Optimización no restringida

Gamaliel Moreno

Maestría en Ciencias del Procesamiento de la Información

Ene-Julio, 2021

Métodos Cuasi-Newton

En los 50's W.C. Davidon, un físico del Argonne National Laboratory. estaba utilizando el método de descenso para realizar un cálculo de optimización largo. En ese momento, las computadoras no eran muy estables, y para frustración de Davidon, el sistema informático siempre estaba bloqueado antes de que se completara el cálculo. Así que Davidon decidió encontrar una manera de acelerar la iteración. El algoritmo que desarrolló, el primer algoritmo de Quasi-Newton: resultó ser una de las ideas más revolucionarias en términos de mejora no lineal. Pronto, Fletcher y Powell demostraron que el nuevo algoritmo era mucho más rápido y más confiable que los otros métodos existentes, y este progreso dramático transformó la optimización no lineal. Durante los siguientes veinte años, se propusieron numerosas variantes y se dedicaron cientos de artículos a su estudio. Una interesante ironía histórica es que el artículo de Davidon no fue aceptado para su publicación; permaneció como un informe técnico durante más de treinta años hasta que apareció en el primer número del SIAM Journal on Optimization en 1991.

Métodos Cuasi-Newton

Características

- Al igual que el método del paso más descendente sólo requiere el gradiente.
- Por el cambio del gradiente puede construir un modelo de la función objetivo los suficientemente buena para producir una convergencia superlineal.
- Dado que la segunda derivada no es requerida, estos métodos son generalmente más eficientes que el de Newton.

Existen dos métodos muy populares Davidon-Fletcher-Powell (DFP) y Broyden-Fletcher-Goldfarb-Shanno (BFGS).
La expresión basada en la serie de Taylor para modelar cuadraticamente la función objetivo en un punto actual x_i esta dada por:

$$f(x_i) \approx f(x_{i+1}) + \nabla f(x_{i+1})^T (x_i - x_{i+1}) + \frac{1}{2}(x_i - x_{i+1})^T B_{i+1}(x_i - x_{i+1})$$

donde (∇f) es el gradiente, and B una aproximación a la matriz Hessian. El gradiente de la aproximación respecto a x_i esta dado por

$$\nabla f(x_i) \approx \nabla f(x_{i+1}) + B_{i+1}(x_i - x_{i+1})$$

Reordenando la expresión

$$B_{i+1}(x_i - x_{i+1}) \approx \nabla f(x_i) - \nabla f(x_{i+1})$$

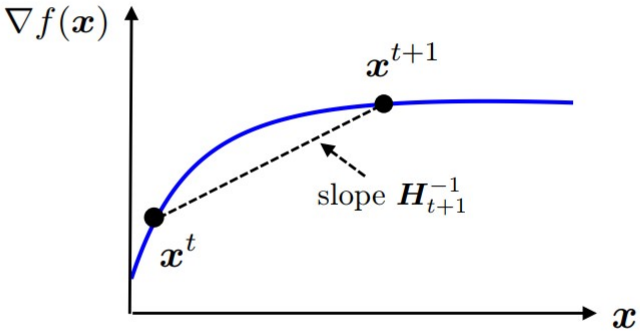
Se puede observar que hay un diferencia en el gradiente y en el punto x, si nombramos $s_i = x_{i+1} - x_i$ a la diferencia de x, y a la diferencia del gradiente $y_i = \nabla f(x_i) - \nabla f(x_{i+1})$, así la expresión anterior se convierte en

$$B_{i+1} s_i = yi$$

Esta ultima expresión es conocida como la ecuación de la secante

$$B_{i+1}(x_i - x_{i+1}) \approx \nabla f(x_i) - \nabla f(x_{i+1})$$

$$B_{i+1}s_i = yi$$



Métodos Cuasi-Newton

Si hay un desplazamiento en s_i, entonces a un cambio de gradiente y_i, la ecuación de la secante requiere que la matriz B_{i+1} sea simétrica, definida y positiva para poder mapear s_i a y_i. Esto es

$$s_k^T y_k > 0$$

Dado que el sistema tiene n soluciones los métodos basado en cuasi-newton se plantena una matriz B_{i+1} cercana B_i, entonces resuelven el siguiente problemas

$$min||Bi - B_{i+1}||$$

$$\text{Sujeto a } B = B^T \quad Bs_k = y_k$$

Método Davidon-Fletcher-Powell (DFP)

En el método DFP, la inversa de la Hessiana se aproxima mediante una matriz [A] y la dirección de búsqueda viene dada por

$$S_i = -[A]\nabla f(x_i)$$

La inversa de la matriz Hessiana se aproxima por

$$A_{i+1} = -A_i + \frac{\Delta x \Delta x^T}{\Delta x^T \Delta g} - \frac{A_i \Delta g \Delta g^T A_i}{\Delta g^T A_i \Delta g}$$

donde

$$\Delta g = \nabla g_i - \nabla g_{i-1}$$

$$\Delta x = \Delta x_i - \Delta x_{i-1}$$

TABLE 3.7

Algorithm for the DFP Method

Step 1: Given x_i (starting value of design variable)
ε_1 (tolerance of function value from previous iteration)
ε_2 (tolerance on gradient value)
Δx (required for gradient computation)
$[A]$ (initialize to identity matrix)

Step 2: Compute $f(x_i)$ and $\nabla f(x_i)$ (function and gradient vector)
$S_i = -\nabla f(x_i)$ (search direction)
$x_{i+1} = x_i + \alpha S_i$ (update the design vector)
Minimize $f(x_{i+1})$ and determine α (use the golden section method)

Step 3: Compute Δx and ∇g

$$[A]_{i+1} = [A]_i + \frac{\Delta x \Delta x^T}{\Delta x^T \nabla g} - \frac{[A]_i \nabla g \nabla g^T [A]_i}{\nabla g^T [A]_i \nabla g}$$

$S_{i+1} = -[A]_{i+1} \nabla f(x_{i+1})$
$x_{i+2} = x_{i+1} + \alpha S_{i+1}$
Minimize $f(x_{i+2})$ and determine α (use the golden section method)
If $|f(x_{i+2}) - f(x_{i+1})| > \varepsilon_1$ or $\|\nabla f(x_{i+1})\| > \varepsilon_2$
then goto Step 3
else goto Step 4

Step 4: Converged. Print $x^* = x_{i+2}$, $f(x^*) = f(x_{i+2})$

Initial function value = 1452.2619

No.	x-vector		f(x)	Deriv.
1	0.095	0.023	-2.704	1006.074
2	0.179	0.145	-5.418	37.036
3	0.508	0.145	-9.576	23.983
4	0.501	0.122	-9.656	7.004
5	0.504	0.122	-9.656	0.396
6	0.504	0.122	-9.656	0.053
7	0.504	0.122	-9.656	0.038
8	0.504	0.122	-9.656	0.028
9	0.504	0.122	-9.656	0.005

```
>> A
A =
    0.0091    0.0005
    0.0005    0.0033
>> inv(hessian(x,delx,n_of_var))
ans =
    0.0091    0.0005
    0.0005    0.0033
```

Método Broyden-Fletcher-Goldfarb-Shanno (BFGS)

En el método BFGS, la Hessiana se aproxima usando la matriz métrica variable [A] dada por la ecuación

$$A_{i+1} = A_i + \frac{g\Delta g^T}{\Delta g^T \Delta x} + \frac{\nabla f(x_i)\nabla f(x_i)^T}{\nabla f(x_i)^T S_i}$$

Método Broyden-Fletcher-Goldfarb-Shanno (BFGS)

Algorithm for the BFGS Method

Step 1: Given x_i (starting value of design variable)
ε_1 (tolerance of function value from previous iteration)
ε_2 (tolerance on gradient value)
Δx (required for gradient computation)
$[A]$ (initialize to identity matrix)

Step 2: Compute $f(x_i)$ and $\nabla f(x_i)$ (function and gradient vector)
$S_i = -\nabla f(x_i)$ (search direction)
$x_{i+1} = x_i + \alpha S_i$ (update the design vector)
Minimize $f(x_{i+1})$ and determine α (use golden section method)

Step 3: Compute Δx and ∇g

$$[A]_{i+1} = [A]_i + \frac{g \nabla g^T}{\nabla g^T \Delta x} + \frac{\nabla f(x_i) \nabla f(x_i)^T}{\nabla f(x_i)^T S_i}$$

$S_{i+1} = -[[A]_{i+1}]^{-1} \nabla f(x_{i+1})$
$x_{i+2} = x_{i+1} + \alpha S_{i+1}$
Minimize $f(x_{i+2})$ and determine α (use the golden section method)
If $\left| f(x_{i+2}) - f(x_{i+1}) \right| > \varepsilon_1$ or $\|\nabla f(x_{i+1})\| > \varepsilon_2$
then goto Step 3
else goto Step 4

Step 4: Converged. Print $x^* = x_{i+2}$, $f(x^*) = f(x_{i+2})$

BFGS Algorithm

$H_0 \in \mathbb{R}^{n\times n}$ initial symmetric, pos. def. matrix.
$x_0 \in \mathbb{R}^n$, $k = 0$ $\quad g_k = \nabla f(x_k)$

Step 1. $d_k = -H_k g_k$ [Search direction]

Step 2. $\alpha_k = \arg\min_{\alpha>0} f(x_k + \alpha d_k)$ [Line search]

$x_{k+1} = x_k + \alpha_k d_k \qquad p_k = x_{k+1} - x_k = \alpha_k d_k$

$g_{k+1} = \nabla f(x_{k+1})$

Step 3. $q_k = g_{k+1} - g_k$

$$H_{k+1}^{BFGS} = H_k^{BFGS} + \left(1 + \frac{q_k^T H_k^{BFGS} q_k}{p_k^T q_k}\right)\frac{p_k p_k^T}{p_k^T q_k} - \frac{p_k q_k^T H_k^{BFGS} + H_k^{BFGS} q_k p_k^T}{q_k^T p_k}$$

$k = k + 1$ and return to Step 1.

BFGS is almost the same as DFP, only the H update is different.

In practice BFGS seems to work better than DFP.

Método Broyden-Fletcher-Goldfarb-Shanno (BFGS)

```
        Initial function value = 1452.2619
No.         x-vector              f(x)           Deriv.
-------------------------------------------------------
1      0.095       0.023       -2.704        1006.074
2      0.179       0.145       -5.418          37.036
3      0.508       0.145       -9.578          24.017
4      0.501       0.122       -9.655           6.900
5      0.504       0.122       -9.656           0.471
6      0.504       0.122       -9.656           0.077
7      0.504       0.122       -9.656           0.056
8      0.504       0.122       -9.656           0.040
9      0.504       0.122       -9.656           0.007
-------------------------------------------------------

>> A
A =
    110.5001  -16.9997
    -16.9997  306.7238
>> hessian(x,delx,n_of_var)
ans =
    111.0981  -15.9640
    -15.9640  308.5603
```