Wojciech Myszka

Laboratorium 5 wer. 38 z drobnymi modyfikacjami!

2017-12-06 08:50:59 +0100

1. Laboratorium 5: Optymalizacja

1.1. Wstęp

Zadania optymalizacyjne to podstawa całej teorii sterowania. Zazwyczaj nie tylko nam chodzi, żeby osiągnąć zadany cel, ale również, aby osiągnąć go albo w najkrótszym czasie, albo z wykorzystaniem najmniejszej liczby zasobów.

Zadanie optymalizacji pojawić się może tylko wtedy, gdy możliwe jest więcej niż jedno rozwiązanie.

1.2. Minimalizacja funkcji jednej zmiennej

Problem jest – w zasadzie – bardzo prosty: należy znaleźć takie x, dla którego $f(x)$ przyjmuje wartość minimalną.

Mówimy też, że $\tilde{x}$ jest minimum lokalnym jeżeli

$$f(\tilde{x}) < f(x), \qquad ||x - \tilde{x}|| \leq \varepsilon$$

Funkcja może mieć wiele minimów lokalnych. Jeżeli dla $\tilde{x} \in X$ zachodzi:

$$f(\tilde{x}) = \inf_{x \in X} f(x)$$

wówczas mówimy, że $\tilde{x}$ jest minimum globalnym w zbiorze X.

Zadanie w najogólniejszym przypadku jest bardzo trudne! Jeżeli przyjąć dodatkowe założenia – że funkcja $f(x)$ jest ciągła (co najmniej odcinkami ciągła), a zbiór X jest zwarty i domknięty[1] (co to znaczy?) to można kusić się o tworzenie jakichkolwiek sensownych algorytmów szukania minimum. (Jeżeli funkcja $f(x)$ nie jest co najmniej odcinkami ciągła – może okazać się, że jedyną metodą poszukiwania minimum jest przegląd wartości funkcji dla wszystkich dopuszczalnych argumentów.)

Jeżeli funkcja jest różniczkowalna szukanie minimum może sprowadzić się do rozwiązania równania (układu równań):

$$f'(x) = 0$$

[1] Zbiór jest **domknięty**, gdy granica każdego ciągu, którego elementy należą do zbioru również należy do zbioru. Gdy dodatkowo zbiór jest ograniczony — jest **zawarty**.

i przeanalizowania wszystkich „podejrzanych punktów”, w których zeruje się pierwsza pochodna funkcji.

Ale, szczerze mówiąc problem sprowadza się do zastąpienia jednego trudnego zadania innym trudnym zadaniem.

Jedyna ogólna metoda aby znaleźć minimum globalne to znaleźć wszystkie minima lokalne i wybrać z nich najmniejsze. Zatem, tak na prawdę, potrzebujemy dobrych algorytmów wyszukiwania minimów lokalnych.

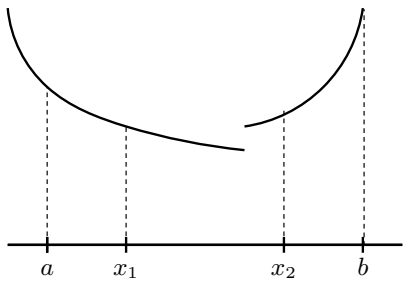
Na ogół X jest pewnym podzbiorem większej przestrzeni (mówiąc bardzo nieprecyzyjnie). Zadanie wyszukiwania minimum w zbiorze X jest wówczas zadaniem poszukiwania minimum z ograniczeniami. W pewnych przypadkach, gdy ograniczenia zadane są za pomocą równości stosując metodę mnożników Lagrange'a można zadanie z ograniczeniami sprowadzić do zadania poszukiwania minimum bez ograniczeń.

Gdy nie da się problemu rozwiązać analitycznie (na przykład rozwiązując układ równań), pozostaje stosowanie metod iteracyjnych. Jedną z nich opiszę poniżej.

1.2.1. Metoda złotego podziału

Zakładamy, że odcinkami ciągła funkcja funkcja $f(x)$ określona dla $a \leq x \leq b$ ma tylko jedno minimum lokalne. Chcemy stworzyć algorytm, który będzie generował ciąg liczb zbieżny do tego minimum.

Wyliczamy wartości funkcji na końcach przedziału oraz w dwu punktach wewnętrznych x_1 i x_2, porównamy wszystkie (4) wartości funkcji i wybierzemy najmniejszą z nich. Dla skupienia uwagi niech będzie to punkt x_1. Mamy trzy przedziały $[a, x_1]$, $[x_1, x_2]$ oraz $[x_2, b]$. Jasne jest, że możemy odrzucić przedział $[x_2, b]$. (Czemu!?)



Dalsza procedura polega na tym, żeby na odcinku $[a, x_2]$ wybrać kolejne dwa wewnętrzne punkty, wyznaczyć w nich wartości funkcji i procedurę powtórzyć...

Warto jednak uwzględnić fakt, że mamy już wyliczone wartości funkcji w trzech punktach: a, x_1 i x_2 – warto więc tak wybrać czwarty punkt aby wykorzystać wszystko to co już mamy.

Kolejny punkt tak dodajemy, aby kolejny (krótszy) odcinek podzielony był w sposób „podobny” do wyjściowego. Powinny być spełnione następujące zależności:

$$b - x_2 = x_1 - a, \qquad \frac{x_1 - a}{b - a} = \frac{x_2 - x_1}{x_2 - a}$$

Rozwiązanie jest następujące:

$$\frac{b - x_2}{b - a} = \frac{x_1 - a}{b - a} = \xi, \qquad \xi = \frac{2}{3 + \sqrt{5}} \approx 0{,}38$$

W każdej iteracji długość odcinka skraca się $1 - \xi \approx 0{,}62$ raza; zatem po n wyliczeniach wartości funkcji, długość wynikowego odcinek wynosi $(1 - \xi)^{n-3}$ początkowej długości. Gdy $n \to \infty$ długość odcinka zmierza do zera co gwarantuje zbieżność metody...

Dla ujednolicenia przyjmiemy $a = x_0$, $b = x_1$; na pierwszym kroku wyliczamy:

$$x_2 = x_0 + \xi(x_1 - x_0), \qquad x_3 = x_1 - \xi(x_1 - x_0).$$

Po wyliczeniu wartości funkcji i wybraniu minimum – odrzucony może być punkt z dowolnym indeksem... Może to sprawiać problem.

Jeżeli na pewnym etapie mamy cztery punkty x_i, x_j, x_k, x_l (któreś z nich są końcami przedziału). Załóżmy, że w punkcie x_i funkcja f osiąga minimalną wartość. Odrzucamy ten punkt, który jest najbardziej oddalony od x_i (Czemu takie działanie jest uprawnione? I w jakich warunkach?). Niech będzie to punkt x_l. Trzy pozostałe punkty szeregujemy w kolejności:

$$x_k < x_i < x_j.$$

Kolejny (wewnętrzny punkt) wyznaczamy ze wzoru:

$$x = x_j + x_k - x_i$$

(Czy na pewno spełnia on wymagania złotego podziału?)

Obliczenia kończymy gdy

$$|x_j - x_k| < \delta$$

1.2.2. Naiwna metoda Monte-Carlo

Losowo wybieramy punkty z zadanego obszaru. Po kilku iteracjach (ilu?) porównujemy wartości funkcji i wybieramy najmniejszą.

1.2.3. Minimum funkcji wielu zmiennych

Najprostsza metoda to „cięcie” funkcji wielu zmiennych „po osiach”. W każdym kierunku dokonujemy minimalizacji funkcji jednej zmiennej.

1.2.4. Zadania

1. Zaimplementować (matlab, python, C, C++, Mathematica) metodę złotego podziału i przetestować na funkcji $y = x^2$ w przedziale $[-2, 2]$, funkcji $y = \sin(x)$ w przedziale $[0, 3\pi]$. Zliczać liczbę obliczeń wartości funkcji do osiągnięcia zadanej dokładności. Porównać osiągnięty wynik z Naiwną Metodą Monte-Carlo (dla tej samej liczby obliczeń wartości funkcji)
2. Powtórzyć powyższe w przypadku funkcji dwu zmiennych: $F(x, y) = 10(y - \sin(x))^2 + 0{,}1x^2$

1.3. Programowanie liniowe

Programowanie liniowe to szczególny przypadek poszukiwania minimum. Funkcja celu jest liniowa, na przykład postaci:

$$f'x = \sum_{i=1}^{N} f_i x_i$$

gdzie f to wektor współczynników funkcji, a x wektor zmiennych niezależnych.

Ponieważ natura funkcji liniowej jest taka, że jest to funkcja monotoniczna (albo maleje, albo rośnie[2] wraz ze wzrostem wartości składowej x_i) — może przyjąć dowolnie wielkie lub dowolnie małe wartości. Minimalizacja/maksymalizacja nie mają sensu.

Zadanie zaczyna mieć sens, gdy wprowadzimy ograniczenia na zmienne. Najprostszy przypadek to ograniczenia w postaci zestawu nierówności:

$$Ax \leq b$$

(A to jakaś macierz) ale nierówności mogą mieć znacznie bardziej skomplikowaną postać:

$$A_{\text{eq}} x = b_{\text{eq}}$$

(A_{eq} to pewna macierz, a b_{eq} — wektor).

W matlabie problem rozwiązuje funkcja linprog. Zadanie może mieć wiele różnych postaci, najprostsze to $Ax \leq b$. Aby problem rozwiązać:

```
x = linprog(f,A,b)
```

Zadanie z ograniczeniami nierównościowymi i równościowymi:

```
x = linprog(f,A,b,Aeq,beq)
```

[2] Pomijam przypadek, gdy wartość jest stała.

`Aeq` to macierz A_{eq}, a `beq` to wektor $b_t exteq$.

Znaleźć mamy maksimum funkcji cx przy ograniczeniach $Ax = b$ i $x \geq 0$. A jest macierzą $m \times n$, $\text{rank}(A) = m$, i $b \geq 0$ (b, x, c to wektory odpowiednich rozmiarów).

W Mathematici, można użyć jednej z funkcji: **LinearProgramming**, **FindMinimum**, **FindMaximum**, **NMinimize**, **NMaximize**, **Minimize** oraz **Maximize**. Pełnię możliwości ma LinearProgramming. Opis w dokumentacji.

1.3.1. Ćwiczenia

Właściciel ciężarówki może przewieźć z miejscowości A do B cukier, mąkę i chipsy. W ciężarówce mieści się towar o objętości co najwyżej 7000 litrów i wadze co najwyżej 5 ton. 1 kg cukru zajmuje objętość 1,5 litra, 1 kg mąki 2 litry, natomiast 1 kg chipsów zajmuje objętość 4 litrów. Zysk od przewozu poszczególnych towarów jest następujący:
— za 100 kg cukru 8 zł,
— za 100 kg mąki 10 zł,
— za 100 kg chipsów 25 zł.

Ile cukru, mąki i chipsów powinien załadować właściciel ciężarówki aby zmaksymalizować swój zysk? Matematyczny model tak postawionego zadania jest następujący: Oznaczmy przez:
— x_1 – ilość cukru,
— x_2 – ilość mąki,
— x_3 – ilość chipsów
(za każdym razem w setkach kilogramów). Skoro ciężarówka może zabrać co najwyżej 5 ton towarów, musi zachodzić nierówność:

$$100x_1 + 100x_2 + 100x_3 \leq 5000$$

Z kolei ograniczenie objętości wyraża się wzorem

$$150x_1 + 200x_2 + 400x_3 \leq 7000$$

Zysk właściciela wynosi:

$$z \ = 8x_1 + 10x_2 + +25x_3$$

x_1, x_2 i x_3 muszą być oczywiście nieujemne. Po uproszczeniach otrzymamy więc problem programowania matematycznego:

$f(x_1, x_2, x_3) = 8x_1 + 10x_2 + 25x_3 \to \max$

w zbiorze $\begin{cases} x_1 & +x_2 & +x_3 & \leq 50 \\ 1,5x_1 & +2x_2 & +4x_3 & \leq 70 \\ & x_j \geq 0 & \text{dla} & j = 1, 2, 3. \end{cases}$

Kolejne ćwiczenie będzie równie proste.

$$
\begin{array}{rcrrrl}
 & & x_1 & +2x_2 & +3x_3 & \leq 5 \\
 & & 2x_1 & +3x_2 & +5x_3 & \leq 8 \\
 & & 3x_1 & +x_2 & +3x_3 & \leq 4 \\
 & & & & & x_i \geq 0 \quad (i = 1, ..., 3) \\
\hline
z & = & 2x_1 & +x_2 & +3x_3 & \rightarrow \quad \max
\end{array}
\tag{1.1}
$$

Trzeba je rozwiązać za pomocą programu **linprog** a następnie „przedyskutować” uzyskane rozwiązanie (to znaczy sprawdzić czy spełnione są wszystkie ograniczenia, i zastanowić się, czy rozwiązanie jest optymalne[3]...) Można również użyć Mathematici.

1.4. Instrukcja w postaci jednego pliku...

...jest również dostępna.

[3] Jak to wykazać?