
\alpha (\boldsymbol{x}^T \boldsymbol{y}_k)^2 &= \boldsymbol{y}_k^T (\boldsymbol{s}_k - \mathbf{H}_k \boldsymbol{y}_k)
\end{aligned}$$

Por lo que finalmente se tiene

$$\mathbf{H}_{k+1}^{SR1} = \mathbf{H}_k + \frac{(\boldsymbol{s}_k - \mathbf{H}_k \boldsymbol{y}_k)(\boldsymbol{s}_k - \mathbf{H}_k \boldsymbol{y}_k)^T}{\boldsymbol{y}_k^T (\boldsymbol{s}_k - \mathbf{H}_k \boldsymbol{y}_k)}$$

Orden del Tema

Symmetric rank 2

La corrección de rango 2 se puede escribir como sigue

$$\mathbf{H}_{k+1} = \mathbf{H}_k + \alpha \boldsymbol{x}\boldsymbol{x}^T + \beta \boldsymbol{y}\boldsymbol{y}^T$$

satisfaciendo

$$\mathbf{H}_{k+1}\boldsymbol{y}_k = \boldsymbol{s}_k$$

Symmetric rank 2

Luego

$$\begin{aligned}
\mathbf{H}_{k+1}\boldsymbol{y}_k &= \mathbf{H}_k\boldsymbol{y}_k + \alpha\boldsymbol{x}\boldsymbol{x}^T\boldsymbol{y}_k + \beta\boldsymbol{y}\boldsymbol{y}^T\boldsymbol{y}_k \\
\alpha\boldsymbol{x}(\boldsymbol{x}^T\boldsymbol{y}_k) + \beta\boldsymbol{y}(\boldsymbol{y}^T\boldsymbol{y}_k) &= \boldsymbol{s}_k - \mathbf{H}_k\boldsymbol{y}_k \\
\alpha(\boldsymbol{x}^T\boldsymbol{y}_k)\boldsymbol{x} + \beta(\boldsymbol{y}^T\boldsymbol{y}_k)\boldsymbol{y} &= \boldsymbol{s}_k - \mathbf{H}_k\boldsymbol{y}_k
\end{aligned}$$

La ecuación anterior tiene infinitas soluciones, por lo que podemos tomar o podemos definir convenientemente

$$\begin{aligned}
\boldsymbol{x} &= \boldsymbol{s}_k \\
\boldsymbol{y} &= \mathbf{H}_k\boldsymbol{y}_k
\end{aligned}$$

Symmetric rank 2

$$[\alpha(\boldsymbol{x}^T\boldsymbol{y}_k)]\boldsymbol{s}_k + [\beta(\boldsymbol{y}^T\boldsymbol{y}_k)]\mathbf{H}_k\boldsymbol{y}_k = \boldsymbol{s}_k - \mathbf{H}_k\boldsymbol{y}_k$$

Luego se toma

$$\begin{aligned} \alpha(\boldsymbol{x}^T\boldsymbol{y}_k) &= 1 \\ \beta(\boldsymbol{y}^T\boldsymbol{y}_k) &= -1 \end{aligned}$$

es decir

$$\begin{aligned} \alpha &= \frac{1}{\boldsymbol{x}^T\boldsymbol{y}_k} = \frac{1}{\boldsymbol{s}_k^T\boldsymbol{y}_k} \\ \beta &= -\frac{1}{\boldsymbol{y}^T\boldsymbol{y}_k} = -\frac{1}{\boldsymbol{y}_k^T\mathbf{H}_k\boldsymbol{y}_k} \end{aligned}$$

Symmetric rank 2

Finalmente se tiene

$$\mathbf{H}_{k+1}^{DFP} \quad = \quad \mathbf{H}_k + \frac{\boldsymbol{s}_k \boldsymbol{s}_k^T}{\boldsymbol{s}_k^T \boldsymbol{y}_k} - \frac{\mathbf{H}_k \boldsymbol{y}_k \boldsymbol{y}_k^T \mathbf{H}_k}{\boldsymbol{y}_k^T \mathbf{H}_k \boldsymbol{y}_k}$$

Algoritmo DFP: aproximando la inversa del Hessiano

Algorithm 1 DFP

Require: $\boldsymbol{x}_0$ y $\mathbf{H}_0$

Ensure: $\boldsymbol{x}^*$

1: $k = 0$
2: **while** $\|\nabla f_k\| \neq 0$ (No conveja) **do**
3: $\boldsymbol{d}_k = -\mathbf{H}_k \nabla f_k$
4: Calcular α_k usando búsqueda en línea.
5: $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \alpha_k \boldsymbol{d}_k$
6: Cacular ∇f_{k+1}, $\boldsymbol{y}_k$, $\boldsymbol{s}_k$ y actualizar $\mathbf{H}_{k+1}$

$$\mathbf{H}_{k+1} = \mathbf{H}_k - \frac{\mathbf{H}_k \boldsymbol{y}_k \boldsymbol{y}_k^T \mathbf{H}_k}{\boldsymbol{y}_k^T \mathbf{H}_k \boldsymbol{y}_k} + \frac{\boldsymbol{s}_k \boldsymbol{s}_k^T}{\boldsymbol{y}_k^T \boldsymbol{s}_k}$$

7: $k = k + 1$
8: **end while**

Orden del Tema

DFP usando aproximación del Hessiano

Se tiene

$$\mathbf{H}_{k+1} = \mathbf{H}_k - \frac{\mathbf{H}_k \boldsymbol{y}_k \boldsymbol{y}_k^T \mathbf{H}_k}{\boldsymbol{y}_k^T \mathbf{H}_k \boldsymbol{y}_k} + \frac{\boldsymbol{s}_k \boldsymbol{s}_k^T}{\boldsymbol{s}_k^T \boldsymbol{y}_k}$$

Definamos $\mathbf{U}$, $\mathbf{V}$, a y b y eliminemos los indices por comodidad

$$\begin{aligned} \mathbf{U} &= [a\boldsymbol{s}, b\mathbf{H}\boldsymbol{y}] \\ \mathbf{V} &= \begin{bmatrix} a\boldsymbol{s}^T \\ -b\boldsymbol{y}^T\mathbf{H} \end{bmatrix} \\ a^2 &= \frac{1}{\boldsymbol{y}^T\boldsymbol{s}} \\ b^2 &= \frac{1}{\boldsymbol{y}^T\mathbf{H}\boldsymbol{y}} \end{aligned}$$

DFP usando aproximación del Hessiano

Nota que

$$
\begin{aligned}
\mathbf{UV} &= [a\boldsymbol{s}, b\mathbf{H}\boldsymbol{y}] \begin{bmatrix} a\boldsymbol{s}^T \\ -b\boldsymbol{y}^T\mathbf{H} \end{bmatrix} \\
&= a^2\boldsymbol{s}\boldsymbol{s}^T - b^2\mathbf{H}\boldsymbol{y}\boldsymbol{y}^T\mathbf{H} \\
&= \frac{\boldsymbol{s}\boldsymbol{s}^T}{\boldsymbol{y}^T\boldsymbol{s}} - \frac{\mathbf{H}\boldsymbol{y}\boldsymbol{y}^T\mathbf{H}}{\boldsymbol{y}^T\mathbf{H}\boldsymbol{y}}
\end{aligned}
$$

DFP usando aproximación del Hessiano

Luego podemos escribir

$$\begin{aligned} \mathbf{H}_{k+1} &= \mathbf{H}_k - \frac{\mathbf{H}_k \boldsymbol{y}_k \boldsymbol{y}_k^T \mathbf{H}_k}{y_k^T \mathbf{H}_k \boldsymbol{y}_k} + \frac{\boldsymbol{s}_k \boldsymbol{s}_k^T}{\boldsymbol{y}_k^T \boldsymbol{s}_k} \\ &= \mathbf{H}_k + \mathbf{U}_k \mathbf{V}_k \end{aligned}$$

Apliquemos la formula de Sherman-Morrison-Woodbury

$$(\mathbf{A} + \mathbf{U}\mathbf{V})^{-1} = \mathbf{A}^{-1} - \mathbf{A}^{-1}\mathbf{U}\left(\mathbf{I} + \mathbf{V}\mathbf{A}^{-1}\mathbf{U}\right)^{-1}\mathbf{V}\mathbf{A}^{-1}$$

DFP usando aproximación del Hessiano

Como $\mathbf{B} = \mathbf{H}^{-1}$, ie, $\mathbf{BH} = \mathbf{I}$

$$\begin{aligned}
(\mathbf{H} + \mathbf{UV})^{-1} &= \mathbf{H}^{-1} - \mathbf{H}^{-1}\mathbf{U}\left(I + \mathbf{VH}^{-1}\mathbf{U}\right)^{-1}\mathbf{VH}^{-1} \\
&= \mathbf{B} - \mathbf{BU}\left(\mathbf{I} + \mathbf{VBU}\right)^{-1}\mathbf{VB}
\end{aligned}$$

Primero calculemos $\mathbf{VBU}$ y luego $(\mathbf{I} + \mathbf{VBU})^{-1}$

DFP usando aproximación del Hessiano

$$
\begin{aligned}
\mathbf{VBU} &= \begin{bmatrix} a\boldsymbol{s}^T \\ -b\boldsymbol{y}^T\mathbf{H} \end{bmatrix} \mathbf{B}[a\boldsymbol{s}, b\mathbf{H}\boldsymbol{y}] \\
&= \begin{bmatrix} a\boldsymbol{s}^T \\ -b\boldsymbol{y}^T\mathbf{H} \end{bmatrix} [a\mathbf{B}\boldsymbol{s}, b\mathbf{B}\mathbf{H}\boldsymbol{y}] \\
&= \begin{bmatrix} a^2\boldsymbol{s}^T\mathbf{B}\boldsymbol{s} & ab\boldsymbol{s}^T\mathbf{B}\mathbf{H}\boldsymbol{y} \\ -ab\boldsymbol{y}^T\mathbf{H}\mathbf{B}\boldsymbol{s} & -b^2\boldsymbol{y}^T\mathbf{H}\mathbf{B}\mathbf{H}\boldsymbol{y} \end{bmatrix} \\
&= \begin{bmatrix} a^2\boldsymbol{s}^T\mathbf{B}\boldsymbol{s} & ab\boldsymbol{s}^T\boldsymbol{y} \\ -ab\boldsymbol{y}^T\boldsymbol{s} & -b^2\boldsymbol{y}^T\mathbf{H}\boldsymbol{y} \end{bmatrix} \\
&= \begin{bmatrix} a^2\boldsymbol{s}^T\mathbf{B}\boldsymbol{s} & ab\boldsymbol{s}^T\boldsymbol{y} \\ -ab\boldsymbol{y}^T\boldsymbol{s} & -1 \end{bmatrix}
\end{aligned}
$$

DFP usando aproximación del Hessiano

$$
\begin{aligned}
\mathbf{I}+\mathbf{VBU} &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} + \begin{bmatrix} a^2\boldsymbol{s}^T\mathbf{B}\boldsymbol{s} & ab\boldsymbol{s}^T\boldsymbol{y} \\ -ab\boldsymbol{y}^T\boldsymbol{s} & -1 \end{bmatrix} \\
&= \begin{bmatrix} a^2\boldsymbol{s}^T\mathbf{B}\boldsymbol{s}+1 & ab\boldsymbol{s}^T\boldsymbol{y} \\ -ab\boldsymbol{y}^T\boldsymbol{s} & 0 \end{bmatrix}
\end{aligned}
$$

Luego

$$
(\mathbf{I}+\mathbf{VBU})^{-1} = \frac{1}{(ab)^2(\boldsymbol{y}^T\boldsymbol{s})^2} \begin{bmatrix} 0 & -ab\boldsymbol{s}^T\boldsymbol{y} \\ ab\boldsymbol{y}^T\boldsymbol{s} & a^2\boldsymbol{s}^T\mathbf{B}\boldsymbol{s}+1 \end{bmatrix}
$$

DFP usando aproximación del Hessiano

$$
\begin{aligned}
& \mathbf{U}(\mathbf{I}+\mathbf{VBU})^{-1}\mathbf{V} \\
= \quad & \frac{1}{(ab)^2(\boldsymbol{y}^T\boldsymbol{s})^2}[a\boldsymbol{s}, b\mathbf{H}\boldsymbol{y}]\begin{bmatrix} 0 & -ab\boldsymbol{y}^T\boldsymbol{s} \\ ab\boldsymbol{y}^T\boldsymbol{s} & a^2\boldsymbol{s}^T\mathbf{B}\boldsymbol{s}+1 \end{bmatrix}\begin{bmatrix} a\boldsymbol{s}^T \\ -b\boldsymbol{y}^T\mathbf{H} \end{bmatrix} \\
= \quad & \frac{1}{(ab)^2(\boldsymbol{y}^T\boldsymbol{s})^2}[a\boldsymbol{s}, b\mathbf{H}\boldsymbol{y}]\begin{bmatrix} ab^2\boldsymbol{y}^T\boldsymbol{s}\boldsymbol{y}^T\mathbf{H} \\ a^2b\boldsymbol{y}^T\boldsymbol{s}\boldsymbol{s}^T - b(a^2\boldsymbol{s}^T\mathbf{B}\boldsymbol{s}+1)\boldsymbol{y}^T\mathbf{H} \end{bmatrix} \\
= \quad & \frac{a^2{\color{red}b^2}\boldsymbol{y}^T\boldsymbol{s}\boldsymbol{s}\boldsymbol{y}^T\mathbf{H} + a^2{\color{red}b^2}\boldsymbol{y}^T\boldsymbol{s}\mathbf{H}\boldsymbol{y}\boldsymbol{s}^T - {\color{red}b^2}(a^2\boldsymbol{s}^T\mathbf{B}\boldsymbol{s}+1)\mathbf{H}\boldsymbol{y}\boldsymbol{y}^T\mathbf{H}}{{\color{blue}a^2}{\color{red}b^2}(\boldsymbol{y}^T\boldsymbol{s})^2} \\
= \quad & \frac{{\color{red}a^2\boldsymbol{y}^T\boldsymbol{s}\mathbf{H}\boldsymbol{y}\boldsymbol{s}^T} + {\color{blue}a^2\boldsymbol{y}^T\boldsymbol{s}\boldsymbol{s}\boldsymbol{y}^T\mathbf{H}} - (a^2\boldsymbol{s}^T\mathbf{B}\boldsymbol{s}+1)\mathbf{H}\boldsymbol{y}\boldsymbol{y}^T\mathbf{H}}{\boldsymbol{y}^T\boldsymbol{s}} \\
= \quad & {\color{blue}a^2\mathbf{H}\boldsymbol{y}\boldsymbol{s}^T} + {\color{red}a^2\boldsymbol{s}\boldsymbol{y}^T\mathbf{H}} - \frac{(a^2\boldsymbol{s}^T\mathbf{B}\boldsymbol{s}+1)\mathbf{H}\boldsymbol{y}\boldsymbol{y}^T\mathbf{H}}{\boldsymbol{y}^T\boldsymbol{s}}
\end{aligned}
$$

DFP usando aproximación del Hessiano

Como $\rho = a^2 = \frac{1}{\boldsymbol{y}^T\boldsymbol{s}}$ entonces

$$\mathbf{U}(\mathbf{I}+\mathbf{VBU})^{-1}\mathbf{V} \quad = \quad \rho\mathbf{H}\boldsymbol{y}\boldsymbol{s}^T + \rho\boldsymbol{s}\boldsymbol{y}^T\mathbf{H} - \rho(\rho\boldsymbol{s}^T\mathbf{B}\boldsymbol{s}+1)\mathbf{H}\boldsymbol{y}\boldsymbol{y}^T\mathbf{H}$$

Luego

$$\begin{aligned} & \mathbf{BU}(\mathbf{I}+\mathbf{VBU})^{-1}\mathbf{VB} \\ = \quad & \rho\mathbf{BH}\boldsymbol{y}\boldsymbol{s}^T\mathbf{B} + \rho\mathbf{B}\boldsymbol{s}\boldsymbol{y}^T\mathbf{HB} - \rho(\rho\boldsymbol{s}^T\mathbf{B}\boldsymbol{s}+1)\mathbf{BH}\boldsymbol{y}\boldsymbol{y}^T\mathbf{HB} \\ = \quad & \rho\boldsymbol{y}\boldsymbol{s}^T\mathbf{B} + \rho\mathbf{B}\boldsymbol{s}\boldsymbol{y}^T - \rho(\rho\boldsymbol{s}^T\mathbf{B}\boldsymbol{s}+1)\boldsymbol{y}\boldsymbol{y}^T \end{aligned}$$

DFP usando aproximación del Hessiano

$$
\begin{aligned}
& (\mathbf{H}+\mathbf{UV})^{-1} \\
= \;& \mathbf{B}-\mathbf{BU}(\mathbf{I}+\mathbf{VBU})^{-1}\mathbf{VB} \\
= \;& \mathbf{B}-\rho \boldsymbol{y}\boldsymbol{s}^T\mathbf{B}-\rho\mathbf{B}\boldsymbol{s}\boldsymbol{y}^T+\rho^2\boldsymbol{s}^T\mathbf{B}\boldsymbol{s}\boldsymbol{y}\boldsymbol{y}^T+\rho\boldsymbol{y}\boldsymbol{y}^T \\
= \;& \mathbf{B}-\rho \boldsymbol{y}\boldsymbol{s}^T\mathbf{B}-\rho\mathbf{B}\boldsymbol{s}\boldsymbol{y}^T+\rho^2\boldsymbol{y}\boldsymbol{s}^T\mathbf{B}\boldsymbol{s}\boldsymbol{y}^T+\rho\boldsymbol{y}\boldsymbol{y}^T \\
= \;& (\mathbf{I}-\rho\boldsymbol{y}\boldsymbol{s}^T)\mathbf{B}-(\mathbf{I}-\rho\boldsymbol{y}\boldsymbol{s}^T)\rho\mathbf{B}\boldsymbol{s}\boldsymbol{y}^T+\rho\boldsymbol{y}\boldsymbol{y}^T \\
= \;& (\mathbf{I}-\rho\boldsymbol{y}\boldsymbol{s}^T)\mathbf{B}-(\mathbf{I}-\rho\boldsymbol{y}\boldsymbol{s}^T)\rho\mathbf{B}\boldsymbol{s}\boldsymbol{y}^T+\rho\boldsymbol{y}\boldsymbol{y}^T \\
= \;& (\mathbf{I}-\rho\boldsymbol{y}\boldsymbol{s}^T)(\mathbf{B}-\rho\mathbf{B}\boldsymbol{s}\boldsymbol{y}^T)+\rho\boldsymbol{y}\boldsymbol{y}^T \\
= \;& (\mathbf{I}-\rho\boldsymbol{y}\boldsymbol{s}^T)\mathbf{B}(\mathbf{I}-\rho\boldsymbol{s}\boldsymbol{y}^T)+\rho\boldsymbol{y}\boldsymbol{y}^T
\end{aligned}
$$

DFP usando aproximación del Hessiano

Finalmente, incluyendo los subíndices

$$\mathbf{B}_{k+1} = (\mathbf{I} - \rho_k \boldsymbol{y}_k \boldsymbol{s}_k^T)\mathbf{B}_k(\mathbf{I} - \rho_k \boldsymbol{s}_k \boldsymbol{y}_k^T) + \rho_k \boldsymbol{y}_k \boldsymbol{y}_k^T$$

con $\rho_k = \frac{1}{\boldsymbol{y}_k^T \boldsymbol{s}_k}$

Note que $\mathbf{B}_{k+1} \succ 0$ siempre que $\mathbf{B}_k \succ 0$, $\rho_k > 0$ y no se halla llegado a la solucion, esto ultimo significa que $\boldsymbol{y}_k \neq 0$. Para ello, se puede verificar que $\boldsymbol{x}^T \mathbf{B}_{k+1} \boldsymbol{x} > 0$ para todo $\boldsymbol{x} \neq 0$. Una idea es considerar los casos $\boldsymbol{x}^T \boldsymbol{y}_k = 0$ y $\boldsymbol{x}^T \boldsymbol{y}_k \neq 0$

DFP usando aproximación del Hessiano

En resumen, para Davidon-Fletcher-Powell (DFP) se cumple

$$\mathbf{H}_{k+1}^{DFP} = \mathbf{H}_k - \frac{\mathbf{H}_k \boldsymbol{y}_k \boldsymbol{y}_k^T \mathbf{H}_k}{y_k^T \mathbf{H}_k y_k} + \frac{\boldsymbol{s}_k \boldsymbol{s}_k^T}{\boldsymbol{y}_k^T \boldsymbol{s}_k}$$

Usando la formula de Sherman-Morrison-Woodbury

$$\mathbf{B}_{k+1}^{DFP} = (\mathbf{I} - \rho_k \boldsymbol{y}_k \boldsymbol{s}_k^T)\mathbf{B}_k(\mathbf{I} - \rho_k \boldsymbol{s}_k \boldsymbol{y}_k^T) + \rho_k \boldsymbol{y}_k \boldsymbol{y}_k^T$$

DFP- aproximando el Hessiano

Algorithm 2 DFP- aproximando el Hessiano

Require: $\boldsymbol{x}_0$ y $\mathbf{B}_0$
Ensure: $\boldsymbol{x}^*$

1: $k = 0$
2: **while** $\|\nabla f_k\| \neq 0$ (No conveja) **do**
3: $\quad \boldsymbol{d}_k = -\mathbf{B}_k^{-1} \nabla f_k$
4: $\quad$ Calcular α_k usando búsqueda en línea.
5: $\quad \boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \alpha_k \boldsymbol{d}_k$
6: $\quad$ Cacular ∇f_{k+1}, $\boldsymbol{y}_k$, $\boldsymbol{s}_k$, ρ_k y actualizar $\mathbf{B}_{k+1}$
$\quad \mathbf{B}_{k+1} = (\mathbf{I} - \rho_k \boldsymbol{y}_k \boldsymbol{s}_k^T)\mathbf{B}_k(\mathbf{I} - \rho_k \boldsymbol{s}_k \boldsymbol{y}_k^T) + \rho_k \boldsymbol{y}_k \boldsymbol{y}_k^T$
7: $\quad k = k + 1$
8: **end while**

Algoritmo DFP

Algorithm 3 DFP- aproximando la inversa del Hessiano

Require: $\boldsymbol{x}_0$ y $\mathbf{H}_0$

Ensure: $\boldsymbol{x}^*$

1: $k = 0$
2: **while** $\|\nabla f_k\| \neq 0$ (No conveja) **do**
3: $\boldsymbol{d}_k = -\mathbf{H}_k \nabla f_k$
4: Calcular α_k usando búsqueda en línea.
5: $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \alpha_k \boldsymbol{d}_k$
6: Cacular ∇f_{k+1}, $\boldsymbol{y}_k$, $\boldsymbol{s}_k$ y actualizar $\mathbf{H}_{k+1}$

$$\mathbf{H}_{k+1} = \mathbf{H}_k - \frac{\mathbf{H}_k \boldsymbol{y}_k \boldsymbol{y}_k^T \mathbf{H}_k}{\boldsymbol{y}_k^T \mathbf{H}_k \boldsymbol{y}_k} + \frac{\boldsymbol{s}_k \boldsymbol{s}_k^T}{\boldsymbol{y}_k^T \boldsymbol{s}_k}$$

7: $k = k + 1$
8: **end while**

Orden del Tema

Algoritmo BFGS

- En el BFGS la idea es calcular la aproximacion de matriz inversa del Hessiano $\mathbf{H}_{k+1}$ basado en $\mathbf{B}_{k+1}$
- Para ello se considera la ecuacion DFP, con $\rho_k = \frac{1}{\boldsymbol{s}_k^T \boldsymbol{y}_k}$

$$\mathbf{B}_{k+1} = (\mathbf{I} - \rho_k \boldsymbol{y}_k \boldsymbol{s}_k^T)\mathbf{B}_k(\mathbf{I} - \rho_k \boldsymbol{s}_k \boldsymbol{y}_k^T) + \rho_k \boldsymbol{y}_k \boldsymbol{y}_k^T$$

y las relaciones

$$\begin{array}{ccc} \mathbf{B}_{k+1} \boldsymbol{s}_k & = & \boldsymbol{y}_k \\ \updownarrow \quad \updownarrow & & \updownarrow \\ \mathbf{H}_{k+1} \boldsymbol{y}_k & = & \boldsymbol{s}_k \end{array}$$

Ahora $\rho_k = \frac{1}{\boldsymbol{y}_k^T \boldsymbol{s}_k}$ y por tanto:

$$\mathbf{H}_{k+1} = (\mathbf{I} - \rho_k \boldsymbol{s}_k \boldsymbol{y}_k^T)\mathbf{H}_k(\mathbf{I} - \rho_k \boldsymbol{y}_k \boldsymbol{s}_k^T) + \rho_k \boldsymbol{s}_k \boldsymbol{s}_k^T$$

Algoritmo BFGS

Algorithm 4 Algoritmo BFGS

Require: $\boldsymbol{x}_0$ y $\mathbf{H}_0$
Ensure: $\boldsymbol{x}^*$

```
k = 0
while ‖∇f_k‖ ≠ 0 (No conveja) do
    d_k = −H_k ∇f_k
    Calcular α_k usando búsqueda en línea.
    x_{k+1} = x_k + α_k d_k
    Cacular ∇f_{k+1}, y_k, s_k, ρ_k y actualizar H_{k+1}
    H_{k+1} = (I − ρ_k s_k y_k^T) H_k (I − ρ_k y_k s_k^T) + ρ_k s_k s_k^T
    k = k + 1
end while
```

Línea por línea:

$k = 0$

while $\|\nabla f_k\| \neq 0$ (No conveja) **do**

$\quad \boldsymbol{d}_k = -\mathbf{H}_k \nabla f_k$

$\quad$ Calcular α_k usando búsqueda en línea.

$\quad \boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \alpha_k \boldsymbol{d}_k$

$\quad$ Cacular ∇f_{k+1}, $\boldsymbol{y}_k$, $\boldsymbol{s}_k$, ρ_k y actualizar $\mathbf{H}_{k+1}$

$$\mathbf{H}_{k+1} = (\mathbf{I} - \rho_k \boldsymbol{s}_k \boldsymbol{y}_k^T)\mathbf{H}_k(\mathbf{I} - \rho_k \boldsymbol{y}_k \boldsymbol{s}_k^T) + \rho_k \boldsymbol{s}_k \boldsymbol{s}_k^T$$

$\quad k = k + 1$

end while

BFGS aproximación del Hessiano

$$
\begin{aligned}
\mathbf{H}_{k+1}^{DFP} &= \mathbf{H}_k - \frac{\mathbf{H}_k \boldsymbol{y}_k \boldsymbol{y}_k^T \mathbf{H}_k}{y_k^T \mathbf{H}_k y_k} + \frac{\boldsymbol{s}_k \boldsymbol{s}_k^T}{\boldsymbol{y}_k^T \boldsymbol{s}_k} \\
\mathbf{B}_{k+1}^{BFGS} &= \mathbf{B}_k - \frac{\mathbf{B}_k \boldsymbol{s}_k \boldsymbol{s}_k^T \mathbf{B}_k}{s_k^T \mathbf{B}_k s_k} + \frac{\boldsymbol{y}_k \boldsymbol{y}_k^T}{\boldsymbol{s}_k^T \boldsymbol{y}_k}
\end{aligned}
$$

DFP vs BFGS

En resumen

$$
\begin{aligned}
\mathbf{H}_{k+1}^{DFP} &= \mathbf{H}_k - \frac{\mathbf{H}_k \boldsymbol{y}_k \boldsymbol{y}_k^T \mathbf{H}_k}{y_k^T \mathbf{H}_k y_k} + \frac{\boldsymbol{s}_k \boldsymbol{s}_k^T}{\boldsymbol{y}_k^T \boldsymbol{s}_k} \\
\mathbf{B}_{k+1}^{BFGS} &= \mathbf{B}_k - \frac{\mathbf{B}_k \boldsymbol{s}_k \boldsymbol{s}_k^T \mathbf{B}_k}{s_k^T \mathbf{B}_k s_k} + \frac{\boldsymbol{y}_k \boldsymbol{y}_k^T}{\boldsymbol{s}_k^T \boldsymbol{y}_k}
\end{aligned}
$$

$$
\begin{aligned}
\mathbf{B}_{k+1}^{DFP} &= (\mathbf{I} - \rho_k \boldsymbol{y}_k \boldsymbol{s}_k^T)\mathbf{B}_k(\mathbf{I} - \rho_k \boldsymbol{s}_k \boldsymbol{y}_k^T) + \rho_k \boldsymbol{y}_k \boldsymbol{y}_k^T \\
\mathbf{H}_{k+1}^{BFGS} &= (\mathbf{I} - \rho_k \boldsymbol{s}_k \boldsymbol{y}_k^T)\mathbf{H}_k(\mathbf{I} - \rho_k \boldsymbol{y}_k \boldsymbol{s}_k^T) + \rho_k \boldsymbol{s}_k \boldsymbol{s}_k^T
\end{aligned}
$$

Si el tamaño de paso cumple las condiciones de Wolfe con $0 < c_2 < 1$, y si $\nabla f_k \neq 0$, entonces

$$
\begin{aligned}
\nabla f_{k+1}^T \boldsymbol{d}_k &\geq c_2 \nabla f_k^T \boldsymbol{d}_k \\
\nabla f_{k+1}^T \alpha_k \boldsymbol{d}_k &\geq c_2 \nabla f_k^T \alpha_k \boldsymbol{d}_k \\
\boldsymbol{g}_{k+1}^T \boldsymbol{s}_k &\geq c_2 \boldsymbol{g}_k^T \boldsymbol{s}_k \\
\boldsymbol{g}_{k+1}^T \boldsymbol{s}_k - \boldsymbol{g}_k^T \boldsymbol{s}_k &\geq c_2 \boldsymbol{g}_k^T \boldsymbol{s}_k - \boldsymbol{g}_k^T \boldsymbol{s}_k \\
\boldsymbol{y}_k^T \boldsymbol{s}_k &\geq (c_2 - 1) \boldsymbol{g}_k^T \boldsymbol{s}_k \\
\boldsymbol{y}_k^T \boldsymbol{s}_k &\geq \underbrace{(c_2 - 1)}_{<0} \overbrace{\alpha_k}^{>0} \underbrace{\boldsymbol{g}_k^T \boldsymbol{d}_k}_{<0}
\end{aligned}
$$

Luego

$$
\boldsymbol{y}_k^T \boldsymbol{s}_k \geq (c_2 - 1) \alpha_k \boldsymbol{g}_k^T \boldsymbol{d}_k > 0
$$

por tanto $\rho_k > 0$ y $\mathbf{H}_{k+1}$ es definida positiva.

Orden del Tema

Algoritmo L-BFGS

- El algoritmo BGFS es presenta un buen desempleño y no requiere el cálculo del Hessiano ni de su inversa.
- La principal limitación del algoritmo BFGS es en problemas donde el número de variable es muy grande (por ejemplo, un millón de variables o mas) pues es casi imposible guardar la matriz de tamaño $n \times n$.
- Una solución a problemas de *gran escala* es el algoritmo *L-BFGS* ('Limited memory BFGS').

Algoritmo L-BFGS

- **Observación 1:** La matriz

$$\mathbf{H}_k = (\mathbf{I} - \rho_{k-1}\boldsymbol{s}_{k-1}\boldsymbol{y}_{k-1}^T)\mathbf{H}_{k-1}(\mathbf{I} - \rho_{k-1}\boldsymbol{y}_{k-1}\boldsymbol{s}_{k-1}^T) + \rho_{k-1}\boldsymbol{s}_{k-1}\boldsymbol{s}_{k-1}^T$$

se actualiza usando matrices de rango 1, i.e.,

$$\begin{aligned} & \boldsymbol{s}_{k-1}\boldsymbol{s}_{k-1}^T, \\ \mathbf{V}_{k-1} &= \mathbf{I} - \rho_{k-1}\boldsymbol{y}_{k-1}\boldsymbol{s}_{k-1}^T, \\ \mathbf{V}_{k-1}^T &= \mathbf{I} - \rho_{k-1}\boldsymbol{s}_{k-1}\boldsymbol{y}_{k-1}^T. \end{aligned}$$

Algoritmo L-BFGS

- **Observación 2:** Para resolver el problema de optimización, lo que se require es calcular el paso $\boldsymbol{d}_k = -\mathbf{H}_k \nabla f_k$
- Es decir, en la práctica no es interés conocer $\mathbf{H}_k$ sino el producto $\mathbf{H}_k \nabla f_k$
- **Observación 3:** De

$$\mathbf{H}_k \nabla f_k = \mathbf{V}_{k-1}^T \mathbf{H}_{k-1} \mathbf{V}_{k-1} \nabla f_k + \rho_{k-1} \boldsymbol{s}_{k-1} (\boldsymbol{s}_{k-1}^T \nabla f_k)$$

donde $\mathbf{V}_{k-1} \nabla f_k = \nabla f_k - \rho_{k-1} \boldsymbol{y}_{k-1} (\boldsymbol{s}_{k-1}^T \nabla f_k)$ y $\rho_{k-1} \boldsymbol{s}_{k-1} (\boldsymbol{s}_{k-1}^T \nabla f_k)$ son vectores, y las operaciones son sumas y productos interiores.

Algoritmo L-BFGS

- La ecuación del Hessiano se puede reescribir, sustituyendo $\mathbf{H}_{k-1}$ por su ecuación en función de $\mathbf{H}_{k-2}$

$$\begin{aligned}\mathbf{H}_k &= (\mathbf{V}_{k-1}^T \mathbf{V}_{k-2}^T)\mathbf{H}_{k-2}(\mathbf{V}_{k-1}\mathbf{V}_{k-2}) \\ &\quad + \mathbf{V}_{k-1}^T \boldsymbol{s}_{k-2}\rho_{k-2}\boldsymbol{s}_{k-2}^T \mathbf{V}_{k-1} + \boldsymbol{s}_{k-1}\rho_{k-1}\boldsymbol{s}_{k-1}^T\end{aligned}$$

- En azul y rojo son vectores o matrices donde el cálculo solo involucra productos interiores.
- En verde aparece la matriz $\mathbf{H}_{k-2}$ y otras operaciones matriciales.

Algoritmo L-BFGS

- Al multiplicar por ∇f_k

$$\begin{aligned}\mathbf{H}_k \nabla f_k &= (\mathbf{V}_{k-1}^T \mathbf{V}_{k-2}^T)\mathbf{H}_{k-2}(\mathbf{V}_{k-1}\mathbf{V}_{k-2})\nabla f_k \\ &\quad + \mathbf{V}_{k-1}^T \boldsymbol{s}_{k-2}\rho_{k-2}\boldsymbol{s}_{k-2}^T \mathbf{V}_{k-1}\nabla f_k \\ &\quad + \boldsymbol{s}_{k-1}\rho_{k-1}\boldsymbol{s}_{k-1}^T \nabla f_k\end{aligned}$$

- En azul son vectores, y su cálculo solo involucra productos interiores.
- En rojo el producto resultante es un número real y su cálculo solo involucra productos interiores.
- En verde aparece la matriz $\mathbf{H}_{k-2}$ y otras operaciones matriciales.

Algoritmo L-BFGS

- Continuado el proceso

$$
\begin{aligned}
\mathbf{H}_k = \\
(\mathbf{V}^T_{-1}\mathbf{V}^T_{-2}\cdots\mathbf{V}^T_{-m})\mathbf{H}^0_{-m}(\mathbf{V}_{-1}\mathbf{V}_{-2}\cdots\mathbf{V}_{-m}) \\
+(\mathbf{V}^T_{-1}\mathbf{V}^T_{-2}\cdots\mathbf{V}^T_{-m+1})\boldsymbol{s}_{-m}\rho_{-m}\boldsymbol{s}^T_{-m}(\mathbf{V}_{-1}\mathbf{V}_{-2}\cdots\mathbf{V}_{-m+1}) \\
+(\mathbf{V}^T_{-1}\mathbf{V}^T_{-2}\cdots\mathbf{V}^T_{-m+2})\boldsymbol{s}_{-m+1}\rho_{-m+1}\boldsymbol{s}^T_{-m+1}(\mathbf{V}_{-1}\mathbf{V}_{-2}\cdots\mathbf{V}_{-m+2}) \\
\cdots \\
+\mathbf{V}^T_{-1}\boldsymbol{s}_{-2}\rho_{-2}\boldsymbol{s}^T_{-2}\mathbf{V}_{-1} \\
+\boldsymbol{s}_{-1}\rho_{-1}\boldsymbol{s}^T_{-1}
\end{aligned}
$$

Nota: para el índice $k-i$ se usa $-i$ y $\mathbf{H}^0_{-m} = \mathbf{H}_{k-m}$

Algoritmo L-BFGS

- $\mathbf{H}^0_{-m}$ juega el papel de $\mathbf{H}_0$ en el BFGS y es como si en lugar de iniciar en $\boldsymbol{x}_0$ hubiéramos iniciado en $\boldsymbol{x}_{-m}$.
- En problema en el algoritmo anterior es que se requiere calcular $\mathbf{H}^0_{-m}\boldsymbol{q}$, y no queremos guardar $\mathbf{H}^0_{-m}$.
- Para ello se toma $\mathbf{H}^0_{-m}$ usando una forma simple, por ejemplo, un múltiplo de la matriz identidad.
- En la práctica $\mathbf{H}^0_{-m} = \gamma_k \mathbf{I}$ con $\gamma_k = \frac{\boldsymbol{s}^T_{k-1}\boldsymbol{y}_{k-1}}{\boldsymbol{y}^T_{k-1}\boldsymbol{y}_{k-1}}$ ha mostrado buenos resultados.

Algoritmo L-BFGS

- El cáculo de $\mathbf{H}_k \nabla f_k$ se puede realizar guardando, por ejemplo, m pareja de vectores $\{\boldsymbol{s}_i, \boldsymbol{y}_i\}$ anteriores
- El cáculo se realiza a través de productos interiores y sumas.
- Para la siguiente iteración se guarda la última pareja $\{\boldsymbol{s}_k, \boldsymbol{y}_k\}$ y se remueve la primera.
- Notar que se gana en cuanto a memoria, sin embargo, aumenta el tiempo de cómputo!!!.

Algoritmo L-BFGS

Algorithm 5 Algoritmo para calcular el paso L-BFGS

Require: ∇f_k, $\boldsymbol{s}_{-i}$, $\boldsymbol{y}_{-i}$ para $i = 1, \cdots, m$

Ensure: $\boldsymbol{r}$

$\boldsymbol{q} = \nabla f_k$
for $(i = 1 \cdots m)$ **do**
  $\alpha_i = \rho_{-i} \boldsymbol{s}_{-i}^T \boldsymbol{q}$
  $\boldsymbol{q} = \boldsymbol{q} - \alpha_i \boldsymbol{y}_{-i}$
end for
$\boldsymbol{r} = \mathbf{H}_{-m}^0 \boldsymbol{q}$
for $(i = m \cdots 1)$ **do**
  $\beta = \rho_{-i} \boldsymbol{y}_{-i}^T \boldsymbol{r}$
  $\boldsymbol{r} = \boldsymbol{r} + \boldsymbol{s}_{-i}(\alpha_i - \beta)$
end for

Algoritmo L-BFGS

Por qué trabaja?

- Sea $\boldsymbol{q}_i$ el valor de $\boldsymbol{q}$ despueś de la i-ésima iteración del primer ciclo y $\boldsymbol{q}_0 = \nabla f_k$. De acuerdo al algoritmo

$$
\begin{aligned}
\boldsymbol{q}_i &= \boldsymbol{q}_{i-1} - \rho_{-i}\boldsymbol{y}_{-i}\boldsymbol{s}_{-i}^T\boldsymbol{q}_{i-1} = \mathbf{V}_{-i}\boldsymbol{q}_{i-1} = (\mathbf{V}_{-i}\mathbf{V}_{-i+1})\boldsymbol{q}_{i-2} \\
&= \cdots = (\mathbf{V}_{-i}\cdots\mathbf{V}_{-1})\nabla f_k
\end{aligned}
$$

Luego

$$
\alpha_i = \rho_{-i}\boldsymbol{s}_{-i}^T\boldsymbol{q}_{i-1} = \cdots = \rho_{-i}\boldsymbol{s}_{-i}^T(\mathbf{V}_{-i+1}\cdots\mathbf{V}_{-1})\nabla f_k
$$

Algoritmo L-BFGS

Por qué trabaja?

- Calculando $\boldsymbol{r}$.

$$\boldsymbol{r}_m = \mathbf{H}_{-m}\boldsymbol{q}_m = \mathbf{H}_{-m}(\mathbf{V}_{-m}\cdots\mathbf{V}_{-1})\nabla f_k$$

- Por otro lado

$$\begin{aligned} \boldsymbol{r}_i &= \boldsymbol{r}_{i+1} + \boldsymbol{s}_{-i}(\alpha_i - \rho_{-i}\boldsymbol{y}_{-i}^T\boldsymbol{r}_{i+1}) \\ &= (I - \boldsymbol{s}_{-i}\rho_{-i}\boldsymbol{y}_{-i}^T)\boldsymbol{r}_{i+1} + \boldsymbol{s}_{-i}\alpha_i \\ &= \mathbf{V}_{-i}^T\boldsymbol{r}_{i+1} + \boldsymbol{s}_{-i}\alpha_i \end{aligned}$$

Algoritmo L-BFGS

Por qué trabaja?

- Comenzando por $\boldsymbol{r}_1$ y usando la recursividad anterior y la formula para α_i

$$
\begin{aligned}
\boldsymbol{r}_1 &= \mathbf{V}_{-1}^T \boldsymbol{r}_2 + \boldsymbol{s}_{-1}\rho_{-1}\boldsymbol{s}_{-1}^T \nabla f_k \\
&= (\mathbf{V}_{-1}^T \mathbf{V}_{-2}^T)\boldsymbol{r}_3 + \mathbf{V}_{-1}^T \boldsymbol{s}_{-2}\rho_{-2}\boldsymbol{s}_{-2}^T \mathbf{V}_{-1}\nabla f_k + +\boldsymbol{s}_{-1}\rho_{-1}\boldsymbol{s}_{-1}^T \nabla f \\
&= \cdots \\
&= (\mathbf{V}_{-1}^T \mathbf{V}_{-2}^T \cdots \mathbf{V}_{-m}^T) H_{-m}^0 (\mathbf{V}_{-1}\mathbf{V}_{-2}\cdots\mathbf{V}_{-m})\nabla f_k \\
&\quad + \cdots + \\
&\quad + \mathbf{V}_{-1}^T \boldsymbol{s}_{-2}\rho_{-2}\boldsymbol{s}_{-2}^T \mathbf{V}_{-1}\nabla f_k \\
&\quad + \boldsymbol{s}_{-1}\rho_{-1}\boldsymbol{s}_{-1}^T \nabla f_k \\
&= \mathbf{H}_k \nabla f_k
\end{aligned}
$$

Algoritmo L-BFGS

Algorithm 6 Algoritmo L-BFGS

for $(i = 1, 2, \cdots)$ **do**
  Seleccionar $\mathbf{H}^0_{-m}$
  Calcular $\boldsymbol{d}_k = -\mathbf{H}_k \nabla f_k$, usando el algoritmo anterior de 'Limited Memory'
  Calcular α_k usando búsqueda en línea.
  $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \alpha_k \boldsymbol{d}_k$
  if $k > m$ **then**
    Eliminar la primera pareja $\{\boldsymbol{s}_{k-m}, \boldsymbol{y}_{k-m}\}$
  end if
  Cacular y Guardar $\{\boldsymbol{s}_k = \boldsymbol{x}_{k+1} - \boldsymbol{x}_k, \boldsymbol{y}_k = \nabla f_{k+1} - \nabla f_k\}$
end for