OPTIMIZACIÓN SIN RESTRICCIONES

En optimización sin restricciones se minimiza una función objetivo que depende de variables reales sin restricciones sobre los valores de esas variables. La formulación matemática es:

$$(\mathcal{OSR}) \begin{cases} \min\limits_{x \in \mathbb{R}^n} & f(x) \end{cases}$$

donde f es una función suficientemente regular.

Caracterización de un mínimo

¿Qué es una solución?

Un punto x^* es un MÍNIMO GLOBAL si $f(x^*) \leq f(x)$ para todo $x \in \mathbb{R}^n$.

Sin embargo el mínimo global puede ser difícil de encontrar pues nuestro conocimiento de f es usualmente local. La mayoría de los algoritmos calculan **mínimos locales** que son puntos en los que se alcanza el menor valor de f en su entorno.

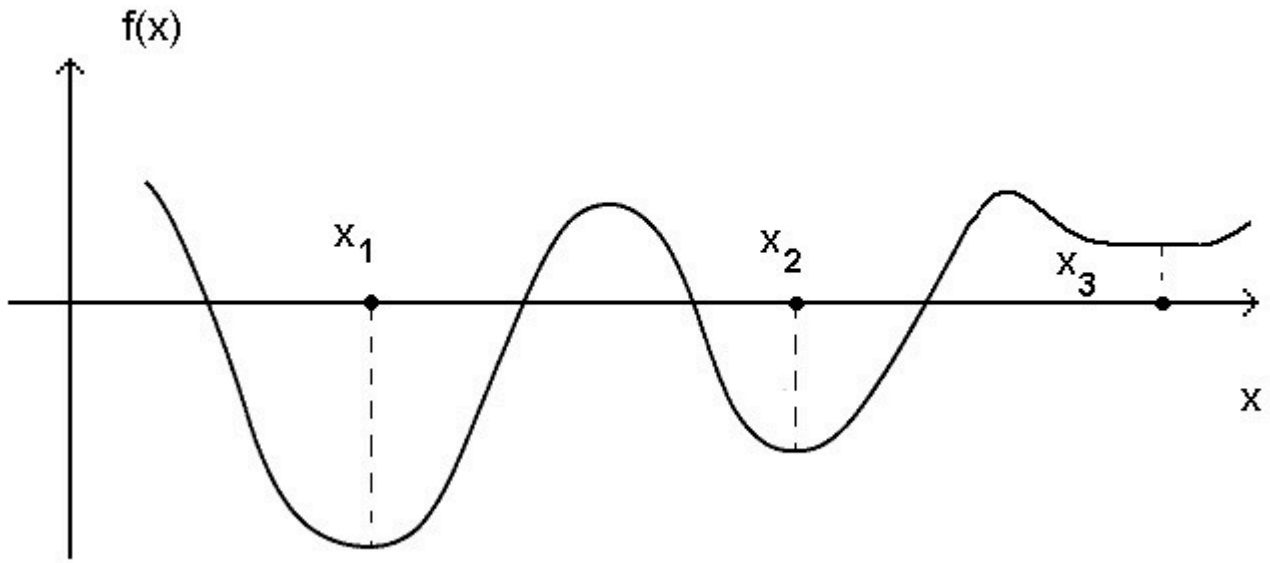
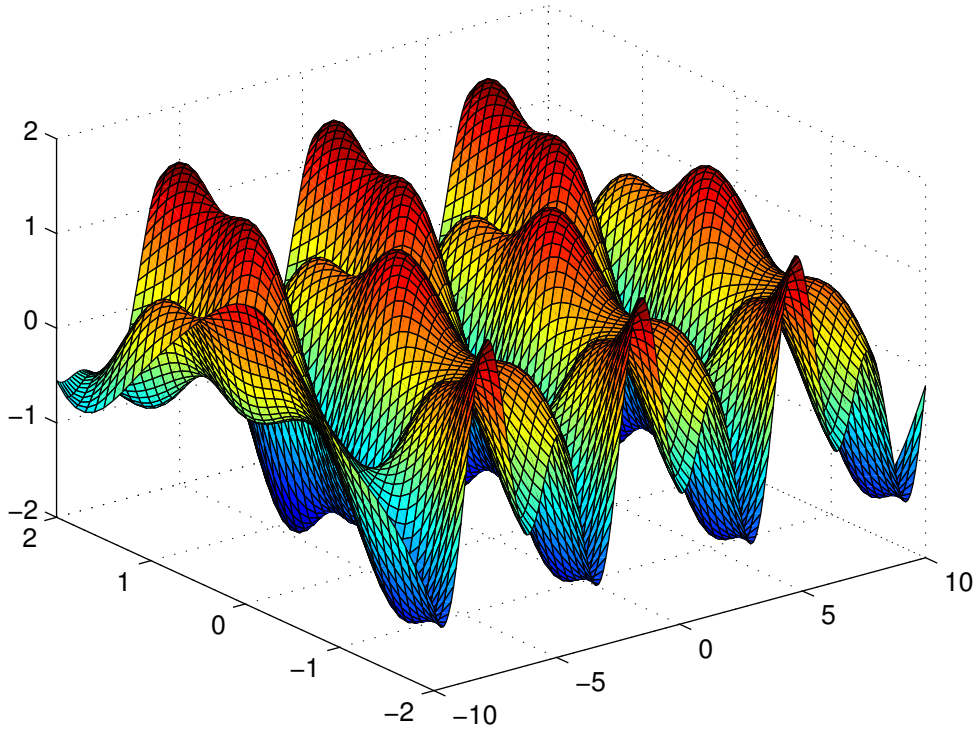


Figure 1: Ejemplos de mínimos: x_1 mínimo *global*, x_2 mínimo *local estricto*, x_3 mínimo *local no estricto*

Formalmente:

Un punto x^* es un MÍNIMO LOCAL si existe un entorno $\mathcal{N}$ de x^* tal que $f(x^*) \leq f(x)$ para todo $x \in \mathcal{N}$.

Hay funciones con muchos mínimos locales. Es generalmente difícil encontrar el mínimo global para tales funciones. Un ejemplo con millones de mínimos locales aparece en la determinación de la conformación de una molécula con energía potencial mínima.



¿Cómo reconocer un mínimo local?

Si la función f es suficientemente regular existen modos eficientes y prácticos de identificar mínimos locales. En particular si f es dos veces diferenciable puede decirse si x^* es un mínimo local examinando su gradiente $\nabla f(x^*)$ y su hessiano $\nabla^2 f(x^*)$.

En concreto:

- CONDICIÓN NECESARIA DE PRIMER ORDEN:

  Si x^* es un mínimo local y f es continuamente diferenciable en un entorno abierto de x^*, entonces $\nabla f(x^*) = 0$. (x^* punto estacionario)

- CONDICIONES NECESARIAS DE SEGUNDO ORDEN:

  Si x^* es un mínimo local y $\nabla^2 f$ es continua en un entorno abierto de x^*, entonces $\nabla f(x^*) = 0$ y $\nabla^2 f(x^*)$ es semidefinida positiva.

- CONDICIONES SUFICIENTES DE SEGUNDO ORDEN:

  Si $\nabla^2 f$ es continua en un entorno abierto de x^*, $\nabla f(x^*) = 0$ y $\nabla^2 f(x^*)$ es definida positiva, entonces x^* es un mínimo local estricto de f.

Las condiciones suficientes no son necesarias. Un ejemplo simple está dado por la función $f(x) = x^4$ para la que el punto $x^* = 0$ es un mínimo local estricto y sin embargo su hessiano es nulo (y por tanto no definido positivo).

Cuando la función objetivo es **convexa**, los mínimos locales y globales son fáciles de caracterizar:

Cuando f es convexa, cualquier mínimo local x^* es un mínimo global de f. Si además f es diferenciable, entonces cualquier punto estacionario x^* es un mínimo global de f.

EJEMPLO:

$$f(x_1, x_2) = x_1^2 + x_2^2 + \alpha x_1 x_2 + x_1 + 2x_2 =$$
$$= \frac{1}{2} \begin{pmatrix} x_1 & x_2 \end{pmatrix} \begin{pmatrix} 2 & \alpha \\ \alpha & 2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \begin{pmatrix} 1 & 2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$$

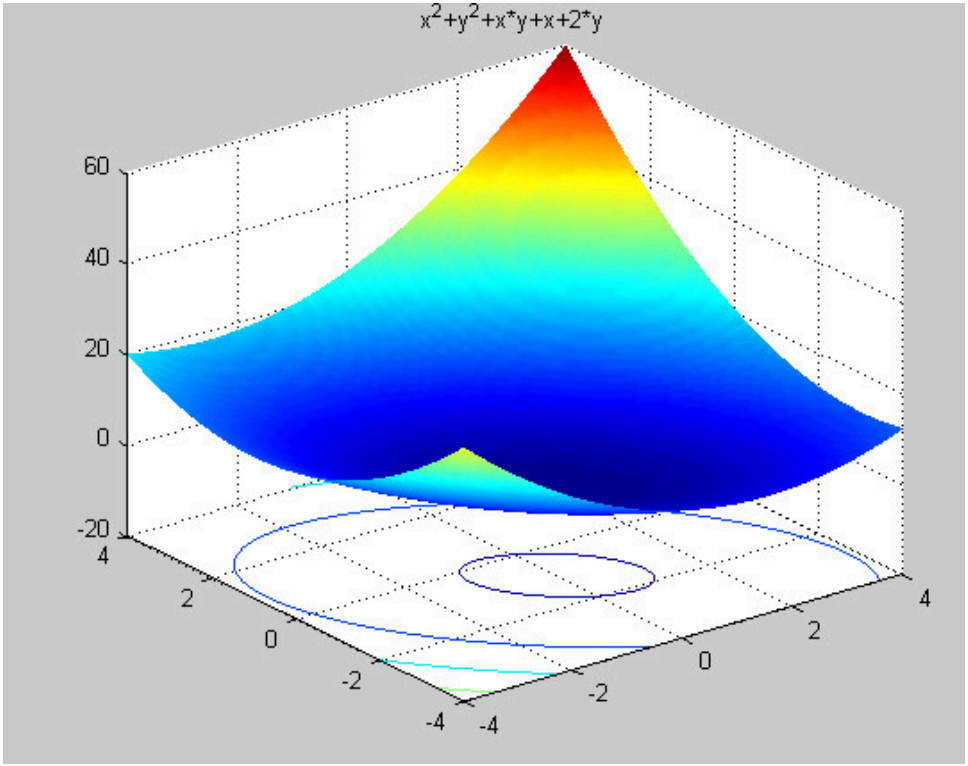


Figure 2: Caso $\alpha = 1$

$$\nabla f(x_1, x_2) = \begin{pmatrix} 2x_1 + \alpha x_2 + 1 \\ 2x_2 + \alpha x_1 + 2 \end{pmatrix}$$

$$\nabla^2 f(x_1, x_2) = \begin{pmatrix} 2 & \alpha \\ \alpha & 2 \end{pmatrix}$$

En este caso el hessiano es definido positivo si y solo si $4 - \alpha^2 > 0$ o equivalentemente, $|\alpha| < 2$. En este caso existe un mínimo y es único (ver Figura 2).

Si $|\alpha| = 2$ el hessiano es semidefinido positivo: podría existir mínimo o no. En este caso concreto no existe dicho mínimo pues para $\alpha = 2$ y $\alpha = -2$ el sistema:

$$\nabla f(x_1, x_2) = \begin{pmatrix} 2x_1 + \alpha x_2 + 1 \\ 2x_2 + \alpha x_1 + 2 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

es incompatible (ver Figura 3).

Por último, si $|\alpha| > 2$ el hessiano es indefinido y no existe mínimo aunque sí puntos silla (ver Figura 4).

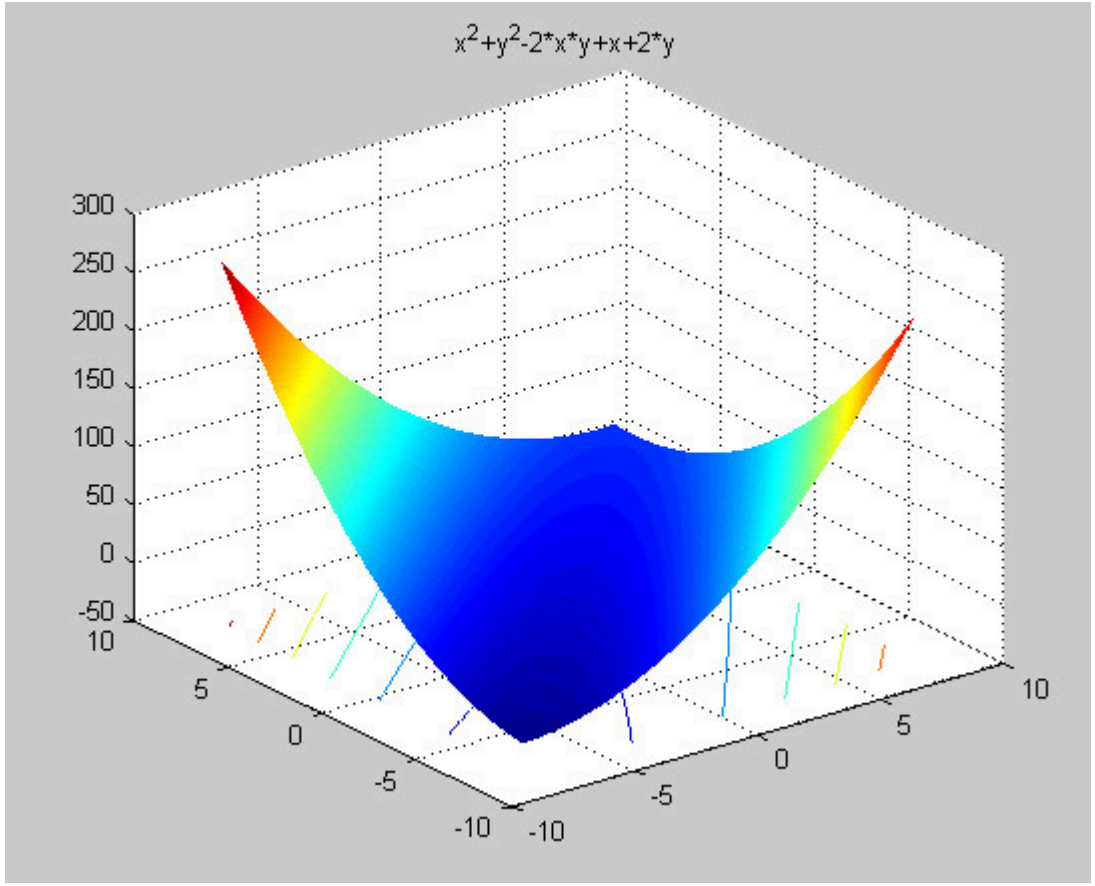


Figure 3: Caso $\alpha = -2$

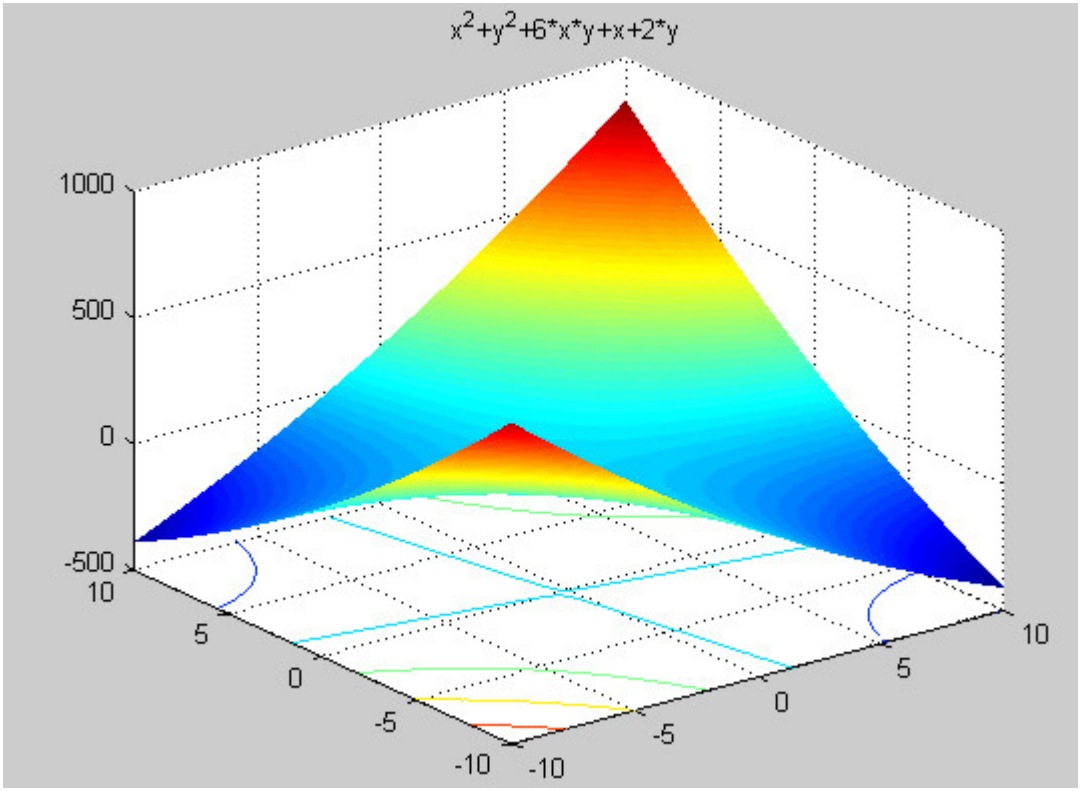


Figure 4: Caso $\alpha = 6$

Problemas no regulares

Existen muchos problemas interesantes en los que las funciones involucradas pueden ser no diferenciables e incluso discontinuas.

- Si la función está formada por unos pocos trozos regulares con discontinuidades entre los trozos, puede ser posible encontrar el mínimo analizando cada trozo por separado.
- Si la función es continua en todo punto pero es no diferenciable en ciertos puntos, puede identificarse la solución analizando los subgradientes, o los gradientes generalizados que son una generalización del concepto de gradiente al caso no regular.

Una visión de conjunto de los algoritmos

- Se considera sólo el caso regular.

- Todos los algoritmos para minimización sin restricciones necesitan que el usuario suministre un punto inicial, que se denotará por x_0.

- Comenzando en x_0, los algoritmos de optimización generan una sucesión $\{x_k\}$ que terminan cuando el algoritmo no puede progresar más o cuando parece que un punto solución se ha aproximado con precisión suficiente.

- Para decidir como pasar de un iterante x_k al siguiente los algoritmos utilizan información sobre f y sus derivadas en el iterante x_k o incluso de todos los iterantes anteriores $x_0, x_1, \ldots, x_{k-1}, x_k$.

Existen dos **estrategias fundamentales para pasar de un iterante a otro**. La mayoría de los algoritmos siguen alguna de estas estrategias.

Dos estrategias: Búsqueda de línea y Región de confianza

- **Búsqueda de línea**:
  - el algoritmo elige una dirección d_k

– busca a lo largo de esa dirección desde el iterante actual a un nuevo iterante con un menor valor de la función objetivo

– la distancia que debe moverse el iterante puede encontrarse resolviendo de manera aproximada el siguiente problema de minimización unidimensional que consiste en encontrar un paso t tal que:

$$(\mathcal{BL}) \begin{cases} \min\limits_{t>0} f(x_k + td_k) \end{cases}$$

(Resolviendo exactamente este problema se obtendría el mayor beneficio, pero hacerlo es caro e innecesario pues el objetivo final es la optimización de f y no la de $f(x_k + td_k)$.)

– en cada nuevo punto se calcula una nueva dirección de búsqueda y un nuevo paso y se repite el proceso.

- **Región de confianza**:

  – se construye una función modelo $m_k(x)$ cuyo comportamiento cerca del iterante actual x_k es similar al de la función objetivo f.

– como el modelo m_k puede no ser una buena aproximación de f cuando x está lejos de x_k, restringimos la búsqueda de un óptimo de m_k en alguna región en torno a x_k. Es decir encontramos el candidato d_k resolviendo de manera aproximada el siguiente subproblema:

$$\begin{cases} \min\limits_{d} m_k(x_k + d) \end{cases} \quad \text{donde } x_k + d \text{ está dentro de la región de confianza}$$

– Si la solución candidata no produce un decrecimiento suficiente en f se concluye que la región de confianza es demasiado grande y debe reducirse para después resolver el nuevo subproblema asociado.

Direcciones de búsqueda para métodos con búsqueda lineal

- **dirección de mayor descenso**:
  - Entre todas las direcciones en que podemos movernos desde x_k la dirección $d_k = -\nabla f(x_k)$ es aquella a lo largo de la cuál f decrece más deprisa.
  - Sin embargo si el condicionamiento de la matriz hessiana en el óptimo es grande la convergencia es muy lenta.

- **dirección de Newton**:
  - Esta dirección se deriva de la aproximación de Taylor de segundo orden en $f(x_k + d)$:

$$f(x_k + d) \approx f(x_k) + d^t \nabla f(x_k) + \frac{1}{2} d^t \nabla^2 f(x_k) d$$

    La dirección de Newton es entonces $d_k = -\nabla^2 f(x_k)^{-1} \nabla f(x_k)$.
  - Si el hessiano $\nabla^2 f(x_k)$ es definido positivo, d_k es una dirección de descenso y minimiza el modelo dado por la aproximación de Taylor. En ese caso el paso ideal en la búsqueda lineal es 1.
  - Los métodos usando la dirección de Newton son muy rápidos pero costosos puesto que hay que calcular y almacenar el hessiano.

- **direcciones de cuasi-Newton**:
  - Constituyen una atractiva alternativa a la dirección de Newton puesto que no necesitan el cálculo del hessiano y tienen convergencia superlineal.

– En lugar del hessiano $\nabla^2 f(x_k)$ utilizan una aproximación B_k que se actualiza en cada paso usando los gradientes de f en esa iteracion y la anterior. La dirección es entonces $d_k = -B_k^{-1}\nabla f(x_k)$.

- **métodos no lineales de gradiente conjugado**: en este caso la dirección tiene la forma $d_k = -\nabla f(x_k) + \beta_k d_{k-1}$, donde β_k es un escalar que asegura que d_k y d_{k-1} son conjugados.

Modelos para métodos con región de confianza

El modelo en un método de región de confianza está usualmente definido por una función cuadrática de la forma:

$$m_k(x_k + d) = f(x_k) + d^t \nabla f(x_k) + \frac{1}{2} d^t B_k d$$

donde la matriz B_k es, o bien el hessiano $\nabla^2 f(x_k)$, o alguna aproximación de éste.

- Si $B_k = 0$ y se define la región de confianza utilizando la norma euclídea, el problema de región de confianza es:

$$\begin{cases} \min\limits_{d} \ f(x_k) + d^t \nabla f(x_k) \\ \end{cases} \quad \text{sujeto a } \|d\| \leq \Delta_k$$

Puede calcularse directamente la solución de este problema y tiene la forma

$$d_k = -\frac{\Delta_k \nabla f(x_k)}{\|\nabla f(x_k)\|}$$

Este es un simple paso de descenso en el que la longitud del paso está determinado por el radio de la región de confianza; la búsqueda lineal y la región de confianza son esencialmente lo mismo en este caso.

- Un algoritmo de región de confianza más interesante se obtiene eligiendo B_k como el hessiano $\nabla^2 f(x_k)$ en el modelo cuadrático. Como se tiene la restricción de la región de confianza $\|d\| \leq \Delta_k$ no es necesario que $\nabla^2 f(x_k)$ sea definido positivo pues se tiene asegurada la existencia de una solución d_k. Este método llamado **método de Newton con región de confianza** es muy eficiente en la práctica.

- Si la matriz B_k en el modelo cuadrático se define con una aproximación cuasi-Newton se obtiene un **método cuasi-Newton con región de confianza**.

Un algoritmo general de descenso

Se desea resolver numéricamente el problema

$$(\mathcal{OSR}) \left\{ \begin{array}{l} \min \\ x \in \mathbb{R}^n \end{array} \right. f(x)$$

bajo la hipótesis de regularidad:

(H1) f es continuamente diferenciable.

Se dice que la dirección d de $\mathbb{R}^n$ es una **dirección de descenso** en x si

$$f(x + td) < f(x) \quad \text{para } t \text{ positivo suficientemente pequeño}$$

Una condición suficiente para que d sea una dirección de descenso es que se verifique

$$\nabla f(x)^t d < 0$$

Una gran parte de los algoritmos están basados en la elección de una dirección de descenso y de un paso t que asegure un decrecimiento suficiente de la función objetivo.

MÉTODO DE DESCENSO (ALGORITMO A1)

1. Elegir $x_0 \in \mathbb{R}^n$, $k = 0$.
2. Si $\nabla f(x_k) = 0$, PARAR.
3. Test de convergencia: (por ejemplo $\|\nabla f(x_k)\| < \epsilon$, ϵ pequeño dado)
   - si se verifica, PARAR
   - si no se verifica, continuar

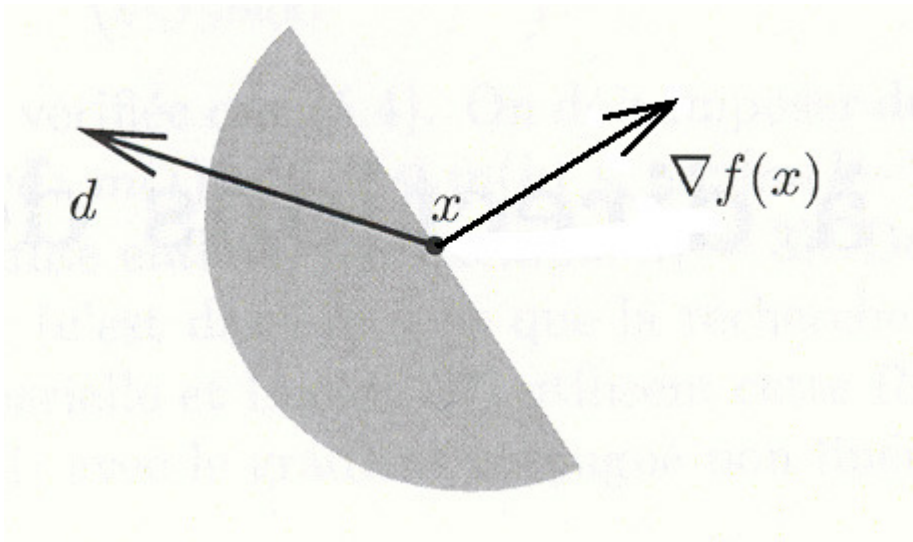


Figure 5: Direcciones de descenso en x

4. Elegir una dirección de descenso d_k.
5. Elegir un paso $t_k > 0$ verificando $f(x_k + t_k d_k) < f(x_k)$
6. Hacer $x_{k+1} = x_k + t_k d_k$, $k = k + 1$ e ir a 2

- *Hay métodos de descenso que no se pueden considerar dentro del algo-*

*ritmo A*1. *Este es el caso, por ejemplo de los métodos con región de confianza.*

- *Se obtienen algoritmos específicos precisando en A*1 *las reglas de elección de* d_k *y* t_k*.*
- *La convergencia será dudosa si* d_k *tiende a ser normal a* $\nabla f(x_k)$*. Para evitar eso puede pedirse la hipótesis:*

(H2) Existe $\alpha > 0$*, tal que* $\nabla f(x_k)^t d_k < -\alpha \|d_k\| \|\nabla f(x_k)\|$*.*

Reglas de búsqueda lineal

Vamos a suponer que ya se ha elegido una buena dirección de descenso (en las secciones siguientes veremos cómo hacer esto).

Nuestro problema es ahora **cómo calcular el paso**. Para ello conocemos:

- x, punto de partida de la búsqueda lineal
- d, dirección de descenso

- una función de mérito $q(t) = f(x + td)$ (notemos que $q'(0) = \nabla f(x)^t d < 0$ pues d es una dirección de descenso)

ESQUEMA GENERAL DE LA BÚSQUEDA LINEAL

Debe construirse un test con 3 salidas tal que, dado $t > 0$, responda si:

(a) t es adecuado

(b) t es demasiado grande

(c) t es demasiado pequeño

Este test se define según el valor de $q(t)$ y eventualmente de $q'(t)$.

EJEMPLO:

Un ejemplo muy simple, y **no muy bueno**, es el siguiente:
Se define el test con:

(a) si $q'(t) = 0$ (t parece adecuado pues es un punto estacionario)

(b) si $q'(t) > 0$ (t es demasiado grande pues el argumento que hace mínimo $q(t)$ debería ser más pequeño que t)

(c) si $q'(t) < 0$ (t es demasiado pequeño pues el argumento que hace mínimo $q(t)$ debería ser más grande que t)

Hay algo que no marcha en este ejemplo: la propiedad $q'(t) = 0$ no asegura en general que $q(t) < q(0)$, es decir, un paso adecuado según este test no tendría porque producir un descenso en el valor de f.

Llamaremos t_p a un paso t demasiado pequeño y t_g a uno demasiado grande. Para inicializar el algoritmo haremos $t_p = 0$, pero también se inicializa $t_g = 0$ (esto indica que aún no se ha encontrado un paso demasiado grande, podría iniciarse $t_g = +\infty$ pero esto carece de sentido en el ordenador).

Una vez elegido el test (a), (b), (c) el **algoritmo esquemático de búsqueda lineal** será:

Etapa 0 Se parte de $t > 0$ dado y se inicializan $t_p = 0$ y $t_g = 0$.

Etapa 1 Se prueba t:

- si t cumple (a), se termina.
- si t cumple (b), se hace $t_g = t$ y se va a la etapa 2.
- si t cumple (c), se hace $t_p = t$ y se va a la etapa 2.

Etapa 2

- Si no se ha encontrado ningún t_g se calcula un nuevo $t_+ > t_p$ (*EXTRAPOLACIÓN: supongamos que no disponemos de cota superior. Se quiere que* t_+ *sea claramente más grande que* t*, entonces hacemos* $t_+ = at$ *con* $a > 1$ *fijado.*). Se hace $t = t_+$.
- Si se tiene un t_g (i.e. $t_g \neq 0$) se calcula un nuevo $t \in (t_p, t_g)$ (*INTERPOLACIÓN: Si se tiene ya un* t_g *debe encontrarse un nuevo* t *netamente entre* t_p *y* t_g*. Lo más simple es hacer:*

$$t_+ = \frac{t_p + t_g}{2}$$

es decir una interpolación por dicotomía (la longitud de confianza se divide por 2).
Una alternativa más adecuada es utilizar una técnica de interpolación cúbica (ver F. BONNANS et al.).). Se hace $t = t_+$.

- Se vuelve a la etapa 1.

La elección del t inicial en la etapa 0 no concierne a la búsqueda lineal. Debe ser inicializado por el algoritmo de optimización.

La búsqueda lineal es entonces una sucesión de reducciones del intervalo de confianza $[t_p, t_g]$ eventualmente precedido de una sucesión de extrapolaciones mientras $t_g = 0$.

Vamos a dar algunas reglas para definir $(a), (b), (c)$.

Está claro que no es adecuado elegir un t cercano al óptimo (pues la búsqueda lineal sería muy larga) y realmente sólo estamos interesados en reducir el valor de f.

Las reglas que veremos son las de Wolfe, Goldstein-Price y Armijo.

Regla de Wolfe Es el método que parece más inteligente. Se eligen $0 < m_1 < m_2 < 1$ y los casos $(a), (b), (c)$ se definen como:

(a) si $q(t) \leq q(0) + m_1 t q'(0)$ (paso suficientemente pequeño para tener descenso) y $q'(t) \geq m_2 q'(0)$ (paso no demasiado pequeño, para que el algoritmo no se ralentice), entonces el paso es adecuado y se termina.

(b) si $q(t) > q(0) + m_1 t q'(0)$ entonces $t_g = t$ (el paso t es demasiado grande).

(c) si $q(t) \leq q(0) + m_1 t q'(0)$ y $q'(t) < m_2 q'(0)$ entonces $t_p = t$ (el paso t es demasiado pequeño).

La interpretación es la siguiente, se pide por una parte que f descienda suficientemente y por otra parte la derivada debe aumentar lo bastante para que x_{k+1} no sea excesivamente próximo a x_k (lo cual daría lugar a un método muy lento).

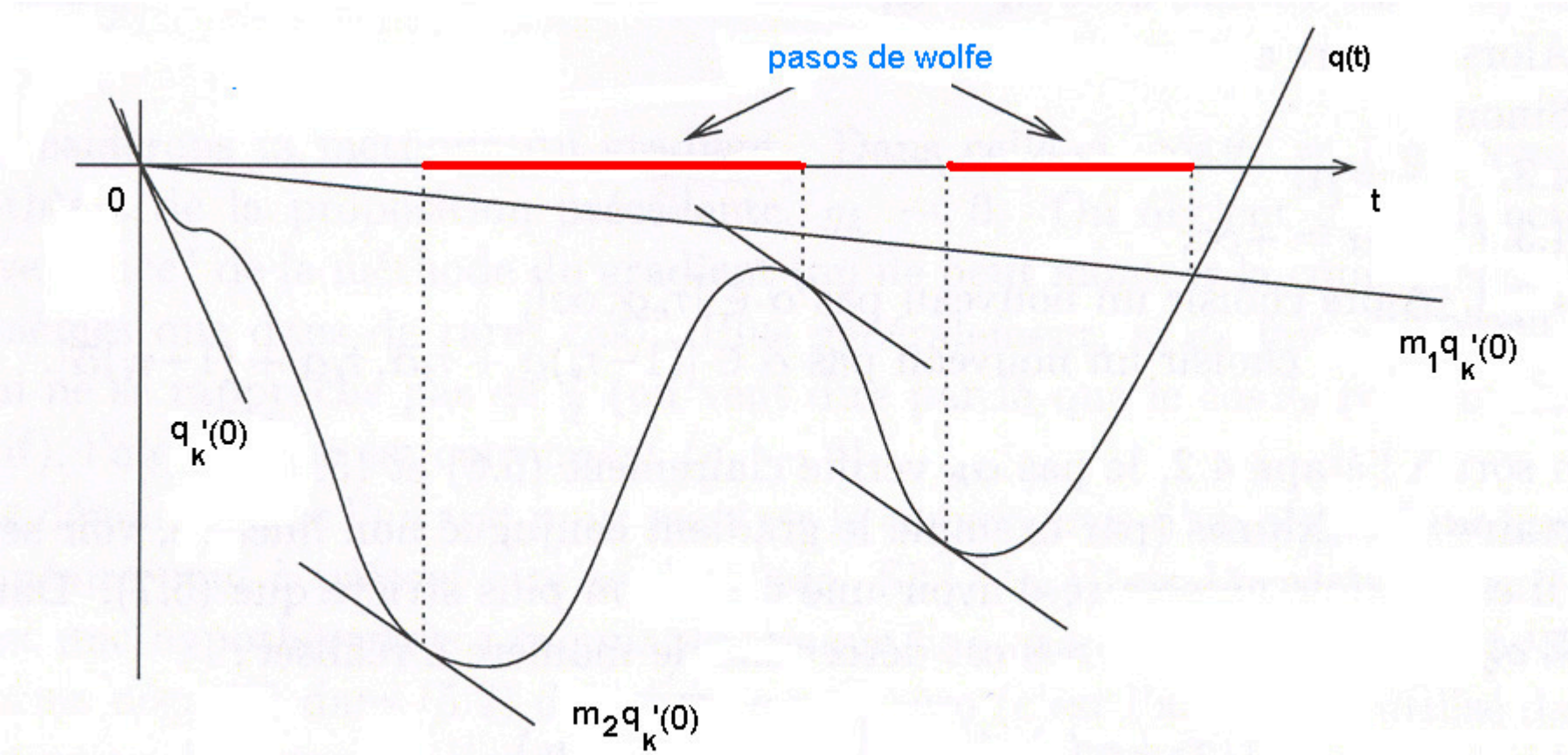


Figure 6: Regla de Wolfe

Regla de Goldstein-Price

Para funcionar la regla de Wolfe necesita conocer $q'(t)$, es decir conocer

$$\nabla f(x_k + td_k)$$

para cada ensayo de un paso t y esto puede ser **muy costoso** en la práctica.

La regla de Goldstein-Price es una regla alternativa que evita este problema. Se eligen $0 < m_1 < m_2 < 1$ y los casos $(a), (b), (c)$ se definen como:

(a) si $q(t) \leq q(0) + m_1 t q'(0)$ (paso suficientemente pequeño para tener descenso) y $q(t) \geq q(0) + m_2 t q'(0)$ (paso no demasiado pequeño, para que el algoritmo no se ralentice), entonces el paso es adecuado y se termina.

(b) si $q(t) > q(0) + m_1 t q'(0)$ entonces $t_g = t$ (el paso t es demasiado grande).

(c) si $q(t) < q(0) + m_2 t q'(0)$ entonces $t_p = t$ (el paso t es demasiado pequeño).

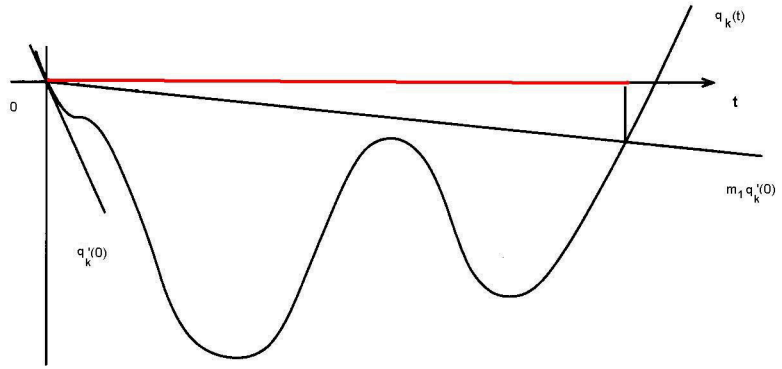


Figure 7: Regla de Armijo

Regla de Armijo Esta regla es un poco especial pues no declara ningún t como demasiado pequeño y de hecho no se extrapola nunca. Se elige $0 < m_1 < 1$ y los casos $(a), (b), (c)$ se definen como:

(a) si $q(t) \leq q(0) + m_1 t q'(0)$ (paso suficientemente pequeño para tener descenso), entonces el paso es adecuado y se termina.

(b) si $q(t) > q(0) + m_1 t q'(0)$ entonces $t_g = t$ (el paso t es demasiado grande).

(c) nunca

Otra escritura, más fácil de programar, de la regla de Armijo es la siguiente:

- *Se elige* $s > 0$, $\beta \in (0,1)$, $\epsilon \in (0,1)$.
- *Se hace* $p = 0$ *y se comprueba si:*

$$q(\beta^p s) \leq q(0) + \beta^p s \epsilon q'(0)$$

si se cumple se hace $t = \beta^p s$ *y si no se cumple se hace* $p = p + 1$ *y se repite la comprobación anterior.*

NOTA SOBRE LA ELECCIÓN DE LAS CONSTANTES:

Cualquiera que sea la regla que se utiliza conviene elegir el coeficiente de descenso $m_1 < 0.5$ y en la regla de Goldstein conviene elegir $m_2 > 0.5$

Algoritmo de gradiente

Es un método de descenso en el que se elige $d^k = -\nabla f(x_k)$.

- Parece interesante por sus simplicidad y por el hecho de que si se elige como producto escalar en $\mathbb{R}^n$ el producto euclídeo, la dirección de mayor descenso en x es $-\nabla f(x_k)$.

- Si f de clase C^2 y el método de gradiente con paso óptimo (conocido como **método de mayor descenso**) converge hacia un punto $\bar{x}$ que verifica las condiciones suficientes de optimalidad de segundo orden, entonces el método tiene una tasa de **convergencia lineal**.

- Si el condicionamiento de la matriz hessiana en el óptimo es muy grande **la convergencia es muy lenta**.

- La utilidad del algoritmo de gradiente es realmente servir de base a otros métodos más eficaces.

- Señalemos también que el método de gradiente con paso constante converge, para un paso **suficientemente pequeño**, bajo las condiciones:

  $\nabla f(x)$ *es lipschitziana* y $(\nabla f(x_1) - \nabla f(x_2))^t(x_1 - x_2) \geq \alpha\|x_1 - x_2\|^2$ *con* $\alpha > 0$.

Condicionamiento

Supongamos que se hace un cambio de variable lineal:

$$x(y) = Ay \quad \text{ó} \quad y(x) = A^{-1}x$$

y se define:

$$h(y) = f(x(y)) = f(Ay)$$

Minimizar h respecto a y equivale a minimizar f respecto a x en el sentido de que los óptimos son los mismos. Pero esta equivalencia es engañosa y falsa numéricamente. Aplicando la regla de la cadena el gradiente de h está dado por la fórmula:

$$\nabla h(y) = A^t \nabla f(Ay)$$

Entonces señalemos lo siguiente:

- partiendo de un x dado el método de gradiente aplicado a f genera el iterante siguiente de la forma:

$$x_+ = x - t\nabla f(x)$$

- partiendo de $y = A^{-1}x$, el método de gradiente aplicado a h genera el iterante siguiente de la forma:

$$y_+ = y - tA^t\nabla f(x)$$

  al que corresponde:

$$\tilde{x}_+ = A(y_+) = A(y - tA^t\nabla f(x)) = x - tAA^t\nabla f(x)$$

  es decir el efecto del cambio de variable viene a ser multiplicar las direcciones por AA^t (lo que cambia todo salvo en el caso de que A sea ortogonal).

Utilizando esto pueden encontrarse cambios de variables adecuados de manera que los métodos de gradiente se comporten lo mejor posible.

Esto es lo que se llama un **precondicionamiento**.

El precondicionamiento más simple es el diagonal que lo que hace es ajustar los factores de escala:

Se toma $y_i = \dfrac{x_i}{\bar{x}_i}$ para $i = 1, \ldots, n$ y se minimiza con respecto a la variable y.
Se tiene una gran ventaja en tomar como $\bar{x}_i$ una amplitud nominal de los x_i lo que hace que los y_i no tengan dimensión.

Método de Newton y variantes

La resolución de un problema de minimización sin restricciones implica en particular la resolución del sistema de n ecuaciones con n incógnitas:

$$\nabla f(x) = 0$$

Recíprocamente, si $\bar{x}$ verifica $\nabla f(\bar{x}) = 0$ y si la matriz $\nabla^2 f(\bar{x})$ es definida positiva, $\bar{x}$ es un mínimo local de f.
Puede entonces aplicarse el método de Newton para resolver el sistema no lineal $\nabla f(x) = 0$. Dado un iterante x el nuevo iterante x_+ es solución de:

$$\nabla f(x) + \nabla^2 f(x)(x_+ - x) = 0$$

entonces, si $\nabla^2 f(x)$ es inversible, x_+ está dado por:

$$x_+ = x - (\nabla^2 f(x))^{-1} \nabla f(x)$$

En concreto el algoritmo construye la sucesión $\{x_k\}$ definida de la manera siguiente:

Algoritmo de Newton

1. Elegir $x_0 \in \mathbb{R}^n$, $k = 0$.
2. Test de convergencia: (por ejemplo $\|\nabla f(x_k)\| < \epsilon$, ϵ pequeño dado)
   - si se verifica, PARAR
   - si no se verifica, continuar
3. Calcular la dirección de Newton d_k^N solución del sistema lineal:

$$\nabla^2 f(x_k) d_k^N = -\nabla f(x_k)$$

4. $x_{k+1} = x_k + d_k^N$, $k = k + 1$. Ir a 2.

- La gran ventaja del método de Newton es que converge muy deprisa:

  Si f es de clase C^2 y su hessiano es definido positivo en un entorno de $\bar{x}$ entonces la convergencia del método de Newton es superlineal. Si ademas f es de clase C^3 entonces la convergencia es cuadrática.

- Sin embargo esto no implica la convergencia global del método de Newton. De hecho los inconvenientes del método de Newton son bien conocidos:

  – en general diverge violentamente

– además, es necesario calcular el hessiano, y despues resolver un sistema, lo que es costoso y lento

Método de Newton discreto

En algunos casos el cálculo exacto de $\nabla^2 f(x)$ es imposible o demasiado costoso pero se puede obtener una aproximación, que llamaremos $\tilde{H}(x)$, por diferencias finitas sobre ∇f.

El algoritmo anterior se extiende sustituyendo $\nabla^2 f(x)$ por $\tilde{H}(x)$. Para evaluar $\tilde{H}(x)$ basta calcular:

$$\frac{\nabla f(x + \alpha e_i) - \nabla f(x)}{\alpha}, \quad i = 1, \ldots, n$$

donde e_i es el i-ésimo vector de una base y α es pequeño.

Sin embargo si n es grande, el cálculo de n gradientes es costoso. En ciertos casos, se pueden calcular los elementos de $\tilde{H}(x)$ en menos de n iteraciones eligiendo de manera astuta las direcciones e_i.

Factorización incompleta del hessiano

El método de Newton necesita, en cada iteración, la resolución del sistema lineal:

$$\nabla^2 f(x)d = -\nabla f(x)$$

Cuando n es grande esto puede presentar dificultades por culpa de:

1. *limitaciones de memoria*, pues el almacenamiento del hessiano necesita $\frac{n(n+1)}{2}$ posiciones.
2. *errores numéricos* en la resolución del sistema lineal por un método directo

Si el tamaño de memoria es insuficiente se puede efectuar una factorización incompleta de $\nabla^2 f(x)$.

Resolución incompleta de la ecuación de Newton

Si la resolución exacta de la ecuación de Newton es costoso, se puede contentar uno con el resultado obtenido despues de algunas iteraciones de la resolución de esta ecuación por un método iterativo (el método de gradiente conjugado, por ejemplo).

Si el punto de partida para el gradiente conjugado es $d_1 = -\nabla f(x)$, las direcciones siguientes $\{d_1, \ldots, d_i\}$ son direcciones de descenso, intermedias, entre $-\nabla f(x)$ y la dirección de Newton.

Los métodos vistos hasta ahora son localmente convergentes.

Vamos a ver como definir métodos globalmente convergentes que tengan las propiedades del método de Newton en un entorno del punto.

Presentaremos un **método con región de confianza** y los **métodos cuasi-Newton**.

Método de Newton con región de confianza

Sea x_0 dado, conociendo $\nabla f(x_0)$ y $\nabla^2 f(x_0)$ se tiene una aproximación cuadrática de f:

$$q(x) = f(x_0) + \nabla f(x_0)^t(x - x_0) + \frac{1}{2}(x - x_0)^t \nabla^2 f(x_0)(x - x_0)$$

que realiza una buena aproximación de f en un entorno de x_0.

Si $\delta > 0$ es pequeño , y si $\nabla f(x_0) \neq 0$, la solución $y \in \mathbb{R}^n$ del problema:

$$\min q(y), \quad \text{sujeto a } \|y - x_0\|^2 \leq \delta$$

verificará $f(y) < f(x_0)$.

Además si δ es grande y si $\nabla^2 f(x_0)$ es definido positivo

$$y - x_0 = -(\nabla^2 f(x_0))^{-1} \nabla f(x_0)$$

que es la dirección de Newton.

Para ajustar δ se puede comparar el decrecimiento previsto por el modelo cuadrático con el decrecimiento real de f.

Un ejemplo sencillo es el siguiente:

Algoritmo con región de confianza del modelo cuadrático

1. Elegir $x_0 \in \mathbb{R}^n$, $k = 0$ y δ_0 iniciales. Sean η_1, η_2, γ_1, γ_2 tales que $0 < \eta_1 < \eta_2 < 1$, $0 < \gamma_1 < 1 < \gamma_2$.
2. Test de convergencia: (por ejemplo $\|\nabla f(x_k)\| < \epsilon$, ϵ pequeño dado)

- si se verifica, PARAR
- si no se verifica, continuar

3. Definir $q_k(x) = f(x_k) + \nabla f(x_k)^t(x - x_k) + \frac{1}{2}(x - x_k)^t \nabla^2 f(x_k)(x - x_k)$

4. Calcular y_k solución de $\min q_k(y)$, sujeto a $\|y - x_k\|^2 \leq \delta_k$ y $R_k = \dfrac{f(x_k) - f(y_k)}{f(x_k) - q_k(y_k)}$.

5. Si $R_k < \eta_1$ (decrecimiento real demasiado pequeño con respecto al decrecimiento previsto por el modelo cuadrático), actualizar $\delta_k = \gamma_1 \delta_k$ e ir a 4

6. Si $R_k > \eta_1$, actualizar $x_{k+1} = y_k$.

7. Si, además $R_k > \eta_2$, entonces el modelo es bueno y la región puede expandirse, y por tanto se hace: $\delta_k = \gamma_2 \delta_k$

8. $\delta_{k+1} = \delta_k$, $k = k + 1$. Ir a 2.

Métodos cuasi-Newton

Ya se ha comentado que la resolución del problema de optimización:

$$(\mathcal{OSR}) \left\{ \min_{x \in \mathbb{R}^n} f(x) \right.$$

se puede reducir a la resolución de la ecuación $\nabla f(x) = 0$, para lo cual es útil disponer del hessiano $\nabla^2 f(x)$ o de una aproximación.

Fijemos x y supongamos conocida una aproximación de $\nabla^2 f(x)$, que denotaremos por B. Sea $\hat{x} \in \mathbb{R}^n$ un punto próximo a x y supongamos conocidos $\nabla f(x)$ y $\nabla f(\hat{x})$.

¿Permite esta información deducir de B una mejor aproximación $\hat{B}$ del hessiano $\nabla^2 f(x)$?

Si definimos $s = \hat{x} - x$ e $y = \nabla f(\hat{x}) - \nabla f(x)$ es natural imponer como condición sobre $\hat{B}$:

$$\hat{B}s = y.$$

Esta ecuación se llama *ecuación de cuasi-Newton.*

Se verán diferentes fórmulas del tipo $\hat{B} = \Phi(B, y, s)$ compatibles con la condición anterior.

Esto permitirá desarrollar los **algoritmos de métrica variable** del tipo siguiente:

1. Elegir $x_0 \in \mathbb{R}^n$ y B_0 matriz $n \times n$ definida positiva ; $k = 0$.
2. Test de convergencia: (por ejemplo $\|\nabla f(x_k)\| < \epsilon$, ϵ pequeño dado)
   - si se verifica, PARAR
   - si no se verifica, continuar
3. Calcular la dirección de búsqueda d_k como solución del sistema lineal:

$$(B_k)d_k = -\nabla f(x_k)$$

4. Calcular $t_k > 0$ por una regla de búsqueda lineal adecuada.
   Hacer $x_{k+1} = x_k + t_k d_k$.
5. Calcular $B_{k+1} = \Phi(B_k, y_k, s_k)$, con $s_k = x_{k+1} - x_k$, e $y_k = \nabla f(x_{k+1}) - \nabla f(x_k)$. Hacer $k = k + 1$. Ir a 2.

Fórmula de Broyden y fórmula simétrica de rango 1

Una primera idea es decir que si $z \in \mathbb{R}^n$ verifica $s^t z = 0$, no hay ninguna información sobre $\nabla^2 f(x)z$. Parece entonces razonable imponer:

$$\hat{B}z = Bz, \qquad \forall z \in \mathbb{R}^n \text{ tal que } s^t z = 0.$$

Esta relación y la ecuación de cuasi-Newton determinan de manera única $\hat{B}$:

$$\hat{B} = B + \frac{(y - Bs)s^t}{s^t s},$$

conocida como *fórmula de Broyden*.

Señalemos que la $\hat{B}$ así obtenida no es en general simétrica.

La fórmula simétrica de rango 1 (obtenida con una corrección de rango 1 de la fórmula anterior) es:

$$\hat{B} = B + \frac{(y - Bs)(y - Bs)^t}{(y - Bs)^t s}.$$

Fórmula de Powell simétrica Broyden (PSB)

Deseamos encontrar una matriz $\hat{B}$ simétrica, tan próxima como sea posible a B, que verifique la ecuación de cuasi-Newton. Una formulación posible del problema es la siguiente:

$$(\mathcal{PB}) \begin{cases} \min \|\tilde{B} - B\| \\ \tilde{B} \text{ simétrica y tal que } \tilde{B}s = y. \end{cases}$$

La solución de este problema depende de la norma matricial elegida. Si se elige la norma de Frobenius:

$$\|B\|_F = [tr(B^t B)]^{1/2} = [\sum_{i=1}^{n} \sum_{j=1}^{n} B_{ij}^2]^{1/2}$$

la única solución del problema viene dada por el siguiente resultado:

La solución única del problema $(\mathcal{PB})$ asociada a la norma de Frobenius viene dada por la **fórmula PSB**:

$$\hat{B}_{PSB} = B + \frac{(y - Bs)s^t + s(y - Bs)^t}{s^t s} - \frac{s^t(y - Bs)}{(s^t s)^2} ss^t.$$

Observación: La norma de Frobenius no siempre está bien adaptada al problema. Por ejemplo, si $n = 2$ y

$$H(x) = \nabla^2 f(x) = \begin{pmatrix} 10^4 & 0 \\ 0 & 1 \end{pmatrix},$$

la matriz:

$$B = \begin{pmatrix} 10^4 & 0 \\ 0 & 10^{-4} \end{pmatrix}$$

es mucho más próxima a $\nabla^2 f(x)$ que la identidad y, sin embargo, también conduce a un problema mal condicionado, ya que:

$$B^{-1}H(x) = \begin{pmatrix} 1 & 0 \\ 0 & 10^4 \end{pmatrix}.$$

Fórmula de Davidon-Fletcher-Powell (DFP)

Sea W una matriz simétrica, definida positiva. Se introduce para el problema $(\mathcal{PB})$ la norma:

$$\|B\|_{W,F} = \|WBW\|_F$$

Es interesante elegir una matriz W próxima a $H(\bar{x})^{-1/2}$.

Por supuesto, $\nabla^2 f(\bar{x})$ es desconocido , por tanto es lógico imponer:

$$W^{-2}s = y.$$

Esta condición basta para determinar $\hat{B}$ de manera única:

Sea W una matriz simétrica, definida positiva verificando $W^{-2}s = y$. La solución del problema $(\mathcal{PB})$ asociada a la norma matricial $\|\,.\,\|_{W,F}$ viene dada por la **fórmula DFP**:

$$\hat{B}_{DFP} = B + \frac{(y - Bs)y^t + y(y - Bs)^t}{s^t y} - \frac{s^t(y - Bs)}{(s^t y)^2} yy^t.$$

Fórmula de Broyden-Fletcher-Goldfarb-Shanno (BFGS)

Como la iteración de un método de métrica variable se escribe:

$$x_{k+1} = x_k - t_k(B_k)^{-1}\nabla f(x_k)$$

se puede, en lugar de aproximar $\nabla^2 f(x)$ por B, aproximar $(\nabla^2 f(x))^{-1}$ por una matriz M.

Esta matriz debe cumplir la *ecuación de cuasi-Newton inversa*:

$$My = s.$$

Todos los métodos descritos anteriormente para la aproximación del hessiano tienen su análogo para la aproximación de la inversa del hessiano: basta reemplazar B por M y permutar s e y.

En particular, el análogo a DFP, llamado **BFGS**, está dado por la fórmula:

$$\hat{M}_{BFGS} = M + \frac{(s - My)s^t + s(s - My)^t}{s^t y} - \frac{y^t(s - My)}{(s^t y)^2} ss^t.$$

Sea $B = M^{-1}$ y $\hat{B} = \hat{M}^{-1}$. Puede comprobarse que $\hat{B}$ se obtiene de B por la relación:

$$\hat{B}_{BFGS} = B + \frac{yy^t}{s^t y} - \frac{Bss^t B}{s^t Bs}.$$

Damos ahora un resultado sobre la conservación del carácter definido positivo por las fórmulas DFP y BFGS:

Sea B una matriz simétrica, definida positiva. Entonces las matrices $\hat{B}_{DFP}$ y $\hat{B}_{BFGS}$ son definidas positivas si y sólo si $s^t y > 0$.

Convergencia de los métodos cuasi-Newton

Hemos visto que la conservación del carácter definido positivo necesita $(s_k)^t y_k > 0$, es decir, $(\nabla f(x_{k+1}) - \nabla f(x_k))^t d_k > 0$. Esta condición se verifica automáticamente para una búsqueda lineal de tipo Wolfe, que parece bien adaptada a los métodos de cuasi-Newton:

Sea f una función convexa de clase C^2 tal que el conjunto:

$$S = \{x \ / \ f(x) \leq f(x_0)\}$$

está acotado y sea B_0 una matriz definida positiva. El algoritmo de métrica variable $M_k = (B_k)^{-1}$, donde B_k es calculado por la fórmula de $BFGS$, y

donde el paso t_k se fija por las condiciones de Wolfe, converge hacia una solución $\bar{x}$ del problema.

Si además $\nabla^2 f(\bar{x})$ es definida positiva y si $\nabla^2 f(x)$ es localmente lipschitziana, entonces $x_k \longrightarrow \bar{x}$ superlinealmente.

Métodos de gradiente conjugado

Gradiente conjugado para un criterio cuadrático

Se considera el problema

$$(\mathcal{PC}) \quad \left\{ \min_{x \in \mathbb{R}^n} f(x) = \frac{1}{2} x^t A x - b^t x \right.$$

donde A es una matriz $n \times n$, simétrica, definida positiva y b es un vector de $\mathbb{R}^n$.

Suponemos conocidos n vectores $d_1, \ldots, d_n$, linealmente independientes y conjugados, es decir tales que:

$$d_k^t A d_j = 0, \qquad \text{si } k \neq j$$

Expresado en la base de los $\{d_j\}$ el problema se transforma en

$$\left\{ \min_{\tilde{x}\in\mathbb{R}^n} \sum_{k=1}^{n} \frac{1}{2} d_k^t A d_k \tilde{x}_k^2 - \tilde{b}^t \tilde{x} \right.$$

donde $\tilde{x}_k$ es la k-ésima componente de x sobre la base de los $\{d_j\}$ y donde $\tilde{b}$ es un vector de $\mathbb{R}^n$ tal que $\tilde{b}_k = b^t d_k$.

Entonces el problema $(\mathcal{PC})$ se descompone en n problemas independientes triviales.

Algoritmo de direcciones conjugadas

1. Elegir $\{v_1, \ldots, v_n\}$ vectores independientes de $\mathbb{R}^n$, x_1 dado, $d_1 = v_1$, $k = 1$.
2. Calcular $t_k = \arg\min_{t\geq 0} f(x_k + t d_k)$.
   Hacer $x_{k+1} = x_k + t_k d_k$
3. Si $k + 1 = n + 1$ PARAR. En otro caso, calcular:

$$d_{k+1} = v_{k+1} + \sum_{j=1}^{k} c_{k+1j} d_j$$

con

$$c_{k+1j} = -\frac{v_{k+1}^t(\nabla f(x_{j+1}) - \nabla f(x_j))}{d_j^t(\nabla f(x_{j+1}) - \nabla f(x_j))}, \quad j = 1, \ldots, k$$

Hacer $k = k + 1$ y volver a 2.

Señalemos que el cálculo de las direcciones conjugadas no necesita el conocimiento explícito de A. Basta poder calcular el gradiente en un punto.

El caso particular del algoritmo anterior cuando $v_k = -\nabla f(x_k)$ se llama **método de gradiente conjugado**. Naturalmente si $\nabla f(x_k) = 0$ el algoritmo se para. Si $\nabla f(x_k) \neq 0$ se puede señalar que se ha minimizado f en el hiperplano generado por las $k - 1$ direcciones d_j que es idéntico al hiperplano generado por las $k - 1$ direcciones $\nabla f(x_j)$. El algoritmo de gradiente conjugado aplicado a las funciones cuadráticas está dado por:

Algoritmo de gradiente conjugado (CG1)

1. Elegir x_1, $d_1 = -\nabla f(x_1)$. Si $d_1 = 0$, PARAR. $k = 1$.

2. Calcular $t_k = \arg\min_{t\geq 0} f(x_k + td_k)$.

   Hacer $x_{k+1} = x_k + t_k d_k$

3. Si $k+1 = n+1$ ó $\nabla f(x_k) = 0$, PARAR. En otro caso, calcular:

$$d_{k+1} = -\nabla f(x_{k+1}) + \frac{\|\nabla f(x_{k+1})\|^2}{\|\nabla f(x_k)\|^2} d_k$$

   Hacer $k = k+1$ y volver a 2.

Como la función es cuadrática el **paso óptimo** se calcula como sigue: Se define la función de mérito

$$q(t) = f(x+td) = f(x) + t\nabla f(x)^t d + \frac{t^2}{2} d^t A d$$

Para minimizar $q(t)$ basta entonces calcular $d^t A d$ pues $q(t)$ alcanza su valor mínimo en

$$t = -\frac{\nabla f(x)^t d}{d^t A d}$$

NOTA:

La convergencia del método de gradiente conjugado es mucho mejor que la del método de gradiente con paso óptimo.

Caso de criterios no cuadráticos

El algoritmo de gradiente conjugado puede extenderse a una función objetivo no cuadrático. Existen varias maneras, equivalentes en el caso cuadrático, de definir la dirección de descenso. Las dos más conocidas son:

Método de Fletcher-Reeves

$$d_k = -\nabla f(x_k) + \frac{\|\nabla f(x_k)\|^2}{\|\nabla f(x_{k-1})\|^2} d_{k-1}$$

Método de Polak-Ribière

$$d_k = -\nabla f(x_k) + \frac{\nabla f(x_k)^t(\nabla f(x_k) - \nabla f(x_{k-1}))}{\|\nabla f(x_{k-1})\|^2} d_{k-1}$$

En este caso es muy costoso calcular un paso t óptimo. Como la condición de conjugación necesita $\nabla f(x_{k+1})^t d_k = 0$, se tomará en la búsqueda lineal una condición del tipo:

$$|\nabla f(x_k + td_k)^t d_k| \leq \epsilon_1 |\nabla f(x_k)^t d_k|$$

$$f(x_k + td_k) \leq f(x_k) + \epsilon_2 t \nabla f(x_k)^t d_k$$

con $\epsilon_2 = 10^{-4}$ y $\epsilon_1 \in [0.1, 0.9]$.

Con estas condiciones en la búsqueda lineal se tiene el siguiente resultado de convergencia:

Si el hessiano $\nabla^2 f(x)$ es definido positivo y sus valores propios están en $[\lambda_m, \lambda_M]$ con $0 < \lambda_m < \lambda_M < +\infty$, entonces el algoritmo de Polak-Ribière

asociado a una búsqueda lineal exacta con o sin redireccionamiento verifica que existe $\alpha > 0$ *tal que:*

$$\nabla f(x_k)^t d_k < -\alpha \|d_k\| \|\nabla f(x_k)\|$$

En consecuencia la sucesión generada por el algoritmo converge hacia la solución $\bar{x}$ *del problema.*

El algoritmo de Polak-Ribière tiene mejores propiedades numéricas que el de Fletcher-Reeves.

Métodos cuasi-Newton con memoria limitada

Vamos a analizar una familia de métodos generalizando, de una manera diferente, el algoritmo de gradiente conjugado.

El método de cuasi-Newton más usado, llamado BFGS, es de la forma:

$$x_{k+1} = x_k + t_k d_k$$

$$d_k = -B_k^{-1}\nabla f(x_k)$$

siendo la matriz B_k una estimación del hessiano de f en x_k obtenida por la fórmula iterativa:

$$(\mathcal{I}) \quad B_k = \Phi(B_{k-1}, y_{k-1}, s_{k-1})$$

con $s_{k-1} = x_k - x_{k-1}$, $y_{k-1} = \nabla f(x_k) - \nabla f(x_{k-1})$ y

$$\Phi(B, y, s) = B + \frac{yy^t}{s^t y} - \frac{Bss^tB}{s^tBs}$$

- Un método **cuasi-Newton con memoria limitada** se obtiene reemplazando $(\mathcal{I})$ por

  $$B_k = \Phi(D_{k-1}, y_{k-1}, s_{k-1})$$

  donde D_{k-1} es una matriz de complejidad reducida tal que el cálculo de B_k sea rápido y no necesite mucho almacenamiento.

- Por ejemplo, si D_{k-1} es la identidad (el **método se dice sin memoria**) se obtiene:

$$B_k = I + \frac{y_{k-1} y_{k-1}^t}{s_{k-1}^t y_{k-1}} - \frac{s_{k-1} s_{k-1}^t}{s_{k-1}^t s_{k-1}}$$

De esta manera el cálculo de la nueva dirección sólo necesita operaciones sencillas y el almacenamiento de dos vectores. Así, el método es aplicable a problemas de gran tamaño.

- Estos métodos están estrechamente relacionados con los algoritmos de gradiente conjugado:

  Por ejemplo, si D_{k-1} es la identidad y la búsqueda lineal es exacta, el algoritmo sin memoria es el algoritmo de gradiente conjugado de Polak-Ribiére.

- La ventaja de los algoritmos cuasi-Newton sin memoria respecto al gradiente conjugado es que no necesitan una regla de búsqueda lineal costosa (se utilizan con la regla de Wolfe que asegura que B_k es definida positiva).

Mínimos cuadrados no lineales

Vamos a exponer algunos algoritmos de resolución del problema de *mínimos cuadrados no lineales*:

$$(\mathcal{MC}) \left\{ \begin{array}{l} \min \\ x \in \mathbb{R}^n \end{array} \right. f(x) = r(x)^t r(x)/2$$

siendo $r : \mathbb{R}^n \to \mathbb{R}^p$, $r(x) = (r_1(x), \ldots, r_p(x))^t$.

EJEMPLO:

Se intenta encontrar una curva que ajuste algunas medidas $y_1, \ldots, y_m$ de una señal tomadas en los tiempos $t_1, \ldots, t_m$. Desde los datos y el conocimiento de la aplicación, se deduce que la señal tiene un comportamiento exponencial y oscilatorio, y se elige modelarlo por la función:

$$\Phi(t, x) = x_1 + x_2 e^{-(x_3 - t)^2 / x_4} + x_5 \cos(x_6 t)$$

Los números reales x_i, $i = 1, \ldots, 6$ son los parámetros del modelo. Se desea seleccionarlos de manera que los valores del modelo $\Phi(t_j, x)$ ajusten los datos

observados y_j tanto como sea posible. Para establecer el objetivo como un problema de optimización, se definen los residuos

$$r_j(x) = y_j - \Phi(t_j, x), \quad j = 1, \ldots, m$$

que miden la discrepancia entre el modelo y los datos observados. La estimación de x se obtendrá resolviendo el problema:

Recordemos que:

$$\nabla f(x) = \sum_{i=1}^{p} r_i(x) \nabla r_i(x)$$

$$\nabla^2 f(x) = \sum_{i=1}^{p} \nabla r_i(x) \nabla r_i(x)^t + \sum_{i=1}^{p} r_i(x) \nabla^2 r_i(x)$$

Vamos a introducir, para simplificar la notación, la matriz $p \times n$

$$A(x) = \begin{pmatrix} \nabla r_1(x)^t \\ \vdots \\ \nabla r_p(x)^t \end{pmatrix}$$

Un cálculo elemental da:

$$\nabla f(x) = A(x)^t r(x)$$

$$\nabla^2 f(x) = A(x)^t A(x) + \sum_{i=1}^{p} r_i(x) \nabla^2 r_i(x)$$

Método de Gauss-Newton

Un medio simple de obtener un modelo local (alrededor de un punto x) de f es linealizar la función de entrada-salida $r(x)$. El modelo de f así obtenido es cuadrático y su minimización permite el cálculo de un nuevo punto.

En concreto el algoritmo construye la sucesión $\{x_k\}$ definida de la manera siguiente:

Algoritmo de Gauss-Newton

1. Elegir $x_0 \in \mathbb{R}^n$, $k = 0$.
2. Test de convergencia: (por ejemplo $\|\nabla f(x_k)\| < \epsilon$, ϵ pequeño dado)
   - si se verifica, PARAR

- si no se verifica, continuar

3. Calcular la dirección de Gauss-Newton d_k^{GN} solución del problema:

$$\begin{cases} \min\limits_{d\in\mathbb{R}^n} \Phi_k(d) = \dfrac{1}{2}\sum\limits_{i=1}^{p}(r_i(x_k) + \nabla r_i(x_k)^t d)^2 = \dfrac{1}{2}\|A(x_k)d + r(x_k)\|^2 \end{cases}$$

4. $x_{k+1} = x_k + d_k^{GN}$, $k = k + 1$. Ir a 2.

- Cada iteración del algoritmo de GN implica entonces la resolución de un problema de mínimos cuadrados lineales que tendrá una solución única si y sólo si se verifica la hipótesis siguiente:

$$A(x_k) \textit{ es inyectiva.}$$

- Señalemos que si $p = n$ y si $A(x_k)$ es inyectiva (por tanto inversible) entonces d_k^{GN} es solución de:

$$r(x_k) + A(x_k)d_k = 0$$

- El método de Gauss-Newton se reduce entonces al método de Newton aplicado a la ecuación $r(x) = 0$. Esto nos indica que la convergencia del método de Gauss-Newton será como mucho local.
- Si el residuo $r(\bar{x})$ es pequeño (errores de modelado y de medidas débiles) y si el punto de partida es bueno, el método de Gauss-Newton converge rápidamente.

Vamos a ver tres procedimientos que permiten globalizar el algoritmo de Gauss-Newton.

Métodos con búsqueda lineal

Consisten simplemente en efectuar una búsqueda lineal en la dirección de Gauss-Newton.

Algoritmo de Gauss-Newton con búsqueda lineal

1. Elegir $x_0 \in \mathbb{R}^n$, $k = 0$.

2. Test de convergencia: (por ejemplo $\|\nabla f(x_k)\| < \epsilon$, ϵ pequeño dado)

   - si se verifica, PARAR
   - si no se verifica, continuar

3. Calcular la dirección de Gauss-Newton d_k^{GN}.

4. Búsqueda lineal: Calcular $t_k > 0$ tal que

$$f(x_k + t_k d_k^{GN}) < f(x_k)$$

5. $x_{k+1} = x_k + t_k d_k^{GN}$, $k = k + 1$. Ir a 2.

El algoritmo tiene sentido si d_k^{GN} es una dirección de descenso de f. Si $A(x_k)$ es inyectiva esto es cierto pues:

$$\nabla f(x_k)^t d_k^{GN} = -\|A(x_k) d_k^{GN}\|^2$$

Método de Levenberg-Marquardt

Es un caso particular de los métodos de región de confianza. El principio del algoritmo es entonces el siguiente:

Algoritmo de Levenberg-Marquardt

1. Elegir $x_0 \in \mathbb{R}^n$, $k = 0$.
2. Test de convergencia: (por ejemplo $\|\nabla f(x_k)\| < \epsilon$, ϵ pequeño dado)
   - si se verifica, PARAR
   - si no se verifica, continuar
3. Calcular d_k^{LM} solución del problema:

$$\begin{cases} \min\limits_{d \in \mathbb{R}^n} \ \Phi_k(d) \ ; \ \dfrac{1}{2}\|d\|^2 \leq \delta_k \end{cases}$$

4. Si $f(x_k + d_k^{LM}) < f(x^k)$ se verifica hacer
   $x_{k+1} = x_k + d_k^{LM}$, si no $x_{k+1} = x_k$.
5. Actualizar δ_k. Hacer $k = k + 1$. Ir a 2.

Aquí Φ_k es el modelo de Gauss-Newton. La puesta al día de δ_k se hace como en el caso general de los *métodos con región de confianza.* También se puede obtener convergencia global imponiendo hipótesis sobre $A(x_k)$.

Métodos penalizando el desplazamiento

El algoritmo es idéntico al de Levenberg-Marquardt pero la etapa 1 se reemplaza por:

Calcular d_k^P solución de

$$\begin{cases} \min\limits_{d\in\mathbb{R}^n} \ \Phi_k(d) + \dfrac{\lambda_k}{2}\|d\|^2 \end{cases}$$

donde el coeficiente de penalización λ^k se aumenta si $x_k + d_k^P$ no es sensiblemente mejor que x_k, en otro caso se disminuye.

Los problemas de cálculo de d_k^{LM} y de d_k^P están estrechamente ligados.

En efecto, como estos problemas son convexos, d_k^{LM} y d_k^P están caracterizados por los sistemas de optimalidad:

$$A(x_k)^t A(x_k) d_k^{LM} + \mu d_k^{LM} = -A(x_k)^t r(x_k)$$

$$\mu \geq 0 \ , \quad \mu \, (\frac{1}{2}\|d_k^{LM}\|^2 - \delta_k) = 0$$

y

$$A(x_k)^t A(x_k) d_k^P + \lambda_k d_k^P = -A(x_k)^t r(x_k)$$

Entonces d_k^{LM} es solución del problema penalizado si $\lambda_k = \mu$ y, recíprocamente, si se toma $\eta = \frac{1}{2}\|d_k^P\|^2$, d_k^P es solución del problema:

$$\begin{cases} \min\limits_{d \in \mathbb{R}^n} \ \Phi_k(d) \ ; \ \frac{1}{2}\|d\|^2 \leq \eta \end{cases}$$

En la práctica es cómodo calcular d_k^{LM} evaluando d_k^P para diferentes valores del parámetro λ_k.

Métodos de cuasi-Newton adaptados

Recordemos que el hessiano del criterio $f(x) := \frac{1}{2}\sum_{i=1}^{p} r_i(x)^2$ es la matriz:

$$\nabla^2 f(x) = A(x)^t A(x) + \sum_{i=1}^{p} r_i(x)\nabla^2 r_i(x)$$

Sólo el segundo término del hessiano utiliza las derivadas segundas de r.

Los algoritmos de cuasi-Newton especializados para los problemas de mínimos cuadrados tienen, en general, como principio construir una sucesión de matrices $\{B^k\}$ que sean una aproximación del término $\Sigma_{i=1}^{p} r_i(x)\nabla^2 r_i(x)$.

El esquema general de estos algoritmos será entonces el siguiente:

Algoritmo de cuasi-Newton para mínimos cuadrados

1. Elegir $x_0 \in \mathbb{R}^n$ y B_0 matriz $n \times n$ simétrica ; $k = 0$.
2. Test de convergencia: (por ejemplo $\|\nabla f(x_k)\| < \epsilon$, ϵ pequeño dado)

- si se verifica, PARAR
- si no se verifica, continuar

3. Calcular d_k solución de

$$(A(x_k)^t A(x_k) + B_k)d_k = -\nabla f(x_k)$$

4. Calcular $t_k > 0$ tal que

$$f(x_k + t_k d_k) < f(x_k)$$

$$x_{k+1} = x_k + t_k d_k$$

5. Calcular B_{k+1} , $k = k + 1$. Ir a 1.

- Es lógico imponer a B_k que sea simétrica pues $\Sigma_{i=1}^{p} r_i(x_k)\nabla^2 r_i(x_k)$ lo es.
- Por el contrario B_k no tiene porque ser definida positiva.

Ecuaciones de cuasi-Newton

Tomemos $s_k = x_{k+1} - x_k$. El problema es definir un vector y^k tal que

$$\sum_{i=1}^{p} r_i(x_{k+1})\nabla^2 r_i(x_{k+1})s_k = y_k + o(s_k)$$

Se podrá entonces imponer a B_{k+1} que verifique una ecuación de cuasi-Newton:

$$B_{k+1}s_k = y_k$$

En lo que sigue se dan las principales elecciones de y_k (notaremos $A_{k+1} = A(x_{k+1}), \ldots$)

Se tiene $\Sigma_{i=1}^{p} r_i(x_{k+1})\nabla^2 r_i(x_{k+1})s_k = y_k + o(s_k)$ *si se toma como* y_k *uno de los dos vectores siguientes:*

1. $y_k = A_{k+1}^t r_{k+1} - A_k^t r_k - (A_{k+1})^t A_{k+1}s_k$
2. $y_k = (A_{k+1} - A_k)^t r_{k+1}$

Escalado y cálculo efectivo de B_k

Las fórmulas usuales de cuasi-Newon consisten en una modificación de rango 1 ó 2 de la matriz y no pueden tener en cuenta un cambio considerable en el tamaño del residuo. Un método de cuasi-Newton con escalado es del tipo siguiente: en cada iteración k, se calcula $\alpha_k \in \mathbb{R}$ coeficiente de escalado y se hace $\hat{B}_k = \alpha_k B_k$. Después se aplica una fórmula de cuasi-Newton del tipo:

$$B_{k+1} = \Phi(\hat{B}_k, y_k, s_k)$$

Entre las fórmulas más utilizadas para los problemas de mínimos cuadrados es la fórmula de rango 1 (debida a Wolfe):

$$\Phi(B, y, s) = B + \frac{(y - Bs)(y - Bs)^t}{(y - Bs)^t s}$$

El escalado se basa en el tamaño de $r(x_{k+1})$. Se han sugerido las siguientes elecciones:

$$\alpha_k = \frac{(r_{k+1})^t r_k}{(r_k)^t r_k}$$

$$\alpha_k = \frac{\|r_k\|_2}{\|r_{k-1}\|_2}$$

$$\alpha_k = \min\{1, \frac{(s_k)^t y_k}{(s_k)^t B_k s_k}\}$$

Bibliografía básica:

- D. BERTSEKAS : *Nonlinear Programming.* 2nd ed., Athena Scientific 1999.

- F. BONNANS - J. Ch. GILBERT - C. LEMARECHAL - C. SAGASTIZABAL : *Numerical Optimization: Theoretical and Practical Aspects.* 2nd ed., Springer, 2006.

- R. FLETCHER : *Practical Methods of Optimization.* 2nd ed., Wiley, 2006.

- D.G. LUENBERGER : *Programación lineal y no lineal.* Addison-Wesley Iberoamericana, 1989.

• J. NOCEDAL - S.J. WRIGHT : *Numerical Optimization.* 2nd ed., Springer, 2006.