Wstęp do metod numerycznych

12. Minimalizacja: funkcje wielu zmiennych

P. F. Góra

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

2018

Strategia minimalizacji wielowymiarowej

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

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

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

$$g_k(\alpha) = f(\mathbf{x}_k + \alpha \mathbf{p}_k)\,. \tag{1}$$

   Zauważmy, że funkcja $g_k(\alpha)$ jest funkcją jednej zmiennej. Geometrycznie jest to "ślad" $N+1$-wymiarowego wykresu funkcji $f(\mathbf{x})$ przeciętego płaszczyzną zawierającą punkt $\mathbf{x}_k$ i wektor $\mathbf{p}_k$.
4. Znanymi metodami jednowymiarowymi znajdujemy α_{min} takie, że funkcja (1) osiąga minimum. Jest to *minimum kierunkowe* funkcji f.
5.

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_{\text{min}} \mathbf{p}_k\,. \tag{2}$$

6. *goto* 1.

Przykład

Niech $f : \mathbb{R}^2 \to \mathbb{R}$, $f(x_1, x_2) = 2.5x_1^2 + x_1x_2 + x_2^2 - x_1 - x_2$. Przypuśćmy, że dany jest punkt $\mathbf{x}_k = (1, 2)$ oraz kierunek poszukiwań $\mathbf{p}_k = [-1, 1]^T$. Wówczas

$$\begin{aligned} g_k(\alpha) &= f(\mathbf{x}_k + \alpha\mathbf{p}_k) \\ &= 2.5(1-\alpha)^2 + (1-\alpha)(2+\alpha) + (2+\alpha)^2 - (1-\alpha) - (2+\alpha) \\ &= 2.5\alpha^2 - 2\alpha + 5.5 \end{aligned} \tag{3}$$

Jest to funkcja jednej zmiennej, α. Osiąga ona minimum dla $\alpha = \frac{2}{5}$. Zatem *następnym* punktem będzie

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \frac{2}{5}\mathbf{p}_k = \left(1 - \frac{2}{5}, 2 + \frac{2}{5}\right) = (0.6, 2.4) \tag{4}$$

Wystarczy teraz wybrać kolejny kierunek poszukiwań $\mathbf{p}_{k+1}$ etc.

Jak wybierać kierunki poszukiwań?

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

Okazuje się, że inaczej powinniśmy wybierać kierunki poszukiwań gdy (spodziewamy się, że) jesteśmy blisko poszukiwanego minimum, inaczej zaś, gdy jesteśmy daleko od minimum.

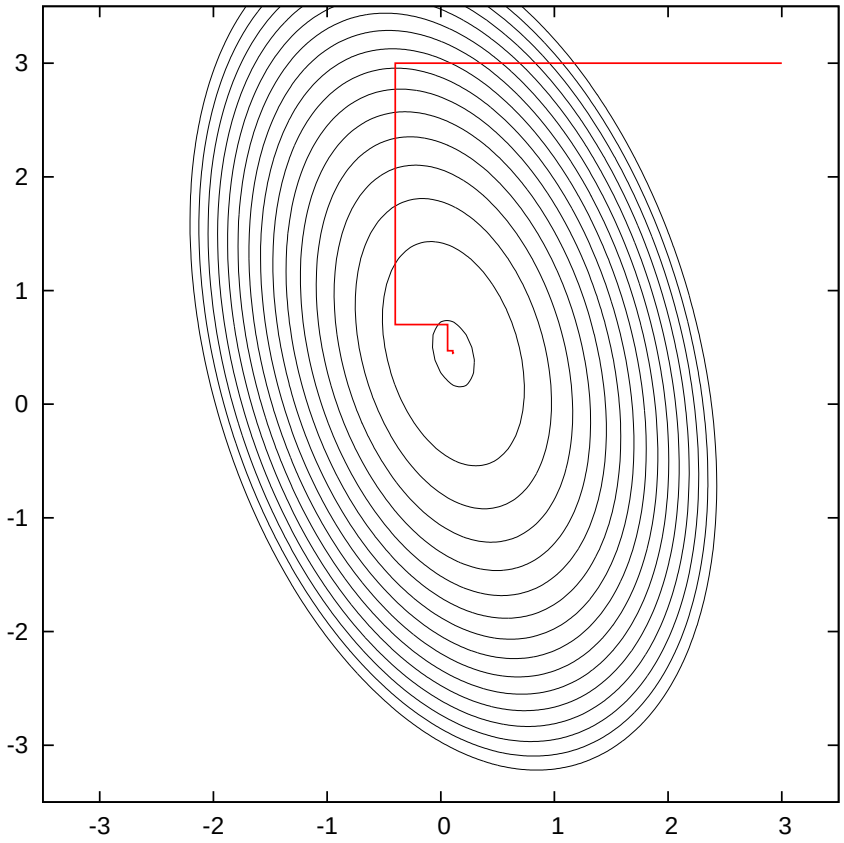
Następujące strategie wyboru kierunkow poszukiwań są bardzo popularne:

- 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 aktualnym punkcie.

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

Przykład — minimalizacja po współrzędnych

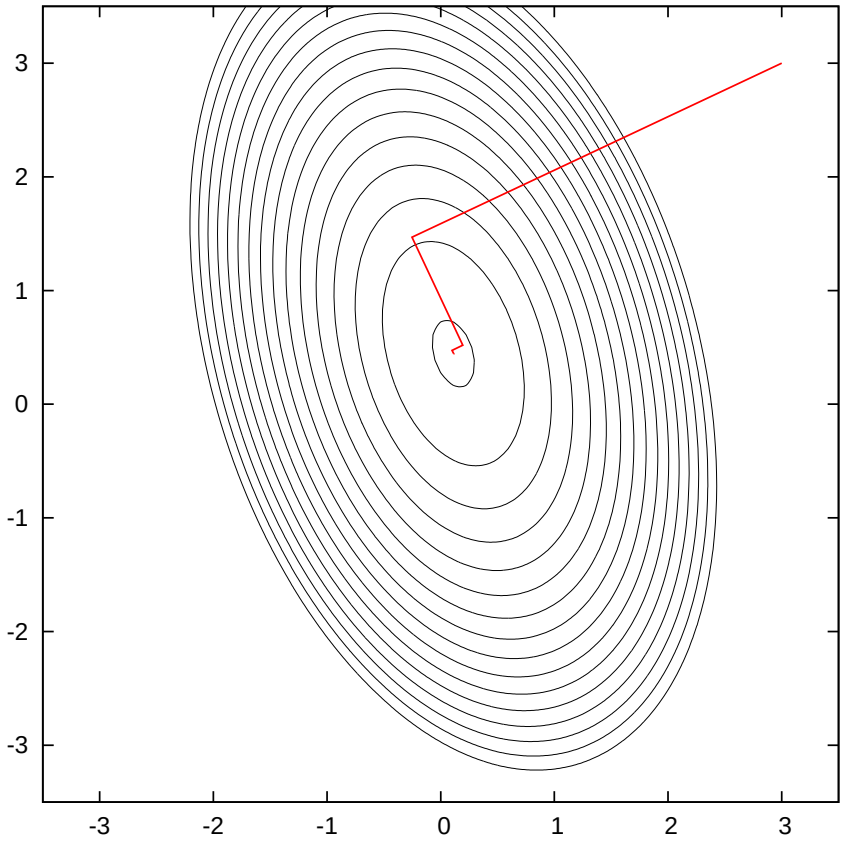


Dużo małych kroków. Osie układu nijak się mają do “naturalnej” geometrii minimalizowanej funkcji.

W metodzie najszybszego spadku co prawda realizujemy najważniejszy pomysł na minimalizację —— zawsze podążamy w kierunku *malejących* wartości (czyli kierunkiem poszukiwań jest minus gradient funkcji w aktualnym punkcie) —— ale czasami poszukiwanie minimów kierunkowych bywa niepotrzebnie kosztowne. W dodatku, jeśli jesteśmy już blisko minimum, okazuje się, że metoda najszybszego spadku także prowadzi do konieczności wykonywania wielu małych kroków, które częściowo niwelują efekty osiągnięte w krokach poprzednich.

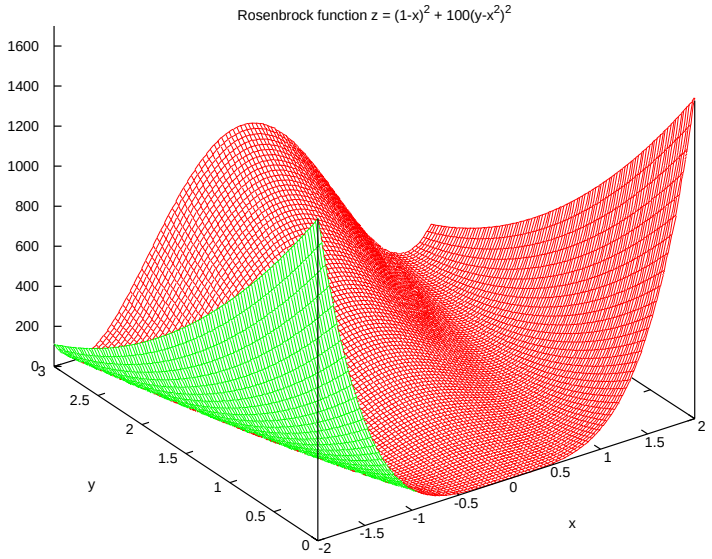
Dlatego gdy jestesmy daleko od minimum, *nie minimalizujemy*, czyli nie poszukujemy *minimów* kierunkowych, a jedynie idziemy w kierunku najszybszego spadku funkcji, nie starając sięe osiągnąć minimum kierunkowego. Gdy zaś jesteśmy blisko minimum, potrzebujemy jakiejś lepszej, bardziej wyrafinowanej metody

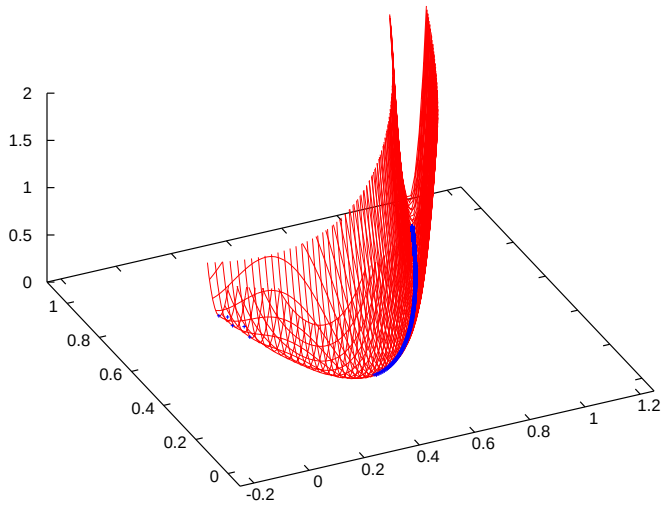
Przykład — metoda najszybszego spadku



Wygląda lepiej, ale kroków prawie tyle samo

Funkcja Rosenbrocka





Funkcja Rosenbrocka $f(x,y) = (1-x)^2 + 100(y-x^2)^2$ jest przykładem funkcji, którą trudno zminimalizować. W klasycznym zadaniu minimalizuje się tę funkcję startując z punktu $(-3,-4)$. Wartość funkcji w tym punkcie wynosi 16916, zaś gradient to $[-15608,-2600]^T$, należy zatem poruszać się *w kierunku* minus gradientu, ale *o bardzo niewielki ułamek* jego długości. Prawy panel pokazuje końcowy przebieg (niezbyt udanej) minimalizacji funkcji Rosenbrocka za pomocą metody najszybszego spadku.

W pobliżu minimum

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

$$\frac{dg_k}{d\alpha} = \sum_i \frac{\partial f}{\partial x_i}(\mathbf{p}_k)_i = \left(\nabla f|_{\mathbf{x}=\mathbf{x}_{\min}}\right)^T \mathbf{p}_k = 0\,. \tag{5}$$

W minimum kierunkowym gradient funkcji jest prostopadły do kierunku poszukiwań. Zatem w metodzie najszybszego spadku kierunek poszukiwań (lokalny kierunek minimalizacji) co prawda zaczyna się prostopadle do poziomnic funkcji, ale kończy się *stycznie* do poziomnic. Natomiast 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

$$f(\mathbf{x}) \simeq \frac{1}{2}\mathbf{x}^T\mathbf{A}\mathbf{x} - \mathbf{b}^T\mathbf{x} + f_0\,, \tag{6}$$

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 (6) gradient funkcji f w punkcie $\mathbf{x}_k$ wynosi

$$\nabla f|_{\mathbf{x}=\mathbf{x}_k} = \mathbf{A}\mathbf{x}_k - \mathbf{b}\,. \tag{7}$$

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

$$\nabla f|_{\mathbf{x}} = \mathbf{A}\mathbf{x}_k + \alpha\mathbf{A}\mathbf{p}_{k+1} - \mathbf{b}\,. \tag{8}$$

Zmiana gradientu wynosi

$$\delta\left(\nabla f\right) = \alpha\mathbf{A}\mathbf{p}_{k+1}\,. \tag{9}$$

Punkt $\mathbf{x}_k$ jest minimum kierunkowym w kierunku $\mathbf{p}_k$, a więc gradient funkcji w tym punkcie spełnia warunek (5). *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*, $\delta \left(\nabla f\right)^T \mathbf{p}_k = 0$. Tak jednak musi być dla *wszystkich* poprzednich kierunków, nie chcemy bowiem naruszyć żadnego z poprzednich minimów kierunkowych. A zatem

$$\mathbf{p}_i^T \mathbf{A} \mathbf{p}_j = 0 \,, \quad i > j \,. \tag{10}$$

Metodę wybierania kierunków poszukiwań spełniających (10) nazywamy *metodą gradientów sprzężonych*.

Jak się obejść bez hessjanu?

Z jednego z poprzednich wykładów 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 1. *Niech f ma postać* (6) *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 (7), $\mathbf{r}_k = -\mathbf{A}\mathbf{x}_k + \mathbf{b}$ oraz

$$\mathbf{r}_{k+1} = -\mathbf{A}(\mathbf{x}_k + \alpha\mathbf{p}_k) + \mathbf{b} = \mathbf{r}_k - \alpha\mathbf{A}\mathbf{p}_k\,. \tag{11}$$

W minimum kierunkowym $\mathbf{p}_k^T \nabla f|_{\mathbf{x}_{k+1}} = -\mathbf{p}_k^T \mathbf{r}_{k+1} = 0$ (por. (5)). Wobec tego mnożąc równanie (11) lewostronnie przez $\mathbf{p}_k^T$, otrzymujemy

$$\alpha = \frac{\mathbf{p}_k^T \mathbf{r}_k}{\mathbf{p}_k^T \mathbf{A}\mathbf{p}_k}\,. \tag{12}$$

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)

$$\beta = \frac{\mathbf{r}_{k+1}^T \mathbf{r}_{k+1}}{\mathbf{r}_k^T \mathbf{r}_k} \,. \tag{13}$$

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

W metodzie gradientów sprzężonych te kroki, które *wymagałyby* znajomości hessjanu w minimum, zastępujemy minimalizacją kierunkową, natomiast te kroki, które *nie wymagają* znajomości hessjanu, wykonujemy tak samo, jak w algebraicznej metodzie gradientów sprzęzonych.

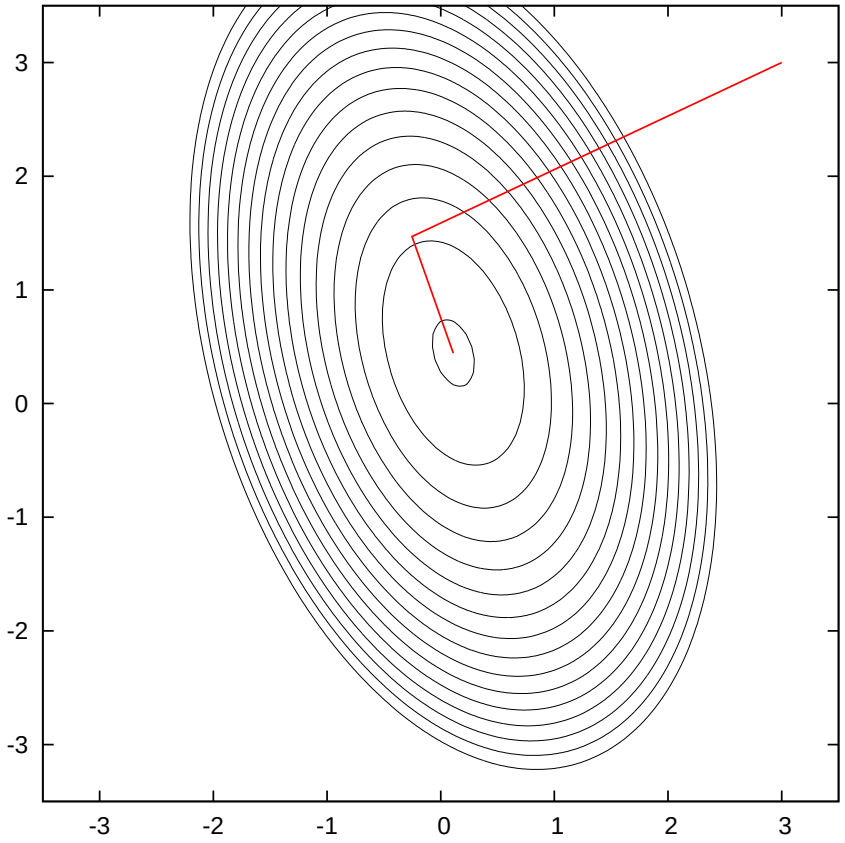
Twierdzenie ze strony 15 gwarantuje, że *formalnie* daje to to samo, co algebraiczna metoda gradientów sprzężonych.

Zamiast używać równania (13), można skorzystać z

$$\beta = \frac{\mathbf{r}_{k+1}^T(\mathbf{r}_{k+1} - \mathbf{r}_k)}{\mathbf{r}_k^T \mathbf{r}_k}\,. \tag{14}$$

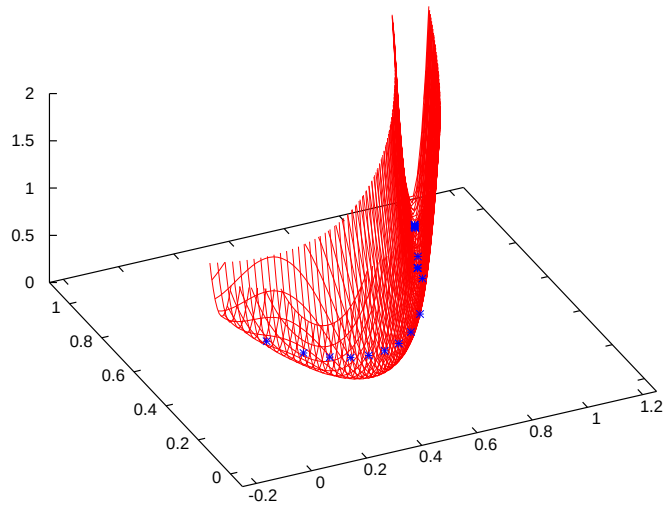
Jeżeli funkcja f ma *ściśle* postać (6), nie ma to znaczenia, gdyż $\mathbf{r}_{k+1}^T \mathbf{r}_k = 0$. Ponieważ jednak f jest tylko w przybliżeniu formą kwadratową, (14) może przyspieszyć obliczenia gdy grozi stagnacja.

Przykład — metoda gradientów sprzężonych



Tylko dwa kroki! Drugi krok nie jest prostopadły do pierwszego, ale jest z nim sprzężony.

Przykład — funkcja Rosenbrocka



Zastosowanie algorytmu gradientów sprzężonych do minimalizacji funkcji Rosenbrocka. Widać *znaczne* przyspieszenie (mniej punktów pośrednich!) w stosunku do przedstawianej powyżej metody najszybszego spadku.

Metoda zmiennej metryki

Formalnie minimum funkcji wielu zmiennych znajdujemy rozwiązując równanie $\nabla f = 0$. Sugeruje to zastosowanie następującego algorytmu opartego na metodzie Newtona:

$$\mathbf{x}_{i+1} = \mathbf{x}_i - \mathbf{H}_i^{-1} \nabla f|_{\mathbf{x}_i} \ , \tag{15}$$

gdzie $\mathbf{H}_i$ oznacza macierz drugich pochodnych cząstkowych funkcji f wyliczanych w punkcie $\mathbf{x}_i$ (aktualnym), natomiast $\nabla f|_{\mathbf{x}_i}$ jest gradientem funkcji w aktualnym punkcie.

Aby krok Newtona w istocie prowadził do zmniejszenia się wartości funkcji, musi zachodzić $(\mathbf{x}_{i+1} - \mathbf{x}_i)^T \nabla f|_{\mathbf{x}_i} < 0$, czyli

$$-(\mathbf{x}_{i+1} - \mathbf{x}_i)^T \mathbf{H}_i (\mathbf{x}_{i+1} - \mathbf{x}_i) < 0 \tag{16}$$

co jest spełnione, jeżeli $\mathbf{H}_i$ jest dodatnio określone.

W metodzie zmiennej metryki, zamiast używać hessjanu (macierzy drugich pochodnych cząstkowych), co może być numerycznie kosztowne, lub wręcz niemożliwe, a poza tym nie daje gwarancji, że hessjan *nie w bezpośrednim pobliżu minimum* będzie dodatnio określony, konstruujemy ciąg dodatnio określonych przybliżeń hessjanu, zbieżny do hessjanu w minimum.

Algorytm wygląda następująco: Startujemy z jakiegoś $\mathbf{x}_0$. Jako początkowe przybliżenie hessjanu $\mathbf{H}_0$ bierzemy jakąś sensowną macierz symetryczną, dodatnio określoną — jeśli nie mamy lepszego pomysłu, może to być macierz jednostkowa. Następnie wykonujemy iterację:

1. Rozwiązujemy równanie $\mathbf{H}_k \mathbf{p}_k = -\nabla f|_{\mathbf{x}_k}$.
2. Przeprowadzamy minimalizację kierunkową funkcji jednowymiarowej $f(\mathbf{x}_k + \alpha \mathbf{p}_k)$. Niech minimum kierunkowemu odpowiada wartość α_k.
3. $\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{p}_k$.
4. Wyliczamy $\mathbf{y}_k = \nabla f|_{\mathbf{x}_{k+1}} - \nabla f|_{\mathbf{x}_k}$.
5. Wyliczamy nowe przybliżenie $\mathbf{H}_{k+1}$, *na przykład* za pomocą formuły BFGS:

$$\mathbf{H}_{k+1} = \mathbf{H}_k + \frac{\mathbf{y}_k \mathbf{y}_k^T}{\alpha_k \mathbf{p}_k^T \mathbf{y}_k} - \frac{\nabla f|_{\mathbf{x}_k} \left(\nabla f|_{\mathbf{x}_k} \right)^T}{\mathbf{p}_k^T \mathbf{H}_k \mathbf{p}_k} \,. \tag{17}$$

Kończymy gdy $\|\mathbf{p}_k\| < \varepsilon$, gdzie ε jest zadaną tolerancją.

Własności metody zmiennej metryki

Można pokazać, że jeżeli $\mathbf{H}_0$ jest symetryczna i dodatnio określona, także następne $\mathbf{H}_k$ mają te własności.

Dodatkowo, jeśli f jest formą kwadratową (6), $\mathbf{H}_{i \geqslant N+1} \equiv \mathbf{H}_{\text{min}}$ w arytmetyce dokładnej. W ogólności ciąg $\mathbf{H}_k$ *nie musi* być zbieżny do "prawdziwego" hessjanu, nawet jeśli metoda daje zadowalające numerycznie przybliżenie minimum.

Metoda zmiennej metryki sprawdza się szczególnie w zagadnieniach o naprawdę dużej liczbie wymiarów.

W pewnym uproszczeniu można powiedzieć, że metoda zmiennej metryki ma się do metody gradientów sprzężonych tak, jak metoda Broydena ma się do wielowymiarowej metody Newtona (litera "B" w akronimie BFGS oznacza właśnie Broydena).

Metoda Levenberga-Marquardta

Tak metoda gradientów sprzężonych, jak i metoda zmiennej metryki, są dostosowane do przypadku, w którym funkcja jest z dobrym przybliżeniem postaci (6), 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 zmiennej metryki (lub gradientów sprzężonych).

Powróćmy do “algorytmu” opartego na metodzie Newtona (15):

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

Daleko od minimum hessjan nie musi być nawet dodatnio określony, co powoduje, iż krok newtonowski wcale nie musi prowadzić do spadku wartości funkcji (por. (16)). My jednak chcemy aby wartość funkcji w kolejnych krokach spadała. Zmodyfikujmy więc hessjan:

$$\widetilde{\mathbf{H}}_{ii} = (1+\lambda)\frac{\partial^2 f}{\partial x_i^2}\,, \tag{18a}$$

$$\widetilde{\mathbf{H}}_{ij} = \frac{\partial^2 f}{\partial x_i \partial x_j}\,, \quad i \neq j\,, \tag{18b}$$

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 (18) stanie się dodatnio określona lub też (ii) jeśli dla żadnego dodatniego λ macierz (18) 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

$$\mathbf{x}_{\text{test}} = \mathbf{x}_i - \widetilde{\mathbf{H}}^{-1}(\mathbf{x}_i)\nabla f(\mathbf{x}_i)\,. \tag{19}$$

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

Komentarz

Daleko od minimum *nie minimalizujemy* (nie poszukujemy minimów kierunkowych), a jedynie podążamy w kierunku malejących wartości funkcji.

Dodatkowo, jeśli $\lambda > \lambda_{\mathrm{max}} \gg 1$, uznajemy, iż znajdujemy się poza basenem atrakcji minimum i algorytm zawodzi. Jeśli natomiast $\lambda < \lambda_{\mathrm{min}} \ll 1$, macierz $\widetilde{\mathbf{H}}$ jest w praktyce równa hessjanowi, a zatem modyfikacja (18) 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 (6).

Ponadto w celu przyspieszenia obliczeń, jeżeli $f(\mathbf{x}_{\mathrm{test}}) < f(\mathbf{x}_i)$, 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.

W metodzie Levenberga-Marquardta używa się *prawdziwego* hessjanu, ale *nie* obliczanego w minimum, którego położenia nie znamy, ale w jakimś punkcie w otoczeniu minimum. Spełnienie warunku $\lambda < \lambda_{\mathrm{min}} \ll 1$ oznacza, że hessjan jest dodatnio określony, a więc krok Newtona faktycznie prowadzi do zmniejszenia wartości funkcji, skąd wnosimy, że znaleźliśmy sie w basenie atrakcji (lokalnego) minimum. W tym momencie możemy się przerzucić na metodę gradientów sprzężonych lub na metodę zmiennej metryki. Jeśli wybierzemy ten drugi wariant, jako przybliżenie początkowe możemy wziąć *już obliczony* hessjan *z otoczenia minimum* — będzie to lepsze przybliżenie, niż macierz jednostkowa.

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

$$\begin{aligned} \mathbf{x}_{\text{test}} &\simeq \mathbf{x}_i - (1+\lambda)^{-1} \text{diag} \left\{ \left(\frac{\partial^2 f}{\partial x_1^2} \right)^{-1}, \left(\frac{\partial^2 f}{\partial x_2^2} \right)^{-1}, \ldots, \left(\frac{\partial^2 f}{\partial x_N^2} \right)^{-1} \right\} \nabla f(\mathbf{x}_i) \\ &\simeq \mathbf{x}_i - \text{const} \, \nabla f(\mathbf{x}_i) \, , \end{aligned} \tag{20}$$

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.

Metoda Powella

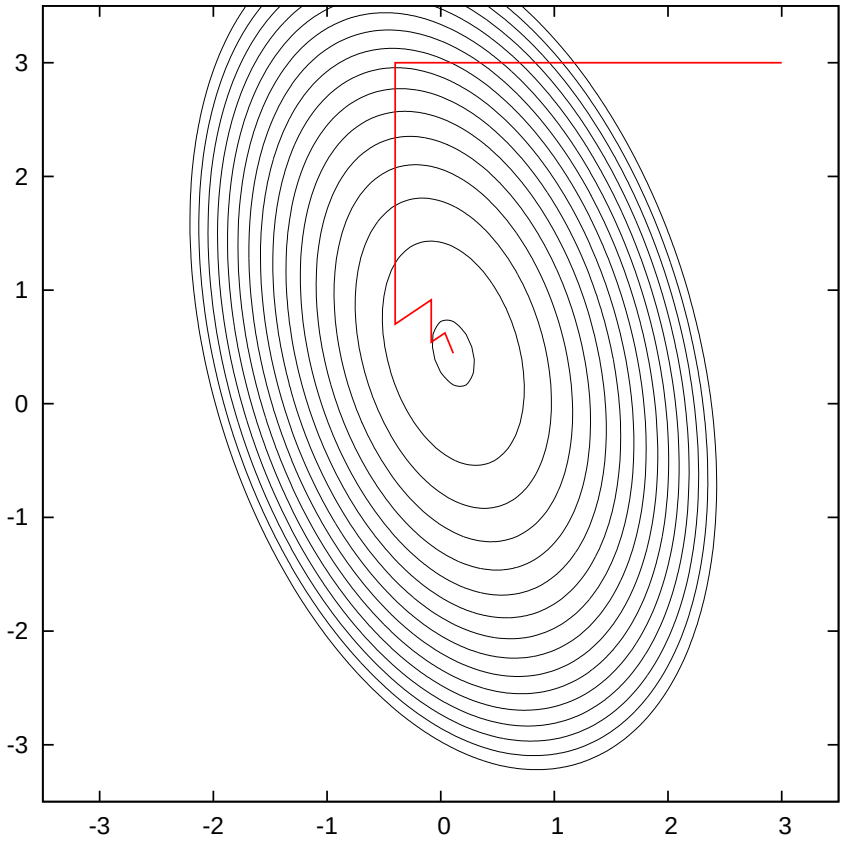
Wszystkie omówione wyżej metody wymagały obliczania pochodnych (gradientu) funkcji $f(\mathbf{x})$. Co jednak zrobić, jeśli obliczenie pochodnych jest kosztowne, niemożliwe lub funkcja jest nieróżniczkowalna? Zasadnicza strategia postępowania — minimalizacja kierunkowa, wybór nowego kierunku etc — pozostaje w mocy, zmienia się tylko sposób wyboru kolejnych kierunków. Metoda Powella polega na konstrukcji kierunków, które z czasem, po wielu iteracjach, stają się sprzężone.

Inicjalizujemy biorąc $\{\mathbf{p}_i\}_{i=1}^N = \{\mathbf{e}_i\}_{i=1}^N$, gdzie $\{\mathbf{e}_i\}_{i=1}^N$ są kolejnymi wersorami (innymi słowy, zaczynamy od minimalizacji po współrzędnych). Następnie

1. Znajdujemy się w pewnym punkcie $\mathbf{X}_0$.
2. Minimalizujemy wzdłuż kolejnych kierunków $\mathbf{p}_i$, osiągając kolejno punkty $\mathbf{X}_i$.
3. Dla $i = 1, \ldots, N-1$: $\mathbf{p}_i = \mathbf{p}_{i+1}$.
4. $\mathbf{p}_N = \mathbf{X}_N - \mathbf{X}_0$.
5. Minimalizujemy wzdłuż (nowego) $\mathbf{p}_N$, oznaczając znaleziony punkt przez $\mathbf{X}_0$.
6. `GOTO 1`

Jeśli badana funkcja jest rozwijalna w szereg Taylora wokół minimum, po N iteracjach powyższej procedury kierunki $\{\mathbf{p}_i\}_{i=1}^N$ stają się sprzężone.

Przykład — metoda Powella



Mniej kroków niż w minimalizacji po współrzędnych. W większej liczbie wymiarów byłoby *jeszcze lepiej*.

Z punktu widzenia Big Data...

W ciągu ostatnich kilkunastu lat rozmiary dostępnych zbiorów danych rosną *szybciej* niż rośnie prędkość procesorów (nawet po uwzględnieniu paralelizacji i obliczeń na GPU). Z tego punktu widzenia możliwości uczenia maszynowego są ograniczone raczej przez możliwości obliczeniowe niż przez dostępne zbiory danych. Tę klasę problemów zwyczajowo określa się jako *Big Data*.

Uczenie maszynowe bardzo często oznacza konieczność minimalizowania funkcji

$$Q(\mathbf{X}) = \frac{1}{N} \sum_{i=1}^{N} Q_i(\mathbf{X}; x_i, y_i) \tag{21}$$

gdzie

- $\mathbf{X}$ jest wielowymiarową zmienną, po której minimalizujemy — wektorem estymatorów, których wartości chcemy znaleźć. Tego typu problemy pojawiają się w zagadnieniu najmniejszych kwadratów.
- $\forall i: Q_i \geqslant 0$,
- Q_i mają taką samą postać funkcyjną, różnią się tylko elementami x_i, y_i (elementami zbioru uczącego),
- $N \gg 1$ (N jest zazwyczaj **BARDZO** duże, rzędu kilkudziesięciu, niekiedy kilkuset tysięcy).

Problem (21) można rozwiązać za pomocą którejś z już omówionych metod (minimalizacja funkcji wielu zmiennych lub rozwiązywanie za pomocą SVD nadokreślonych układów równań, jeżeli mowa o liniowym zagadnieniu najmniejszych kwadratów). Jednak dla bardzo dużych N, czyli dla bardzo dużych zbiorów uczących, inne podejście może być bardziej efektywne.

Punktem wyjścia jest metoda najszybszego spadku (ang. *steepest gradient descent*): W każdym kroku podążamy w kierunku minus gradientu funkcji $Q(\mathbf{p})$, aż do osiągnięcia minimum kierunkowego.

Jak jednak pamiętamy, daleko od minimum nie minimalizujemy, tylko podążamy z arbitralnie dobranym krokiem w kierunku ujemnego gradientu.

Dlatego zamiast minimalizować, w n-tym kroku iteracji przyjmujemy

$$\begin{aligned} \mathbf{X}_{n+1} &= \mathbf{X}_n - \gamma \cdot \nabla_{\mathbf{X}} Q(\mathbf{X})|_{\mathbf{X}_n} \\ &= \mathbf{X}_n - \gamma \cdot \frac{1}{N} \sum_{i=1}^{N} \nabla_{\mathbf{X}} Q_i(\mathbf{X}; x_i, y_i)|_{\mathbf{X}_n} \ . \end{aligned} \tag{22}$$

$\gamma =$ const i jest nazywane *prędkością uczenia* (ang. *learning rate*).

Uwaga: Przyjęcie stałego kroku nie jest aż tak naiwne, jak by się to mogło wydawać. Jeśli rozpatrzymy układ równań różniczkowych typu

$$\frac{d\mathbf{y}}{dx} = \mathbf{f}(\mathbf{y}) \tag{23}$$

to najprostszą (zdecydowanie nie najlepszą, ale najprostszą i najszybszą) metodą jego numerycznego rozwiązywania jest *metoda Eulera*

$$\mathbf{y}_{n+1} = \mathbf{y}_n + h \cdot \mathbf{f}(\mathbf{y}_n)\,, \tag{24}$$

gdzie h jest stałym krokiem narastania zmiennej niezależnej x. Metoda najszybszego spadku (22) ze stałym krokiem γ ma taką samą postać, co metoda Eulera, po utożsamieniu $f \equiv -\nabla Q$.

Stochastic Gradient Descent

Jeśli w (22) $N \gg 1$, czas obliczania sumy gradientów cząstkowych $\sum_i^N \nabla Q_i$ może być bardzo znaczny. Zarazem wyrażenie $\frac{1}{N}\sum_i^N \nabla Q_i$ ma postać średniego gradientu cząstkowego. Jeśli założymy, że poszczególne elementy tej sumy nie odbiegają zbytnio od średniej, możemy średni, kosztowny w wyliczaniu gradient, zastąpić *losowo wybranym* gradientem cząstkowym. Otrzymujemy zatem algorytm *Stochastic Gradient Descent*:

- wybierz losowo $k \in \{1, 2, \ldots, N\}$
- przyjmij

$$\mathbf{X}_{n+1} = \mathbf{X}_n - \gamma \, \nabla_{\mathbf{X}} Q_k(\mathbf{X}; x_k, y_k)|_{\mathbf{X}_n} \tag{25}$$

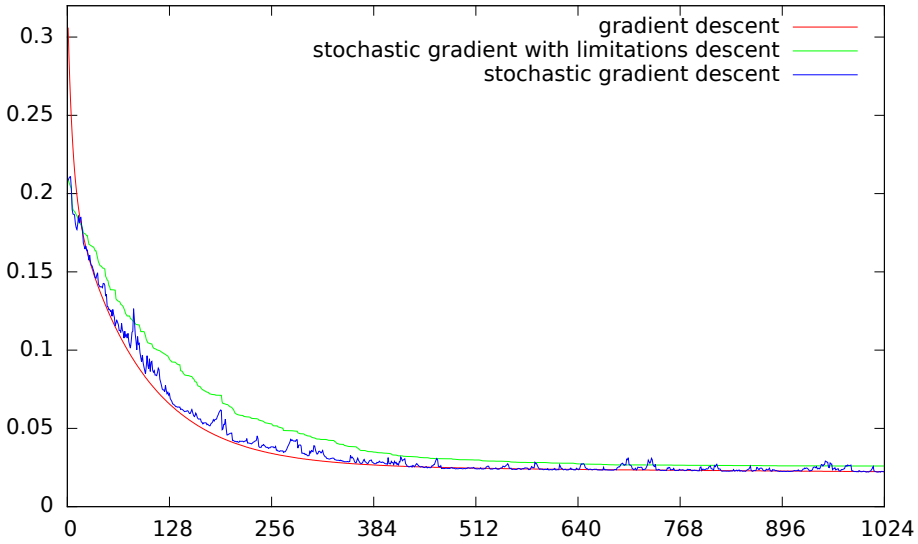
- Dodatkowo akceptujemy powyższy krok tylko jeżeli $Q(\mathbf{X}_{n+1}) < Q(\mathbf{X}_n)$. Jeśli nie, nie wykonujemy kroku, ale losujemy nowe k. Ten wariant nazwiemy *Stochastic Gradient Descent z ograniczeniem*.

Iterację kończymy albo po wykonaniu z góry określonej liczby kroków, albo — częściej — gdy różnica $\left|Q(\mathbf{X}_n) - Q(\mathbf{X}_{n+1})\right| < \varepsilon$, gdzie ε jest ustaloną tolerancją.

Wartość *prędkości uczenia* γ oraz tolerancję ε dobieramy "eksperymentalnie", to znaczy kierując się znajomością natury problemu i doświadczeniem uzyskanym przy rozwiązywaniu podobnych problemów.

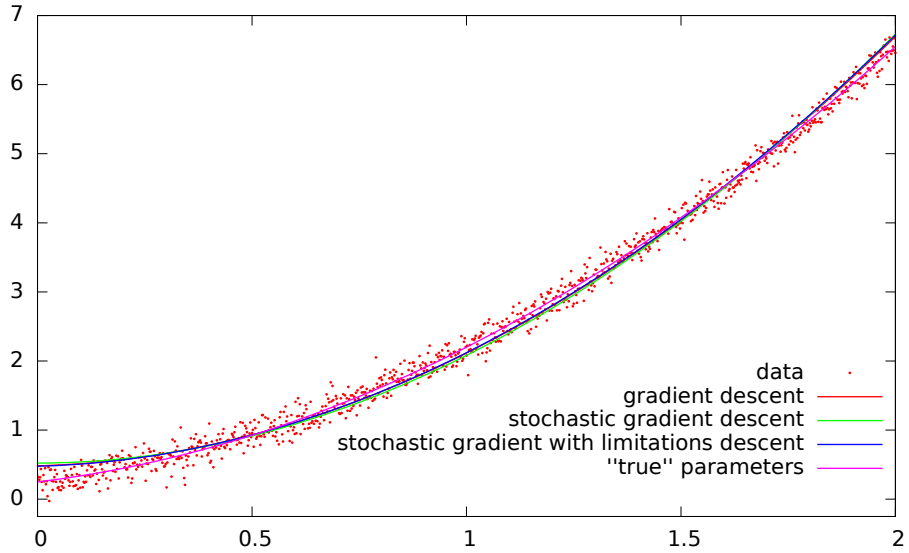
Okazuje się, że metoda *Stochastic Gradient Descent* działa zdumiewająco dobrze dla bardzo dużych zbiorów uczących. Co ciekawe, przyjęcie warunku, iż wartość funkcji Q musi maleć po wykonaniu kroku na ogół *pogarsza* wyniki: Od czasu do czasu można/trzeba wykonać krok "w złą stronę".

Przykład



Wygenerowano $N = 1024$ punktów $y_i = ax_i^2 + bx_i + c + \xi_i$, gdzie $\{\xi\}_{i=1}^N$ są liczbami o rozkładzie normalnym, natomiast $x_i \in [0, 2]$. Rysunek przedstawia kolejne wartości funkcji $Q(a, b, c)$ uzyskane w metodzie najszybszego spadku (czerwony), w metodzie *Stochastic Gradient Descent* (niebieski) oraz *Stochastic Gradient Descent* z ograniczeniami (zielony). $\gamma = 1/128$. W przypadku *Stochastic Gradient Descent* wartość minimalizowanej funkcji niekiedy rośnie, ale po dostatecznie dużej liczbie iteracji wyniki

tej metody są nieco *lepsze*, niż dla metody z ograniczeniem. Co ważniejsze, wyniki są porównywalne z wynikami uzyskanymi w obliczeniowo droższej metodzie najszybszego spadku.



Dopasowane krzywe są niemalże nieodróżnialne w obszarze odpowiadającym danym wejściowym (zawartości zbioru uczącego).