Komputerowa analiza zagadnień różniczkowych

2. Metoda gradientów sprzężonych

Minimalizacja i układy równań algebraicznych

P. F. Góra

http://th-www.if.uj.edu.pl/zfs/gora/

semestr letni 2007/08

Metoda gradientów sprzężonych — motywacja

Rozważmy funcję $f : \mathbb{R}^N \to \mathbb{R}$

$$f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T\mathbf{A}\mathbf{x} - \mathbf{b}^T\mathbf{x} + c\,, \tag{1}$$

gdzie $\mathbf{x}, \mathbf{b} \in \mathbb{R}^N$, $c \in \mathbb{R}$, $\mathbf{A} = \mathbf{A}^T \in \mathbb{R}^{N\times N}$ jest *symetryczna i dodatnio określona*. Przy tych założeniach, funkcja (1) ma dokładnie jedno minimum, będące zarazem minimum globalnym. Szukanie minimów dodatnio określonych form kwadratowych jest (względnie) łatwe i z praktycznego punktu widzenia ważne. Minimum to leży w punkcie spełniającym

$$\nabla f = 0\,. \tag{2}$$

Obliczmy

$$\frac{\partial f}{\partial x_i} = \frac{1}{2}\frac{\partial}{\partial x_i}\sum_{j,k} A_{jk}x_j x_k - \frac{\partial}{\partial x_i}\sum_j b_j x_j + \underbrace{\frac{\partial c}{\partial x_i}}_{0}$$

$$= \frac{1}{2}\sum_{j,k} A_{jk}\left(\underbrace{\frac{\partial x_j}{\partial x_i}}_{\delta_{ij}} x_k + x_j \underbrace{\frac{\partial x_k}{\partial x_i}}_{\delta_{ik}}\right) - \sum_j b_j \underbrace{\frac{\partial x_j}{\partial x_i}}_{\delta_{ij}}$$

$$= \frac{1}{2}\sum_k A_{ik}x_k + \frac{1}{2}\sum_j A_{ji}x_j - b_i = \frac{1}{2}\sum_k A_{ik}x_k + \frac{1}{2}\sum_j A_{ij}x_j - b_i$$

$$= (\mathbf{Ax} - \mathbf{b})_i \ . \tag{3}$$

Widzimy zatem, że funkcja (1) osiąga minimum w punkcie, w którym zachodzi

$$\mathbf{Ax} - \mathbf{b} = 0 \Leftrightarrow \mathbf{Ax} = \mathbf{b}. \tag{4}$$

Rozwiązywanie układu równań liniowych (4) z macierzą symetryczną, dodatnio określoną jest równoważne poszukiwaniu minimum dodatnio określonej formy kwadratowej.

Przypuśćmy, że macierz $\mathbf{A}$ jest przy tym *rzadka* i duża (lub co najmniej średnio-duża). Wówczas metoda gradientów sprzężonych jest godną uwagi metodą rozwiązywania (4)

Metoda gradientów sprzężonych, *Conjugate Gradients*, CG

$\mathbf{A} \in \mathbb{R}^{N\times N}$ symetryczna, dodatnio określona, x_1 — początkowe przybliżenie rozwiązania równania (4), $0 < \varepsilon \ll 1$.

$$
\begin{aligned}
&\mathbf{r}_1 = \mathbf{b} - \mathbf{A}\mathbf{x}_1,\ \mathbf{p}_1 = \mathbf{r}_1 \\
&\textbf{while } \|\mathbf{r}_k\| > \varepsilon \\
&\qquad \alpha_k = \frac{\mathbf{r}_k^T \mathbf{r}_k}{\mathbf{p}_k^T \mathbf{A}\mathbf{p}_k} \\
&\qquad \mathbf{r}_{k+1} = \mathbf{r}_k - \alpha_k \mathbf{A}\mathbf{p}_k \\
&\qquad \beta_k = \frac{\mathbf{r}_{k+1}^T \mathbf{r}_{k+1}}{\mathbf{r}_k^T \mathbf{r}_k} \\
&\qquad \mathbf{p}_{k+1} = \mathbf{r}_{k+1} + \beta_k \mathbf{p}_k \\
&\qquad \mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{p}_k \\
&\textbf{end}
\end{aligned}
\tag{5}
$$

Wówczas zachodzą twierdzenia:

Twierdzenie 1. *Ciągi wektorów* $\{\mathbf{r}_k\}$, $\{\mathbf{p}_k\}$ *spełniają następujące zależności:*

$$\mathbf{r}_i^T \mathbf{r}_j = 0\,, \quad i > j\,, \tag{6a}$$

$$\mathbf{r}_i^T \mathbf{p}_j = 0\,, \quad i > j\,, \tag{6b}$$

$$\mathbf{p}_i^T \mathbf{A} \mathbf{p}_j = 0\,, \quad i > j\,. \tag{6c}$$

Twierdzenie 2. *Jeżeli* $\mathbf{r}_M = 0$*, to* $\mathbf{x}_M$ *jest ścisłym rozwiązaniem równania* (4)*.*

Dowód. Oba (sic!) dowody przebiegają indukcyjnie. □

Ciąg $\{\mathbf{x}_k\}$ jest w gruncie rzeczy “pomocniczy”, nie bierze udziału w iteracjach, służy tylko do konstruowania kolejnych przybliżeń rozwiązania.

Istotą algorytmu jest konstruowanie dwu ciągów wektorów spełniających zależności (6). Wektory $\{\mathbf{r}_k\}$ są wzajemnie prostopadłe, a zatem *w arytmetyce dokładnej* $\mathbf{r}_{N+1} = 0$, wobec czego $\mathbf{x}_{N+1}$ jest poszukiwanym ścisłym rozwiązaniem.

Zauważmy, że ponieważ $\mathbf{A}$ jest symetryczna, dodatnio określona, warunek (6c) oznacza, że wektory $\{\mathbf{p}_k\}$ są wzajemnie prostopadłe w metryce zadanej przez $\mathbf{A}$. Ten właśnie warunek nazywa się warunkiem *sprzężenia względem* $\mathbf{A}$, co daje nazwę całej metodzie.

Koszt metody

W arytmetyce dokładnej metoda zbiega się po N krokach, zatem jej koszt wynosi $O(N \cdot \text{koszt\_jednego\_kroku})$. Koszt jednego kroku zdominowany jest przez obliczanie iloczynu $\mathbf{A}\mathbf{p}_k$. Jeśli macierz $\mathbf{A}$ jest pełna, jest to $O(N^2)$, a zatem całkowity koszt wynosi $O(N^3)$, czyli tyle, ile dla metod dokładnych. Jeżeli jednak $\mathbf{A}$ jest rzadka, koszt obliczania iloczynu jest mniejszy (o ile obliczenie to jest odpowiednio zaprogramowane). Jeśli $\mathbf{A}$ jest pasmowa o szerokości pasma $M \ll N$, całkowity koszt wynosi $O(M \cdot N^2)$.

Problem!

W arytmetyce o skończonej dokładności kolejne generowane wektory nie są *ściśle* ortogonalne do swoich poprzedników — na skutek akumulującego się błędu zaokrąglenia rzut na poprzednie wektory może stać się z czasem znaczny. Powoduje to istotne spowolnienie metody.

Twierdzenie 3. *Jeżeli* $\mathbf{x}$ *jest ścisłym rozwiązaniem równania* (4)*,* $\mathbf{x}_k$ *są generowane w metodzie gradientów sprzężonych, zachodzi*

$$\|\mathbf{x} - \mathbf{x}_k\| \leqslant 2\|\mathbf{x} - \mathbf{x}_1\| \left(\frac{\sqrt{\kappa} - 1}{\sqrt{\kappa} + 1}\right)^{k-1}, \tag{7}$$

gdzie κ *jest współczynnikiem uwarunkowania macierzy* $\mathbf{A}$*.*

Jeżeli $\kappa \gg 1$*, zbieżność może być bardzo wolna.*

“Prewarunkowana” (*preconditioned*) metoda gradientów sprzężonych

Spróbujmy przyspieszyć zbieżność odpowiednio modyfikując równanie (4) i algorytm (5), jednak tak, aby

- nie zmienić rozwiązania,
- macierz zmodyfikowanego układu pozostała symetryczna i dodatnio określona, aby można było zastosować metodę gradientów sprzężonych,
- macierz zmodyfikowanego układu pozostała rzadka, aby jeden krok iteracji był numerycznie tani,
- macierz zmodyfikowanego układu miała niski współczynnik uwarunkowania.

Czy to się w ogóle da zrobić? Okazuje się, że tak!

Postępujemy następująco: Niech $\mathbf{C} \in \mathbb{R}^{N\times N}$ będzie odwracalną macierzą symetryczną, rzeczywistą, dodatnio określoną. Wówczas $\widetilde{\mathbf{A}} = \mathbf{C}^{-1}\mathbf{A}\mathbf{C}^{-1}$ też jest symetryczna, rzeczywista, dodatnio określona.

$$\mathbf{C}^{-1}\mathbf{A}\underbrace{\mathbf{C}^{-1}\mathbf{C}}_{\mathbb{I}}\mathbf{x} = \mathbf{C}^{-1}\mathbf{b}\,, \tag{8a}$$

$$\widetilde{\mathbf{A}}\tilde{\mathbf{x}} = \tilde{\mathbf{b}}\,, \tag{8b}$$

gdzie $\tilde{\mathbf{x}} = \mathbf{C}\mathbf{x}$, $\tilde{\mathbf{b}} = \mathbf{C}^{-1}\mathbf{b}$. Do równania (8b) stosujemy teraz metodę gradientów sprzężonych.

W każdym kroku iteracji musimy obliczyć (tyldy, bo odnosi się to do “tyldowanego” układu (8b))

$$\alpha_k = \frac{\widetilde{\mathbf{r}}_k^T \widetilde{\mathbf{r}}_k}{\widetilde{\mathbf{p}}_k^T \widetilde{\mathbf{A}} \widetilde{\mathbf{p}}_k} = \frac{\widetilde{\mathbf{r}}_k^T \widetilde{\mathbf{r}}_k}{\widetilde{\mathbf{p}}_k^T \mathbf{C}^{-1}\mathbf{A}\mathbf{C}^{-1} \widetilde{\mathbf{p}}_k}\,, \tag{9a}$$

$$\widetilde{\mathbf{r}}_{k+1} = \widetilde{\mathbf{r}}_k - \alpha_k \widetilde{\mathbf{A}} \widetilde{\mathbf{p}}_k = \widetilde{\mathbf{r}}_k - \alpha_k \mathbf{C}^{-1}\mathbf{A}\mathbf{C}^{-1} \widetilde{\mathbf{p}}_k\,, \tag{9b}$$

$$\beta_k = \frac{\widetilde{\mathbf{r}}_{k+1}^T \widetilde{\mathbf{r}}_{k+1}}{\widetilde{\mathbf{r}}_k^T \widetilde{\mathbf{r}}_k}\,, \tag{9c}$$

$$\widetilde{\mathbf{p}}_{k+1} = \widetilde{\mathbf{r}}_{k+1} + \beta_k \widetilde{\mathbf{p}}_k\,, \tag{9d}$$

$$\widetilde{\mathbf{x}}_{k+1} = \widetilde{\mathbf{x}}_k + \alpha_k \widetilde{\mathbf{p}}_k\,. \tag{9e}$$

Równania (9) zawierają jawne odniesienia do macierzy $\mathbf{C}^{-1}$, co nie jest zbyt wygodne. Łatwo się przekonać, iż za pomocą prostych przekształceń macierz tę można „usunąć”, tak, iż pozostaje tylko jedno jej nietrywialne wystąpienie. Zdefiniujmy mianowicie

$$\widetilde{\mathbf{r}}_k = \mathbf{C}^{-1}\mathbf{r}_k\,, \quad \widetilde{\mathbf{p}}_k = \mathbf{C}\mathbf{p}_k\,, \quad \widetilde{\mathbf{x}}_k = \mathbf{C}\mathbf{x}_k\,. \tag{10}$$

W tej sytuacji $\widetilde{\mathbf{r}}_k^T\widetilde{\mathbf{r}}_k = (\mathbf{C}^{-1}\mathbf{r}_k)^T\mathbf{C}^{-1}\mathbf{r}_k = \mathbf{r}_k^T(\mathbf{C}^{-1})^T\mathbf{C}^{-1}\mathbf{r}_k = \mathbf{r}_k^T\mathbf{C}^{-1}\mathbf{C}^{-1}\mathbf{r}_k = \mathbf{r}_k^T(\mathbf{C}^{-1})^2\mathbf{r}_k$ etc.

Wówczas równania (9) przechodzą w

$$\alpha_k = \frac{\mathbf{r}_k^T \left(\mathbf{C}^{-1}\right)^2 \mathbf{r}_k}{\mathbf{p}_k^T \mathbf{A} \mathbf{p}_k}, \tag{11a}$$

$$\mathbf{r}_{k+1} = \mathbf{r}_k - \alpha_k \mathbf{A} \mathbf{p}_k, \tag{11b}$$

$$\beta_k = \frac{\mathbf{r}_{k+1}^T \left(\mathbf{C}^{-1}\right)^2 \mathbf{r}_{k+1}}{\mathbf{r}_k^T \left(\mathbf{C}^{-1}\right)^2 \mathbf{r}_k}, \tag{11c}$$

$$\mathbf{p}_{k+1} = \left(\mathbf{C}^{-1}\right)^2 \mathbf{r}_{k+1} + \beta_k \mathbf{p}_k, \tag{11d}$$

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{p}_k. \tag{11e}$$

W powyższych równaniach rola macierzy $\mathbf{C}$ sprowadza się do obliczenia — *jeden raz w każdym kroku iteracji* — wyrażenia $\left(\mathbf{C}^{-1}\right)^2 \mathbf{r}_k$, co, jak wiadomo, robi się rozwiązując odpowiedni układ równań. Zdefiniujmy

$$\mathbf{M} = \mathbf{C}^2 \,. \tag{12}$$

Macierz $\mathbf{M}$ należy rzecz jasna dobrać tak, aby równanie $\mathbf{Mz} = \mathbf{r}$ można było szybko rozwiązać.

Ostatecznie otrzymujemy następujący algorytm:

$$
\begin{aligned}
&\mathbf{r}_1 = \mathbf{b} - \mathbf{A}\mathbf{x}_1 \\
&\text{rozwiąż } \mathbf{M}\mathbf{z}_1 = \mathbf{r}_1 \\
&\mathbf{p}_1 = \mathbf{z}_1 \\
&\textbf{while } \|\mathbf{r}_k\| > \varepsilon \\
&\qquad \alpha_k = \frac{\mathbf{r}_k^T \mathbf{z}_k}{\mathbf{p}_k^T \mathbf{A}\mathbf{p}_k} \\
&\qquad \mathbf{r}_{k+1} = \mathbf{r}_k - \alpha_k \mathbf{A}\mathbf{p}_k \\
&\qquad \text{rozwiąż } \mathbf{M}\mathbf{z}_{k+1} = \mathbf{r}_{k+1} \\
&\qquad \beta_k = \frac{\mathbf{r}_{k+1}^T \mathbf{z}_{k+1}}{\mathbf{r}_k^T \mathbf{z}_k} \\
&\qquad \mathbf{p}_{k+1} = \mathbf{z}_{k+1} + \beta_k \mathbf{p}_k \\
&\qquad \mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{p}_k \\
&\textbf{end}
\end{aligned}
\tag{13}
$$

Incomplete Cholesky preconditioner

Niech rozkład QR macierzy $\mathbf{C}$ ma postać $\mathbf{C} = \mathbf{Q}\mathbf{H}^T$, gdzie $\mathbf{Q}$ jest macierzą ortogonalną, $\mathbf{H}^T$ jest macierzą trójkątną górną. Zauważmy, że

$$\mathbf{M} = \mathbf{C}^2 = \mathbf{C}^T\mathbf{C} = \left(\mathbf{Q}\mathbf{H}^T\right)^T \mathbf{Q}\mathbf{H}^T = \mathbf{H}\mathbf{Q}^T\mathbf{Q}\mathbf{H}^T = \mathbf{H}\mathbf{H}^T, \qquad (14)$$

a więc macierz $\mathbf{H}$ jest czynnikiem Cholesky'ego macierzy $\mathbf{M}$. Niech rozkład Cholesky'ego macierzy $\mathbf{A}$ ma postać $\mathbf{A} = \mathbf{G}\mathbf{G}^T$. *Przypuśćmy, iż* $\mathbf{H} \simeq \mathbf{G}$.

Wówczas

$$\widetilde{\mathbf{A}} = \mathbf{C}^{-1}\mathbf{A}\mathbf{C}^{-1} = \left(\mathbf{C}^{T}\right)^{-1}\mathbf{A}\mathbf{C}^{-1} = \left(\left(\mathbf{Q}\mathbf{H}^{T}\right)^{T}\right)^{-1}\mathbf{A}\left(\mathbf{Q}\mathbf{H}^{T}\right)^{-1} =$$
$$\left(\mathbf{H}\mathbf{Q}^{T}\right)^{-1}\mathbf{A}\left(\mathbf{H}^{T}\right)^{-1}\mathbf{Q}^{T} = \mathbf{Q}\underbrace{\mathbf{H}^{-1}\mathbf{G}}_{\simeq\mathbb{I}}\underbrace{\mathbf{G}^{T}\left(\mathbf{H}^{T}\right)^{-1}}_{\simeq\mathbb{I}}\mathbf{Q}^{T} \simeq \mathbf{Q}\mathbf{Q}^{T} = \mathbb{I}\,. \quad (15)$$

Ponieważ $\widetilde{\mathbf{A}} \simeq \mathbb{I}$, współczynnik uwarunkowania tej macierzy powinien być bliski jedności.

Niepełny rozkład Cholesky'ego — algorytm w wersji *GAXPY*

for $k = 1{:}N$
  $H_{kk} = A_{kk}$
  for $j = 1{:}k-1$
    $H_{kk} = H_{kk} - H_{kj}^2$
  end
  $H_{kk} = \sqrt{H_{kk}}$
  for $l = k+1{:}N$
    $H_{lk} = A_{lk}$
    if $A_{lk} \neq 0$
      for $j = 1{:}k-1$
        $H_{lk} = H_{lk} - H_{lj}H_{kj}$
      end
      $H_{lk} = H_{lk}/H_{kk}$
    endif
  end
end

Uwagi

- Ponieważ $\mathbf{A}$ jest rzadka, powyższy algorytm na obliczanie przybliżonego czynnika Cholesky'ego wykonuje się szybko. Wykonuje się go *tylko raz*.
- Równanie $\mathbf{Mz} = \mathbf{r}$ rozwiązuje się szybko, gdyż znamy czynnik Cholesky'ego $\mathbf{M} = \mathbf{HH}^T$.
- Obliczone $\mathbf{H}$ jest rzadkie, a zatem równanie $\mathbf{Mz} = \mathbf{r}$ rozwiązuje się szczególnie szybko.
- Mamy nadzieję, że macierz $\tilde{\mathbf{A}}$ ma współczynnik uwarunkowania bliski jedności, a zatem nie potrzeba wielu iteracji (13).

Przykład — macierz pasmowa z pustymi diagonalami

Rozważmy macierz o następującej strukturze:

$$\mathbf{A} = \begin{bmatrix} a_1 & 0 & b_3 & 0 & c_5 & 0 & 0 & 0 & 0 & 0 & \dots \\ 0 & a_2 & 0 & b_4 & 0 & c_6 & 0 & 0 & 0 & 0 & \dots \\ b_3 & 0 & a_3 & 0 & b_5 & 0 & c_7 & 0 & 0 & 0 & \dots \\ 0 & b_4 & 0 & a_4 & 0 & b_6 & 0 & c_8 & 0 & 0 & \dots \\ c_5 & 0 & b_5 & 0 & a_5 & 0 & b_7 & 0 & c_9 & 0 & \dots \\ 0 & c_6 & 0 & b_6 & 0 & a_6 & 0 & b_8 & 0 & c_{10} & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \end{bmatrix} \tag{16}$$

Macierz ta jest symetryczna, zakładamy też, że jest dodatnio określona.

Niepełny czynnik Cholesky'ego macierzy (16) ma postać

$$\mathbf{H} = \begin{bmatrix} p_1 & & & & & & \\ 0 & p_2 & & & & & \\ q_3 & 0 & p_3 & & & & \\ 0 & q_4 & 0 & p_4 & & & \\ r_5 & 0 & q_5 & 0 & p_5 & & \\ 0 & r_6 & 0 & q_6 & 0 & p_6 & \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \end{bmatrix} \tag{17}$$

(W pełnym czynniku Cholesky'ego macierzy (16) zera leżące w (17) *pomiędzy* diagonalą "p" a diagonalną "r" znikłyby — w ogólności mogłyby tam znajdować się jakieś niezerowe liczby.)

Zgodnie z podanym algorytmem, elementy ciągów $\{p_k\}$, $\{q_k\}$, $\{r_k\}$ wyliczamy z następujących wzorów:

$$
\begin{array}{rcl@{\qquad}rcl}
p_1 &=& \sqrt{a_1}, & p_2 &=& \sqrt{a_2}, \\
q_3 &=& b_3/p_1, & q_4 &=& b_4/p_2, \\
r_5 &=& c_5/p_1, & r_6 &=& c_6/p_2, \\
p_3 &=& \sqrt{a_3 - q_3^2}, & p_4 &=& \sqrt{a_4 - q_4^2}, \\
q_5 &=& (b_5 - r_5 q_3)/p_3, & q_6 &=& (b_6 - r_6 q_4)/p_4, \\
r_7 &=& c_7/p_3, & r_8 &=& c_7/p_4, \\
p_5 &=& \sqrt{a_5 - q_5^2 - r_5^2}, & p_6 &=& \sqrt{a_6 - q_5^6 - r_6^2}, \\
q_7 &=& (b_7 - r_7 q_5)/p_5, & q_8 &=& (b_8 - r_8 q_6)/p_6, \\
r_9 &=& c_9/p_5, & r_{10} &=& c_{10}/p_6, \\
p_7 &=& \sqrt{a_7 - q_7^2 - r_7^2}, & p_8 &=& \sqrt{a_8 - q_8^6 - r_8^2}, \\
q_9 &=& (b_9 - r_9 q_7)/p_7, & q_{10} &=& (b_{10} - r_{10} q_8)/p_8, \\
r_{11} &=& c_{11}/p_7, & r_{12} &=& c_{12}/p_8, \\
\ldots & \ldots & \ldots & \ldots & \ldots & \ldots
\end{array}
$$

Macierze niesymetryczne

Jeżeli w równaniu

$$\mathbf{A}\mathbf{x} = \mathbf{b} \tag{18}$$

macierz $\mathbf{A}$ nie jest symetryczna i dodatnio określona, sytuacja się komplikuje. Zakładając, że $\det \mathbf{A} \neq 0$, równanie (18) możemy "zsymetryzować" na dwa sposoby.
CGNR:

$$\mathbf{A}^T\mathbf{A}\mathbf{x} = \mathbf{A}^T\mathbf{b}\,, \tag{19}$$

lub CGNE:

$$\mathbf{A}\mathbf{A}^T\mathbf{y} = \mathbf{b}\,, \tag{20a}$$

$$\mathbf{x} = \mathbf{A}^T\mathbf{y}\,. \tag{20b}$$

Do dwu powyższych równań formalnie rzecz biorąc *można* używać metody gradientów sprzężonych. Trzeba jednak pamiętać, że nawet jeśli macierz $\mathbf{A}$ jest rzadka, macierze $\mathbf{A}^T\mathbf{A}$, $\mathbf{A}\mathbf{A}^T$ nie muszą być rzadkie, a co gorsza, ich współczynnik uwarunkowania jest kwadratem współczynnika uwarunkowania macierzy wyjściowej.

Alternatywnie, zamiast "symetryzować" macierz, można zmodyfikować algorytm, tak aby zamiast dwu, generował on *cztery* ciągi wektorów (*metoda gradientów bi-sprzężonych*). Należy jednak pamiętać, że dla wielu typów macierzy taki algorytm bywa bardzo wolno zbieżny, a niekiedy nawet dochodzi do kompletnej stagnacji przed uzyskaniem rozwiązania.

Strategia minimalizacji wielowymiarowej

Zakładamy, że metody poszukiwania minimów lokalnych fukncji jednej zmiennej są znane.

Rozważmy funkcję $f:\mathbb{R}^N \to \mathbb{R}$. Przedstawimy strategię poszukiwania (lokalnego) minimum tej funkcji w postaci ciągu minimalizacji jednowymiarowych.

1. Aktualnym przybliżeniem minimum jest punkt $\mathbf{x}_k$.
2. Dany jest pewien kierunek poszukiwań $\mathbf{p}_k$.
3. Konstruujemy funcję g_k:$\mathbb{R} \to \mathbb{R}$

$$g_k(\alpha) = f(\mathbf{x}_k + \alpha \mathbf{p}_k)\,. \tag{21}$$

4. Znanymi metodami jednowymiarowymi znajdujemy α_{min} takie, że funkcja (21) osiąga minimum. Jest to *minimum kierunkowe* funkcji f.
5.

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_{\text{min}} \mathbf{p}_k\,. \tag{22}$$

6. *goto* 1.

Jak wybierać kierunki poszukiwań?

Cały problem sprowadza się zatem do wyboru odpowiedniej sekwencji kolejnych kierunków $\{\mathbf{p}_k\}$. Bardzo popularne są metody

- Minimalizacji po współrzędnych — kolejnymi kierunkami poszukiwań są kierunki równoległe do kolejnych osi układu współrzędnych.

- *Metoda najszybszego spadku*, w której kierunek poszukiwań pokrywa się z minus gradientem minimalizowanej funkcji w punkcie startowym.

Jeśli jesteśmy blisko minimum nie są to dobre pomysły, gdyż prowadzą do wielu drobnych kroków, które częściowo likwidują efekty osiągnięte w krokach poprzednich. Dlaczego?

Znajdźmy warunek na to, aby f osiągała minimum kierunkowe, czyli aby g_k osiągała minimum:

$$\frac{dg_k}{d\alpha} = \sum_i \frac{\partial f}{\partial x_i}(\mathbf{p}_k)_i = \left(\nabla f|_{\mathbf{x}=\mathbf{x}_{\text{min}}}\right)^T \mathbf{p}_k = 0\,. \tag{23}$$

W minimum kierunkowym gradient funkcji jest prostopadły do kierunku poszukiwań. Widać stąd natychmiast, że krok minimalizacji w metodzie najszybszego spadku co prawda zaczyna się prostopadle do poziomnic funkcji, ale kończy się *stycznie* do poziomnic. Z kolei w minimalizacji po współrzędnych kolejne kierunki poszukiwań, czyli — tutaj —- kolejne współrzędne, nie zależą od kształtu minimalizowanej funkcji; taka strategia nie może być optymalna.

Przybliżenie formy kwadratowej

Przypuśćmy, że jesteśmy dostatecznie blisko minimum. Rozwijamy minimalizowaną funkcję w szereg Taylora wokół minimum i otrzymujemy

$$f(\mathbf{x}) \simeq \frac{1}{2}\mathbf{x}^T\mathbf{A}\mathbf{x} - \mathbf{b}^T\mathbf{x} + f_0\,, \tag{24}$$

gdzie $\mathbf{A}$ jest macierzą drugich pochodnych cząstkowych (hessjanem) obliczanym w minimum. Z definicji minimum, macierz ta jest dodatnio określona, jeśli zaś funkcja jest dostatecznie gładka, macierz ta jest symetryczna. Zatem w pobliżu minimum, funkcja w przybliżeniu zachowuje się jak dodatnio określona forma kwadratowa.

Gradienty sprzężone

W przybliżeniu (24) gradient funkcji f w punkcie $\mathbf{x}_k$ wynosi

$$\nabla f|_{\mathbf{x}=\mathbf{x}_k} = \mathbf{A}\mathbf{x}_k - \mathbf{b}\,. \tag{25}$$

Kolejne poszukiwania odbywają się w kierunku $\mathbf{p}_{k+1}$. Gradient funkcji w pewnym nowym punkcie $\mathbf{x} = \mathbf{x}_k + \alpha\mathbf{p}_{k+1}$ wynosi

$$\nabla f|_{\mathbf{x}} = \mathbf{A}\mathbf{x}_k + \alpha\mathbf{A}\mathbf{p}_{k+1} - \mathbf{b}\,. \tag{26}$$

Zmiana gradientu wynosi

$$\delta\left(\nabla f\right) = \alpha\mathbf{A}\mathbf{p}_{k+1}\,. \tag{27}$$

Punkt $\mathbf{x}_k$ jest minimum kierunkowym w kierunku $\mathbf{p}_k$, a więc gradient funkcji w tym punkcie spełnia warunek (23). *Jeżeli chcemy aby poszukiwania w nowym kierunku nie zepsuły minimum kierunkowego w kierunku $\mathbf{p}_k$, zmiana gradientu musi być prostopadła do starego kierunku poszukiwań*, $\delta\,(\nabla f)^T\,\mathbf{p}_k = 0$. Tak jednak musi być dla *wszystkich* poprzednich kierunków, nie chcemy bowiem naruszyć żadnego z poprzednich minimów kierunkowych. A zatem

$$\mathbf{p}_i^T \mathbf{A} \mathbf{p}_j = 0\,, \quad i > j\,. \tag{28}$$

Metodę wybierania kierunków poszukiwań spełniających (28) nazywamy *metodą gradientów sprzężonych*.

Jak się obejść bez hessjanu?

Z poprzedniej części wykładu znamy *algebraiczną* metodę gradientów sprzężonych, wydaje się zatem, iż moglibyśmy jej użyć do skonstruowania ciągu wektorów $\{\mathbf{p}_k\}$. Niestety, nie możemy, nie znamy bowiem macierzy $\mathbf{A}$, czyli hessjanu w minimum. Czy możemy się bez tego obejść?

Twierdzenie 4. *Niech f ma postać* (24) *i niech $\mathbf{r}_k = -\nabla f|_{\mathbf{x}_k}$. Z punktu $\mathbf{x}_k$ idziemy w kierunku $\mathbf{p}_k$ do punktu, w którym f osiąga minimum kierunkowe. Oznaczmy ten punkt $\mathbf{x}_{k+1}$. Wówczas $\mathbf{r}_{k+1} = -\nabla f|_{\mathbf{x}_{k+1}}$ jest* tym samym *wektorem, który zostałby skonstruowany w algebraicznej metodzie gradientów sprzężonych.*

Dowód. Na podstawie równania (25), $\mathbf{r}_k = -\mathbf{A}\mathbf{x}_k + \mathbf{b}$ oraz

$$\mathbf{r}_{k+1} = -\mathbf{A}(\mathbf{x}_k + \alpha\mathbf{p}_k) + \mathbf{b} = \mathbf{r}_k - \alpha\mathbf{A}\mathbf{p}_k\,. \tag{29}$$

W minimum kierunkowym $\mathbf{p}_k^T\,\nabla f|_{\mathbf{x}_{k+1}} = -\mathbf{p}_k^T\mathbf{r}_{k+1} = 0$ (por. (23)). Wobec tego mnożąc równanie (29) lewostronnie przez $\mathbf{p}_k^T$, otrzymujemy

$$\alpha = \frac{\mathbf{p}_k^T\mathbf{r}_k}{\mathbf{p}_k^T\mathbf{A}\mathbf{p}_k}\,. \tag{30}$$

Ponieważ w algebraicznej metodzie gradientów sprzężonych $\mathbf{r}_k^T\mathbf{p}_k = \mathbf{r}_k^T\mathbf{r}_k$, otrzymujemy *dokładnie takie samo* α jak we wzorach na metodę algebraiczną, co kończy dowód. □

Algorytm gradientów sprzężonych

Rozpoczynamy w pewnym punkcie $\mathbf{x}_1$. Bierzemy $\mathbf{r}_1 = \mathbf{p}_1 = -\nabla f|_{\mathbf{x}_1}$.

1. Będąc w punkcie $\mathbf{x}_k$, dokonujemy minimalizacji kierunkowej w kierunku $\mathbf{p}_k$; osiągamy punkt $\mathbf{x}_{k+1}$.
2. Obliczamy $\mathbf{r}_{k+1} = -\nabla f|_{\mathbf{x}_{k+1}}$.
3. Obliczamy (jak w metodzie algebraicznej)

$$\beta = \frac{\mathbf{r}_{k+1}^T \mathbf{r}_{k+1}}{\mathbf{r}_k^T \mathbf{r}_k} \,. \tag{31}$$

4. Obliczamy (jak w metodzie algebraicznej) $\mathbf{p}_{k+1} = \mathbf{r}_{k+1} + \beta \mathbf{p}_k$.

Zamiast używać równania (31), można skorzystać z

$$\beta = \frac{\mathbf{r}_{k+1}^T(\mathbf{r}_{k+1} - \mathbf{r}_k)}{\mathbf{r}_k^T \mathbf{r}_k}. \tag{32}$$

Jeżeli funkcja f ma *ściśle* postać (24), nie ma to znaczenia, gdyż $\mathbf{r}_{k+1}^T \mathbf{r}_k = 0$. Ponieważ jednak f jest tylko w przybliżeniu formą kwadratową, (32) może przyspieszyć obliczenia gdy grozi stagnacja.

Układy równań algebraicznych

Poszukiwanie ekstremów funkcji wielu zmiennych *formalnie* sprowadza się do rozwiązywania równania $\nabla f = 0$. *W praktyce minimalizacja funkcji poprzez rozwiązywanie tego równania jest bardzo złą metodą*, jest to jednak dobra okazja do przypomnienia najważniejszych metod rozwiązywania nieliniowych układów równań algebraicznych.

Niech $\mathbf{g}$:$\mathbb{R}^N \to \mathbb{R}^N$ będzie funkcją klasy co najmniej C^1. Rozważamy równanie

$$\mathbf{g}(\mathbf{x}) = 0\,, \tag{33}$$

formalnie równoważne układowi równań

$$\begin{aligned} g_1(x_1, x_2, \ldots, x_N) &= 0\,, && \text{(34a)}\\ g_2(x_1, x_2, \ldots, x_N) &= 0\,, && \text{(34b)}\\ &\ldots \\ g_N(x_1, x_2, \ldots, x_N) &= 0\,. && \text{(34c)} \end{aligned}$$

Metoda Newtona

Rozwijając funkcję $\mathbf{g}$ w szereg Taylora do pierwszego rzędu otrzymamy

$$\mathbf{g}(\mathbf{x}+\delta\mathbf{x}) \simeq \mathbf{g}(\mathbf{x}) + \mathbf{J}\delta\mathbf{x}\,, \tag{35}$$

gdzie $\mathbf{J}$ jest jakobianem funkcji $\mathbf{g}$:

$$\mathbf{J}(\mathbf{x})_{ij} = \left.\frac{\partial g_i}{\partial x_j}\right|_{\mathbf{x}}\,. \tag{36}$$

Jaki krok $\delta\mathbf{x}$ musimy wykonać, aby znaleźć się w punkcie spełniającym równanie (33)? Żądamy aby $\mathbf{g}(\mathbf{x}+\delta\mathbf{x}) = 0$, skąd otrzymujemy

$$\delta\mathbf{x} = -\mathbf{J}^{-1}\mathbf{g}(\mathbf{x})\,. \tag{37}$$

Prowadzi to do następującej iteracji:

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \mathbf{J}^{-1}(\mathbf{x}_k)\mathbf{g}(\mathbf{x}_k)\,. \tag{38}$$

Oczywiście zapis $\mathbf{z} = \mathbf{J}^{-1}\mathrm{g}$ należy rozumieć w ten sposób, że $\mathbf{z}$ spełnia równanie $\mathbf{J}\mathbf{z} = \mathrm{g}$. *Nie należy konstruować jawnej odwrotności jakobianu.*

Uwaga: W metodzie (38) jakobian trzeba obliczać w każdym kroku. Oznacza to, że w każdym kroku trzeba rozwiązywać *inny* układ równań liniowych, co czyni metodę dość kosztowną, zwłaszcza jeśli N (wymiar problemu) jest znaczne. Często dla przyspieszenia obliczeń macierz $\mathbf{J}$ zmieniamy nie co krok, ale co kilka kroków — pozwala to użyć tej samej faktoryzacji $\mathbf{J}$ do rozwiązania kilku kolejnych równań $\mathbf{J}\mathbf{z} = \mathrm{g}(\mathbf{x}_k)$. Jest to dodatkowe uproszczenie, ale jest ono bardzo wydajne przy $N \gg 1$.

Metoda Levenberga-Marquardta

Powracamy do zagadnienia minimalizacji funkcji.

Metoda gradientów sprzężonych jest dostosowana do przypadku, w którym funkcja jest z dobrym przybliżeniem postaci (24), a więc gdy jesteśmy dostatecznie blisko poszukiwanego minimum. Jednak daleko od minimum metody te są powolne — trudno oczekiwać, że wzory słuszne dla formy kwadratowej będą dobrze działać gdy funkcja formą kwadratową *nie* jest. Daleko od minimum jest sens stosować metodę najszybszego spadku. Powinniśmy zatem mieć metodę, która daleko od minimum zachowuje się jak najszybszy spadek, blisko zaś minimum redukuje się do metody gradientów sprzężonych.

Konieczność formalnego rozwiązywania równania $\nabla \mathbf{f} = 0$ sugeruje użycie następującego "algorytmu" opartego na metodzie Newtona:

$$\mathbf{x}_{i+1} = \mathbf{x}_i - \mathbf{H}^{-1}(\mathbf{x}_i)\, \nabla f|_{\mathbf{x}_i} \,,$$

gdzie $\mathbf{H}$ oznacza hessjan (macierz drugich pochodnych) minimalizowanej funkcji. Daleko od minimum hessjan nie musi być nawet dodatnio określony, co powoduje, iż krok newtonowski wcale nie musi prowadzić do spadku wartości funkcji. My jednak chcemy, aby wartość funkcji w kolejnych krokach spadała. Zmodyfikujmy więc hessjan:

$$\widetilde{\mathbf{H}}_{ii} = (1+\lambda)\frac{\partial^2 f}{\partial x_i^2}\,, \tag{39a}$$

$$\widetilde{\mathbf{H}}_{ij} = \frac{\partial^2 f}{\partial x_i \partial x_j}\,, \quad i \neq j\,, \tag{39b}$$

przy czym $\lambda \geqslant 0$.

Zauważmy, że zachodzi jedna z dwu możliwych sytuacji: (i) jeśli znajdujemy się w basenie atrakcji minimum, wówczas dla odpowiednio dużego λ macierz (39) stanie się dodatnio określona lub też (ii) jeśli dla żadnego dodatniego λ macierz (39) nie staje się dodatnio określona, znajdujemy się na monotonicznej gałęzi funkcji, poza basenem atrakcji minimum.

Rozpoczynamy z jakimś niewielkim λ, na przykład $\lambda = \lambda_0 = 2^{-10} = 1/1024$. Przypuśćmy, iż aktualnym przybliżeniem minimum jest punkt $\mathbf{x}_i$. Dostajemy zatem...

verte

Algorytm Levenberga-Marquardta

1. Oblicz $\nabla f(\mathbf{x}_i)$.
2. Oblicz $\widetilde{\mathbf{H}}(\mathbf{x}_i)$.
3. Oblicz

$$\mathbf{x}_{\text{test}} = \mathbf{x}_i - \widetilde{\mathbf{H}}^{-1}(\mathbf{x}_i)\nabla f(\mathbf{x}_i)\,. \tag{40}$$

4. Jeżeli $f(\mathbf{x}_{\text{test}}) > f(\mathbf{x}_i)$, to
   (a) $\lambda \rightarrow 8\lambda$ (można też powiększać o inny znaczny czynnik).
   (b) Idź do punktu 2.
5. Jeżeli $f(\mathbf{x}_{\text{test}}) < f(\mathbf{x}_i)$, to
   (a) $\lambda \rightarrow \lambda/8$ (można też zmniejszać o inny znaczny czynnik).
   (b) $\mathbf{x}_{i+1} = \mathbf{x}_{\text{test}}$.
   (c) Idź do punktu 1.

Komentarz

Dodatkowo, jeśli $\lambda > \lambda_{\text{max}} \gg 1$, uznajemy, iż znajdujemy się poza basenem atrakcji minimum i algorytm zawodzi. Jeśli natomiast $\lambda < \lambda_{\text{min}} \ll 1$, macierz $\widetilde{\mathbf{H}}$ jest w praktyce równa hessjanowi, a zatem modyfikacja (39) przestaje być potrzebna. Możemy wówczas przerzucić się na metodę gradientów sprzężonych lub metodę zmiennej metryki aby wykorzystać ich szybką zbieżność w pobliżu minimum, gdzie funkcja ma postać (24).

Ponadto w celu przyspieszenia obliczeń, jeżeli $f(\mathbf{x}_{\text{test}}) < f(\mathbf{x}_i)$, możemy *chwilowo* zrezygnować ze zmniejszania λ i modyfikowania $\widetilde{\mathbf{H}}$ i przeprowadzić kilka kroków z tą samą macierzą, a więc korzystając z tej samej faktoryzacji.

Zauważmy, iż przypadek $\lambda \gg 1$ oznacza, iż jesteśmy daleko od minimum. Z drugiej strony jeśli $\lambda \gg 1$, macierz $\widetilde{\mathbf{H}}$ staje się w praktyce diagonalna, a zatem

$$\begin{aligned} \mathbf{x}_{\text{test}} &\simeq \mathbf{x}_i - (1+\lambda)^{-1}\text{diag}\left\{\left(\frac{\partial^2 f}{\partial x_1^2}\right)^{-1}, \left(\frac{\partial^2 f}{\partial x_2^2}\right)^{-1}, \ldots, \left(\frac{\partial^2 f}{\partial x_N^2}\right)^{-1}\right\}\nabla f(\mathbf{x}_i) \\ &\simeq \mathbf{x}_i - \text{const}\,\nabla f(\mathbf{x}_i)\,, \end{aligned} \tag{41}$$

o ile drugie pochodne cząstkowe w poszczególnych kierunkach nie różnią się znacznie od siebie. Widać, iż daleko od minimum, gdzie warunek zachowujący raz osiągnięte minima kierunkowe nie zachodzi, algorytm Levenberga-Marquardta zachowuje się prawie jak metoda najszybszego spadku.

Rozwiązywanie równań nieliniowych a minimalizacja

Wracamy do problemu rozwiązywania równań algebraicznych, danego równaniem (33):

$$\mathbf{g}(\mathbf{x}) = 0\,.$$

Zaproponowana powyżej metoda Newtona czasami zawodzi ☹. Ponieważ rozwiązywanie równań algebraicznych jest "trudne", natomiast minimalizacja jest "łatwa", niektórzy skłonni są rozważać funkcję $G{:}\mathbb{R}^N \to \mathbb{R}$

$$G(\mathbf{x}) = \frac{1}{2}\,\|\mathbf{g}(\mathbf{x})\|^2 = \frac{1}{2}\,(\mathbf{g}(\mathbf{x}))^T\,\mathbf{g}(\mathbf{x}) \tag{42}$$

i szukać jej minimum zamiast rozwiązywać (33). *Globalne* minimum $G = 0$ odpowiada co prawda rozwiązaniu (33), jednak G może mieć wiele minimów lokalnych, *nie mamy także gwarancji*, że globalne minimum $G = 0$ istnieje. Nie jest to więc dobry pomysł.

Metoda globalnie zbieżna

Rozwiązaniem jest połączenie idei minimalizacji funkcji (42) i metody Newtona. Przypuśćmy, iż chcemy rozwiązywać równanie (33) metodą Newtona. Krok iteracji wynosi

$$\delta\mathbf{x} = -\mathbf{J}^{-1}\mathbf{g}\,. \tag{43}$$

Z drugiej strony mamy

$$\frac{\partial G}{\partial x_i} = \frac{1}{2}\sum_j \left(\frac{\partial g_j}{\partial x_i}g_j + g_j\frac{\partial g_j}{\partial x_i}\right) = \sum_j J_{ji}g_j \tag{44}$$

a zatem $\nabla G = \mathbf{J}^T\mathbf{g}$.

Jak zmienia się funkcja G (42) po wykonaniu kroku Newtona (43)?

$$(\nabla G)^T \, \delta\mathbf{x} = \mathbf{g}^T \mathbf{J} \left(-\mathbf{J}^{-1}\right) \mathbf{g} = -\mathbf{g}^T \mathbf{g} < 0 \,, \tag{45}$$

a zatem *kierunek kroku Newtona jest* lokalnym *kierunkiem spadku* G. Jednak przesunięcie się o pełną długość kroku Newtona nie musi prowadzić do spadku G. Postępujemy wobec tego jak następuje:

1. $w = 1$. Oblicz $\delta\mathbf{x}$.
2. $\mathbf{x}_{\text{test}} = \mathbf{x}_i + w\,\delta\mathbf{x}$.
3. Jeśli $G(\mathbf{x}_{\text{test}}) < G(\mathbf{x}_i)$, to
   (a) $\mathbf{x}_{i+1} = \mathbf{x}_{\text{test}}$
   (b) *goto* 1
4. Jeśli $G(\mathbf{x}_{\text{test}}) > G(\mathbf{x}_i)$, to
   (a) $w \to w/2$
   (b) *goto* 2

Jest to zatem forma *tłumionej (*damped*) metody Newtona*.

Zamiast połowienia kroku, można używać innych strategii poszukiwania w prowadzących do zmniejszenia się wartości G.

Jeśli wartość w spadnie poniżej pewnego akceptowalnego progu, obliczenia należy przerwać, jednak (45) gwarantuje, że istnieje takie w, iż $w\,\delta\mathbf{x}$ prowadzi do zmniejszenia się G.

Bardzo ważna uwaga

Wszystkie przedstawione tu metody wymagają znajomości analitycznych wzorów na pochodne odpowiednich funkcji.

Używanie powyższych metod w sytuacji, w których pochodne należy aproksymować numerycznie, *na ogół nie ma sensu*.

Wielowymiarowa metoda siecznych — metoda Broydena

Niekiedy analityczne wzory na pochodne są nieznane, niekiedy samo obliczanie jakobianu, wymagające obliczenia N^2 pochodnych cząstkowych, jest numerycznie zbyt kosztowne. W takich sytuacjach *czasami* używa się metody zwanej niezbyt ściśle "wielowymiarową metodą siecznych". Podobnie jak w przypadku jednowymiarowym, gdzie pochodną zastępuje się ilorazem różnicowym

$$g'(x_{i+1}) \simeq \frac{g(x_{i+1}) - g(x_i)}{x_{i+1} - x_i}\,, \tag{46}$$

jakobian w kroku Newtona zastępujemy wyrażeniem przybliżonym: Zamiast $\mathbf{J}\,\delta\mathbf{x} = -\mathbf{g}(\mathbf{x})$ bierzemy $\mathbf{B}\,\Delta\mathbf{x} = -\Delta\mathbf{g}$. Macierz $\mathbf{B}$ jest przybliżeniem jakobianu, poprawianym w każdym kroku iteracji. Otrzymujemy zatem

$$\mathbf{x}_{i+1} = \mathbf{x}_i - \mathbf{B}_i^{-1}\mathbf{g}(\mathbf{x}_i)\,, \tag{47}$$

natomiast poprawki $\mathbf{B}$ obliczamy jako

$$\mathbf{B}_{i+1} = \mathbf{B}_i + \frac{(\Delta\mathbf{g}_i - \mathbf{B}_i\Delta\mathbf{x}_i)\,(\Delta\mathbf{x}_i)^T}{(\Delta\mathbf{x}_i)^T\,\Delta\mathbf{x}_i}\,, \tag{48}$$

gdzie $\Delta\mathbf{x}_i = \mathbf{x}_{i+1} - \mathbf{x}_i$, $\Delta\mathbf{g}_i = \mathbf{g}(\mathbf{x}_{i+1}) - \mathbf{g}(\mathbf{x}_i)$. Ponieważ poprawka do $\mathbf{B}_i$ ma postać iloczynu diadycznego dwu wektorów, $\mathbf{B}_{i+1}^{-1}$ można obliczać korzystając ze wzoru Shermana-Morrisona.

Metoda ta wymaga inicjalizacji poprzez podanie $\mathbf{B}_1$ oraz wektora $\mathbf{x}_1$. To drugie nie jest niczym dziwnym; co do pierwszego, jeśli to tylko możliwe, można przyjąć $\mathbf{B}_1 = \mathbf{J}(\mathbf{x}_1)$.