Termineringsvillkor

I en ideal värld skulle en minimeringsalgoritm terminera med lösningsförslag x_k som uppfyllde $\nabla f(x_k) = 0$ och $\nabla^2 f(x_k)$ positivt semidefinit.

Detta är inte speciellt realistiskt då den ändliga aritmetiken introducerar fel av olika slag, och dessutom är ingen algoritm garanterad att uppnå detta i finit tid.

För att hantera detta skulle vi kunna använda

$$\|\nabla f(x_k)\| \leq \epsilon,$$

för något ϵ, som termineringsvillkor. Detta är dock skalningsberoende, dvs ett byte av enhet i f leder till motsvarande skalning av ∇f och påverkar därmed termineringsvillkorets styrka.

T ex skulle ett byte av enhet från t ex kilometer till millimeter motsvara en skalning med 10^6.

Termineringsvillkor, forts.

En modifiering av termineringsvillkoret till att bli *relativt*

$$\|\nabla f(x_k)\| \leq \epsilon |f(x_k)|$$

löser det problemet, men leder i stället till problem om $f(x_*) \approx 0$, och skulle t o m kunna leda till en oändlig loop om $f(x_k)$ beräknas $= 0$ men $\nabla f(x_k)$ beräknas $\neq 0$ p g a avrundningsfel.

En kombination av dessa båda ger dock ett användbart konvergenskrav:

$$\|\nabla f(x_k)\| \leq \epsilon \left(1 + |f(x_k)|\right),$$

som uppför sig som ett absolut test då $f(x_k) \approx 0$, och som ett relativt test i övrigt.

Termineringsvillkor, forts.

Då inget perfekt konvergensvillkor existerar är det inte ovanligt att man kombinerar flera olika test. Förutom testet på gradienten och hessianen ovan vill man att funktionsvärdena $\{f(x_k)\}$ och iterationssekvensen $\{x_k\}$ ska konvergera. Det kombinerade testet skulle då se ut som följer:

$$\begin{aligned}
\|\nabla f(x_k)\| &\leq \epsilon_1 \left(1 + |f(x_k)|\right) \\
f(x_{k-1}) - f(x_k) &\leq \epsilon_2 \left(1 + |f(x_k)|\right) \\
\|x_{k-1} - x_k\| &\leq \epsilon_3 \left(1 + \|x_k\|\right) \\
\nabla^2 f(x_k) + \epsilon_4 I & \quad \text{positivt semi-definit}
\end{aligned}$$

Om alla värden beräknas med samma precision ϵ_{mach} är värdena $\epsilon_2 = \epsilon_{mach}, \epsilon_1 = \epsilon_3 = \sqrt{\epsilon_2}, \epsilon_4 = \epsilon_2 \|\nabla^2 f(x_k)\|$ vanliga.

Minsta-kvadratproblem

För minstakvadratproblem

$$\min_x f(x) = \frac{1}{2} F(x)^T F(x)$$

jämför man i stället

$$\|Jp\| = \|J(J^T J)^{-1} J^T F\|$$

med $\|F\|$.

Matrisen $J(J^T J)^{-1} J^T$ kallas $P_{\mathcal{R}(J)}$, projektionsmatrisen på värderummet till J.

En fördel med att jämföra $\|Jp\|$ med $\|F\|$ är att de befinner sig i samma rum. Dessutom gäller att

$$\frac{\|Jp\|}{\|F\|} = \cos\alpha,$$

där α är vinkeln mellan residualen F och tangentplanet till F. Nära lösningen går $\|Jp\|/\|F\|$ mot 0 och α mot $\pi/2$, d.v.s. residualen blir ortogonal mot tangentplanet i lösningen.

Val av normer

Testen på $\nabla f(x_k)$ och x_k är baserade på normer. Teoretiskt duger vilken norm som helst, men om $\nabla f(x_k) = (\gamma, \ldots, \gamma)^T$, så är

$$\|\nabla f(x_k)\|_2 = \sqrt{n}|\gamma|$$

och

$$\|\nabla f(x_k)\|_\infty = |\gamma|,$$

så för stora problem är oändlighetsnormen lämpligast.

Skalning, viktad norm

En annan effekt av att använda normer är att olika element i x_k bestäms med olika *relativ* noggrannhet om storleken på elementen varierar kraftigt.

Om $x_* \approx [1 \; 10^{-4} \; 10^{-6}]^T$, så kommer $|x_1|$ att dominera $\|x_*\|$, vilket kan få metoden att terminera innan de övriga elementen bestämts med någon större noggrannhet. Om det i förväg är känt att magnituden på elementen varierar kraftigt, kan en skalning av variablerna vara lämplig. En annan lösning är att använda en *viktad* norm för konvergenstestet:

$$\|x\|_W = \|W^{1/2}x\|_p,$$

där $W^{1/2}$ är en diagonalmatris och p är någon av de vanliga normerna.

Metoder som undviker att beräkna hessianen

En nackdel hos Newtons metod är ett hessianen måste

- härledas, vilket kan ge upphov till fel,
- beräknas, vilket typiskt kräver $\mathcal{O}(n^2)$ operationer,
- lagras, vilket kräver plats för n^2 element,
- "inverteras", vilket kräver $\mathcal{O}(n^3)$ operationer.

Metoder som undviker att beräkna hessianen, forts.

Då vilken positivt definit matris som helst duger för att garantera global konvergens, finns det många metoder som ersätter hessianen i Newtons ekvation

$$\nabla^2 f(x_k)p = -\nabla f(x_k)$$

med någon annan positivt definit matris B_k;

$$B_k p = -\nabla f(x_k). \tag{1}$$

Vi kommer att studera 2 sådana metoder, *steepest descent* och *kvasi-newton*.

Notera att våra "gamla" algoritmer går att se som specialfall av ekvation (1) ovan: $B_k = \nabla^2 f(x_k) + \lambda I$ ger trust-region-algoritmen, $B_k = J(x_k)^T J(x_k)$ ger Gauss-Newtons algoritm och $B_k = J(x_k)^T J(x_k) + \lambda I$ ger Levenberg-Marquardts algoritm.

Steepest descent

Om vi använder den trivialaste "approximationen" av hessianen $B_k = I$ får vi *Steepest descent*-metoden:

$$\begin{aligned} B_k &= I \\ &\Downarrow \\ p_k &= -\nabla f(x_k) \end{aligned}$$

Då samtliga $\lambda_i(B_k) = 1$, uppfyller p_k *sufficient descent*-villkoret med

$$\epsilon = \frac{\lambda_{min}}{\lambda_{max}} = \frac{1}{1} = 1$$

och gradientrelativ-villkoret med

$$m = \frac{1}{\lambda_{min}} = \frac{1}{1} = 1.$$

Steepest descent, konvergens

Steepest descent använder den billigast tänkbara approximationen av hessianen och sökriktningen fås utan några extra beräkningar, vilket gör att kostnaden per iteration blir minimal.

Detta har dock ett pris; dålig konvergenshastighet. Det går att visa att metoden konvergerar linjärt med en konvergenskonstant C begränsad uppåt av

$$C_{sup} = \left(\frac{\kappa(Q) - 1}{\kappa(Q) + 1}\right)^2,$$

där $Q = \nabla^2 f(x_*)$. I praktiken brukar konvergenskonstanterna hamna nära detta värde.

Steepest descent, konvergens

En tabell över konvergenskonstanter visar att konvergenshastigheten lätt kan bli dålig:

$\kappa(Q)$	1	10	100	1000	10000
C	0	0.6694	0.9608	0.9960	0.9996

I praktiken (bör) används (användas) *steepest descent* endast för problem som är kända för att vara välkonditionerade.

Kvasi-newton-metoder

Kvasi-newton kan ses som en generalisering av sekant-metoden i en dimension:

I sekant-metoden används approximationen

$$\begin{aligned} f''(x_k) &\approx \frac{f'(x_k) - f'(x_{k-1})}{x_k - x_{k-1}} \\ f''(x_k)(x_k - x_{k-1}) &\approx f'(x_k) - f'(x_{k-1}) \end{aligned}$$

vilket i flera dimensioner motsvaras av

$$\nabla^2 f(x_k)(x_k - x_{k-1}) \approx \nabla f(x_k) - \nabla f(x_{k-1}).$$

Från detta för vi villkoret vi kommer att använda för att definiera kvasi-newton-approximationerna B_k:

$$B_k(x_k - x_{k-1}) = \nabla f(x_k) - \nabla f(x_{k-1}).$$

Vi kallar detta för *sekantvillkoret*.

Då B_k har n^2 element och sekantvillkoret har n ekvationer finns åtskilliga frihetsgrader kvar för att definiera B_k.

Kvasi-newton-metoder, forts.

För den kvadratiska funktionen $f(x) = \frac{1}{2}x^T Qx - c^T x$ gäller att

$$\begin{aligned} Q(x_k - x_{k-1}) &= (Qx_k - c) - (Qx_{k-1} - c) \\ &= \nabla f(x_k) - \nabla f(x_{k-1}), \end{aligned}$$

dvs hessianen Q uppfyller sekantvillkoret. I princip kommer vi att kräva att approximationen B_k uppför sig som hessianen i en speciell riktning.

Vi inför definitionerna $s_k = x_{k+1} - x_k$ och $y_k = \nabla f(x_{k+1}) - \nabla f(x_k)$. Dessa kommer att användas i definitionerna av olika kvasi-newton-metoder. Med dessa blir sekantvillkoret

$$B_k s_{k-1} = y_{k-1} \text{ eller } B_{k+1} s_k = y_k.$$

Kvasi-newton-metoder, forts.

En av de enklaste kvasi-newton-approximationerna ges av

$$B_{k+1} = B_k + \frac{(y_k - B_k s_k)(y_k - B_k s_k)^T}{(y_k - B_k s_k)^T s_k} \quad (2)$$

Denna illustrerar flera vanliga egenskaper hos kvasi-newton-approximationer:

- Sekantvillkoret kommer att uppfyllas oberoende av hur B_k väljs:

$$\begin{aligned} B_{k+1} s_k &= B_k s_k + \frac{(y_k - B_k s_k)(y_k - B_k s_k)^T}{(y_k - B_k s_k)^T s_k} s_k \\ &= B_k s_k + \frac{(y_k - B_k s_k)((y_k - B_k s_k)^T s_k)}{(y_k - B_k s_k)^T s_k} \\ &= B_k s_k + (y_k - B_k s_k) = y_k. \end{aligned}$$

Kvasi-newton-metoder, forts.

- Den nya approximationen B_{k+1} fås genom att *uppdatera* den gamla approximationen B_k. Därför kallas kvasi-newton-approximationer av typen (2) ovan för *uppdateringsformler*. Som startapproximation B_0 kan I användas, men om en bättre approximation kan fås billigt bör den användas.

- Den nya approximationen B_{k+1} kan beräknas med $\mathcal{O}(n^2)$ operationer, då uppdateringen är en produkt av vektorer. Mer förvånande är att lösningen till ekvationssystemet $B_{k+1}p = -\nabla f(x_{k+1})$ kan beräknas med $\mathcal{O}(n^2)$ operationer (i stället för normala $\mathcal{O}(n^3)$), då det är möjligt att härleda formler för att uppdatera cholesky-faktoriseringen av B_k.

Kvasi-newton-metoder, forts.

Den mest använda kvasi-newton-approximationen är Broyden-Fletcher-Goldfarb-Shanno (BFGS):

$$B_{k+1} = B_k - \frac{(B_k s_k)(B_k s_k)^T}{s_k^T B_k s_k} + \frac{y_k y_k^T}{y_k^T s_k},$$

som bevarar både symmetri och positiv definithet om $y_k^T s_k > 0$, vilket kan lösas om en linjesökningsalgoritm som inkluderar Wolfes villkor används.

Kvasi-newton-metoder, forts.

BFGS tillhör en klass som ibland kallas *Broyden-klassen* av uppdateringar:

$$B_{k+1} = B_k - \frac{(B_k s_k)(B_k s_k)^T}{s_k^T B_k s_k} + \frac{y_k y_k^T}{y_k^T s_k} + \phi(s_k^T B_k s_k) u_k u_k^T,$$

där ϕ är en skalär och

$$u_k = \frac{y_k}{y_k^T s_k} - \frac{B_k s_k}{s_k^T B_k s_k}.$$

$\phi = 0$ ger BFGS. $\phi = 1$ ger en uppdatering kallad Davidon-Fletcher-Powell (DFP).

Hela Broyden-klassen behåller positiv definithet om $y_k^T s_k > 0$.

Kvasi-newton-metoder konvergerar typiskt super-linjärt, vilket kan förklaras med att de går från *steepest descent* mot Newton.