

# NUMERIKUS OPTIMALIZÁLÁS

HAJDER LEVENTE 2007.10.17.

## 1. BEVEZETÉS

A számítógépes rekonstrukció során elfordulnak olyan problémák, amelyekre nem tudunk zárt alakú optimális megoldás adni. Ebben az esetben segítenek a numerikus módszerek, amelyek a megadott hibafüggvényt próbálják meg egy meghatározott helyről csökkenteni.

Sok esetben az is segít az eredetileg meghatározott hibafüggvényünkön, ha azt módosítjuk, és az eredetitől eltérő hibafüggvényt optimalizálunk, amely közel ugyanazt a megoldást szolgáltatja, mint az eredeti. Ebben az esetben is pontosíthatunk a kapott megoldáson numerikus optimalizálás segítségével

## 2. TAYLOR-SOROS KÖZELÍTÉS

A numerikus megoldásainkhoz a legtöbb esetben szükség lesz egy megadott helyen a költségfüggvény alakjára. Jelöljük  $J(x)$ -szel a költségfüggvényt, ahol  $x = (x_1, x_2, \dots, x_n)^T$  egy  $n$ -dimenziós változókat tartalmazó vektor. Az  $x_0$  pontban a költségfüggvényünket így tudjuk meghatározni:

$$J(x_0 + \Delta x) = J(x) + \nabla J(x)_0^T \Delta x + \frac{1}{2!} \Delta x^T \nabla^2 J(x_0) \Delta x + \dots$$

ahol egyrészt a keletkező harmadfokú és ennél is magasabb fokú tagokat elhanyagoljuk, másrészt  $\nabla J(x)$  jelöli a gradiensét a  $J(x)$  függvénynek:

$$\nabla J(x) = \begin{bmatrix} \frac{\partial J(x)}{\partial x_1} \\ \frac{\partial J(x)}{\partial x_2} \\ \vdots \\ \frac{\partial J(x)}{\partial x_n} \end{bmatrix}$$

$\nabla^2 J(x)$  pedig a második parciális deriváltból képzett mátrix:

$$H_{J(x)} = \nabla^2 J(x) = \begin{bmatrix} \frac{\partial^2 J(x)}{\partial x_1^2} & \frac{\partial^2 J(x)}{\partial x_1 \partial x_2} & \dots & \frac{\partial^2 J(x)}{\partial x_1 \partial x_n} \\ \frac{\partial^2 J(x)}{\partial x_2 \partial x_1} & \frac{\partial^2 J(x)}{\partial x_2^2} & \dots & \frac{\partial^2 J(x)}{\partial x_n \partial x_2} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 J(x)}{\partial x_n \partial x_1} & \frac{\partial^2 J(x)}{\partial x_n \partial x_2} & \dots & \frac{\partial^2 J(x)}{\partial x_n^2} \end{bmatrix}$$

A  $H_{J(x)}$  mátrixot Hesse-féle mátrixnak hívják.

Megjegyzés: egyváltozós esetben a Taylor sor az analízisből jól ismert alakban írható fel:

$$J(x + \Delta x) = f(x_0) + \frac{f'(x_0)}{1!} \Delta x + \frac{f''(x_0)}{2!} \Delta x^2 + \frac{f'''(x_0)}{3!} \Delta x^3 + \dots$$

### 3. GRADIENS MÓDSZER

A gradiens módszer a numerikus eljárások közül a legegyszerűbb. Egy megadott  $x_i$  pontban úgy lehet csökkenteni a hibafüggvény értékét, ha a gradienssel ellentétes irányba lépünk:

$$x_{i+1} = x_i - \alpha_i \nabla J(x_i)$$

Ezt a módszert nevezzük gradiens módszernek. Az eljárás addig tart, amíg a gradiens értéke elég kicsi nem lesz (ha a gradiens nulla, akkor érkeztünk szélsőértékhez).

A gradiens módszer kiegészítése a konjugált gradiens módszer. Ebben az esetben a gradiens irányát az ún. konjugált iránnyal módosítjuk, és úgy lépünk tovább. Nagy előnye, hogy  $n$ -dimenziós kvadratikus hibafelületnél  $n$  lépésben a minimumba konvertál. (Sajnos hibafelületeink nem kvadratikusak, de ha azok lennének, zárt alakú megoldással egy lépésben is ki lehetne számítani.) Használatát itt most idő hiányában nem közöljük.

Fontos kérdés  $\alpha_i$  valós skalár meghatározása. A sima gradiens módszernél konstansnak szokás venni, de léteznek szofisztikáltabb számítások is, ahogyan azt a Newton-módszenél is látni fogjuk.

### 4. NEWTON-MÓDSZER

Egydimenziós esetben írjuk (második deriváltig megtartva), hogy

$$J(x + \Delta x) = f(x_0) + \frac{f'(x_0)}{1!} \Delta x + \frac{f''(x_0)}{2!} \Delta x^2$$

Azt szeretnénk, hogy a  $\Delta x$  szerinti elmozdulás a minimumot adná. Ezért deriválni kell a  $J(x + \Delta x)$  függvényt  $\Delta x$  szerint, és zérussal kell egyenlővé tenni:

$$\frac{\partial J(x + \Delta x)}{\partial \Delta x} = f'(x_0) + f''(x_0) \Delta x = 0$$

Azaz  $\Delta x = -\frac{f'(x_0)}{f''(x_0)}$  adódik.

Többdimenziós esetben a parciális derivált így néz ki:

$$\frac{\partial J(x_0 + \Delta x)}{\partial \Delta x} = \nabla J(x_0) + H_{J(x_0)} \Delta x = 0$$

Ebből pedig adódik, hogy

$$\Delta x = -H_{J(x_0)}^{-1} \nabla J(x_0)$$

### 5. GAUSS-NEWTON-MÓDSZER

A Gauss-Newton módszer szintén numerikus optimalizálásra való, és a "sima" Newton-módszerből származtatható. Az eredeti Newton-féle numerikus eljárásnál az alábbi összefüggés segítségével végzi az iterációt:

$$x_{i+1} = x_i - H_{J(x_0)}^{-1} \nabla J(x_i)$$

Tegyük fel, hogy  $J(x)$ -et az alábbi alakra lehet hozni:

$$J(x) = \sum_{i=1}^m (f_i(x))^2 = f(x)^T f(x)$$

ha  $f(x)^T = [f_1(x), f_2(x), \dots, f_m(x)]$ . Úgy kell elképzelni, hogy a hibafüggvényünk legkisebb négyzetes alakban adott, és összesen  $m$  darab - magában skalár - elemre lehet bontani, melyek négyzetét adjuk össze. Ebben az esetben  $J(x)$  gradiensét az alábbi alakban írhatjuk fel:

$$\nabla J(x) = 2\nabla f(x)^T f(x)$$

ahol  $\nabla f(x)$  most egy  $m \times n$ -szeres mátrixot jelöl (amit Jacobi-mátrixnak szokás nevezni):

$$\nabla f(x) = \begin{bmatrix} \frac{\partial f_1(x)}{\partial x^1} & \frac{\partial f_1(x)}{\partial x^2} & \dots & \frac{\partial f_1(x)}{\partial x^n} \\ \frac{\partial f_2(x)}{\partial x^1} & \frac{\partial f_2(x)}{\partial x^2} & \vdots & \frac{\partial f_2(x)}{\partial x^n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_m(x)}{\partial x^1} & \frac{\partial f_m(x)}{\partial x^2} & \dots & \frac{\partial f_m(x)}{\partial x^n} \end{bmatrix}$$

Ezt úgy tudjuk belátni, hogy felírjuk a tetszőleges  $x^k$  szerinti parciális deriváltat: (A változók megkülönböztetésére most kivételesen felső indexet használunk, mert az alsó indexet már az iteráció számára felhasználtuk. Ezért  $x = [x^1, x^2, \dots, x^n]^T$  jelöli most a változóvektorokat.)

$$\frac{\partial J(x)}{\partial x^k} = \frac{\sum \partial(f_i(x))^2}{\partial x^k} = \sum 2f_i(x) \frac{\partial f_i(x)}{\partial x^k}$$

Ezt pedig az  $f(x)$  gradiens vektorának a  $k$ -dik eleme megszorozva  $f_i(x)$ -ek összegével. A Hesse mátrix összeállításához szükség van a második parciális deriváltakra:

$$\frac{\partial J(x)}{\partial x^k \partial x^l} = 2 \sum \left( \frac{\partial f_i(x)}{\partial x^k} \frac{\partial f_i(x)}{\partial x^l} + f_i(x) \frac{\partial^2 f_i(x)}{\partial x^k \partial x^l} \right)$$

Ha a második tagot elhagyjuk, átírhatjuk az összefüggésünket:

$$\frac{\partial J(x)}{\partial x^k \partial x^l} \approx 2 \sum \left( \frac{\partial f_i(x)}{\partial x^k} \frac{\partial f_i(x)}{\partial x^l} \right)$$

Ennek az elhanyagolásnak az az előnye, hogy nem kell a második parciális deriváltakat kiszámítani, hanem elég az első deriváltak szorzatával dolgozni. Mindez akkor igaz, ha az egyes  $f_i(x)$ -ek értéke kicsi, azaz a hibafüggvényünk viszonylag kicsi értékeket tartalmaz.

A Hesse mátrix ebben az esetben az alábbi alakra egyszerűsödik le:

$$H_{J(x_0)} \approx 2\nabla f(x)^T \nabla f(x)$$

Maga az optimalizálás pedig továbbra is egy iteráció, amelyet a visszahelyettesítések elvégzésével az alábbi alakra hozhatunk (a 2-és konstansokat elhanyagolhatjuk):

$$x_{i+1} = x_i - (\nabla f(x_i)^T \nabla f(x_i))^{-1} \nabla f(x_i)^T f(x_i)$$

## 6. LEVENBERG-MARQUARDT ALGORITMUS

A Levenberg-Marquardt algoritmus az egyik leggyakrabban alkalmazott numerikus optimalizálási módszer. Működése nagyon egyszerű: ugyanúgy iteratív eljárás, mint

a korábbiak; és a paraméter megváltoztatását a Gauss-Newton és a gradiens módszer keverésével végzi. Keverés alatt súlyozást értünk. Egy iteráció az alábbi összefüggés segítségével módosítja az optimalizálandó paramétert:

$$x_{i+1} = x_i - \left[ (\nabla f(x_i)^T \nabla f(x_i))^{-1} + \lambda E \right] \nabla f(x_i)^T f(x_i)$$

ahol  $\lambda$  egy paraméter, melyet növelünk/illetve csökkentünk az iteráció során, annak függvényében, hogy sikerült-e a hibát, azaz a  $J(x)$  függvényt csökkenteni. Ha csökken a hiba,  $\lambda$ -t növeljük, és gradiens módszerré alakul az eljárás, ha növekszik a hiba, visszelépünk, és  $\lambda$ -t csökkentjük (azaz a szofisztikáltabb Gauss-Newton módszer súlyát növeljük).