МЕТОД ЭЛЛИПСОИДОВ И OCTAVE-ПРОГРАММА EMSHOR

Стецюк П.И., Ивличев А.В.
stetsyukp@gmail.com

Институт кибернетики им. В.М. Глушкова, Киев

IX Міжнародна школа-семінар "Теорія прийняття рішень"
15-20 квітня 2019 року, Ужгород, Україна

Содержание

Содержание

Метод эллипсоидов предложили

1976 **Юдин Д.Б. и Немировский А.С.** как метод последовательных отсечений [1].

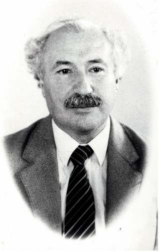
1977 **Шор Н.З.** как вариант метода с растяжением пространства в направлении субградиента [2].

1. Юдин Д.Б., Немировский А.С. *Информационная сложность и эффективные методы решения выпуклых экстремальных задач* // Экономика и математические методы. – 1976. – Вып. 2. – С. 357–369.

2. Шор Н.З. *Метод отсечения с растяжением пространства для решения задач выпуклого программирования* // Кибернетика. – 1977. – № 1. – С. 94–95.

Метод эллипсоидов „с берегов Днепра“ ...

Давид Борисович Юдин
родился 21 мая 1919 года
в Екатеринославе (сегодня - Днепр),
в 1941 году закончил
Днепропетровский университет

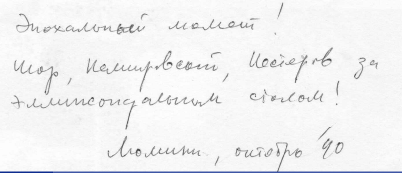


Наум Зуселевич Шор
родился 1 января 1937 года
в Киеве (город на Днепре),
в 1958 году закончил
Киевский университет

Эпохальный момент!

Шор,
Немировский,
Нестеров за
эллипсоидальным
столом!

октябрь 1990



Эпохальный момент!
Шор, Немировский, Нестеров за
эллипсоидальным столом!

Люмини, октябрь '90

XI симпозиум по мат. программированию

Метод эллипсоидов и полученные на его основе результаты о сложности задач математического программирования были центральными на XI международном симпозиуме по математическому программированию (Бонн, ФРГ, август 1982).

3. Канторович Л.В., Михалевич В.С., Рубинштейн Г.Ш., Третьяков Н.В., Шор Н.З., Якимец В.Н. *XI Международный симпозиум по математическому программированию* // Техническая кибернетика. – М.: Изв. АН СССР. – 1983. – № 1. – С. 197–201.

XI симпозиум (метод эллипсоидов)

Три премии – им. Фалкерсона (1), им. Данцига (2):

им. Фалкерсона: Grötchel M., Lóvasz L., Schrijver A., **1981**
им. Данцига: Хачиян Л., **1979**
им. Данцига: Юдин Д., Немировский А., **1976**

Пленарный доклад Шора:

"Generalized gradient methods of nondifferentiable optimization employing space dilatation operations", опубликован в [4]

4. MATHEMATICAL PROGRAMMING: THE STATE OF ART, BONN, 1982 / *Bachem A., Grötchel M., Korte B. (eds.)* – Berlin: Springer-Verlag, 1983. – 655 p.

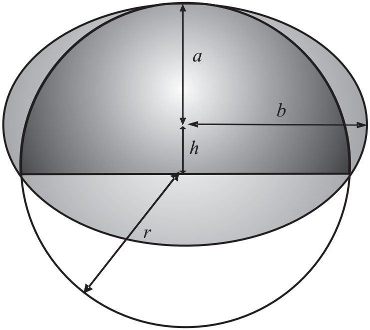
Шор (1982) и Юдин (1983)

Шор в Бонне (1982)

Юдин в Риге (1983)

Содержание

1-d эллипсоид и его свойства



Эллипсоид $\mathcal{E}_n$, содержащий полушар в E^n, имеет параметры

$$b = \left(\alpha + \frac{1}{\alpha}\right)\frac{r}{2}, \quad h = \left(1 - \frac{1}{\alpha^2}\right)\frac{r}{2},$$

где $\alpha = \frac{b}{a}$ и r – радиус шара S_n.

Если пространство „растянуть“ с коэффициентом α в направлении полуоси a, то $\mathcal{E}_n$ станет шаром в преобразованном пространстве.

Отношение объема эллипсоида $\mathcal{E}_n$ к объему шара S_n равно

$$q(n) = \frac{\mathrm{vol}(\mathcal{E}_n)}{\mathrm{vol}(S_n)} = \frac{1}{\alpha}\left(\frac{b}{r}\right)^n = \frac{1}{\alpha}\left(\frac{1}{2}\left(\alpha + \frac{1}{\alpha}\right)\right)^n.$$

Оператор растяжения пространства

Введен Н.З. Шором (1969) и имеет следующий вид

$$R_\alpha(\xi) = I_n + (\alpha - 1)\xi\xi^T, \quad \text{где} \quad \alpha > 1.$$

Здесь: α – коэффициент растяжения пространства
в нормированном направлении $\xi \in E^n$, $\|\xi\| = 1$,
I_n – единичная матрица размером $n \times n$.

В методах используется обратный к нему оператор

$$R_\beta(\xi) = I_n + (\beta - 1)\xi\xi^T, \quad \text{где} \quad \beta = \frac{1}{\alpha} < 1,$$

который означает "сжатие"пространства субградиентов.

Почему метод эллипсоидов сходится?

Отношение объема эллипсоида $\mathcal{E}_n$ к объему шара S_n равно

$$q(n) = \frac{\mathrm{vol}(\mathcal{E}_n)}{\mathrm{vol}(S_n)} = \frac{1}{\alpha}\left(\frac{1}{2}\left(\alpha + \frac{1}{\alpha}\right)\right)^n.$$

Если коэффициент α такой, что $\alpha + 1/\alpha < 2\sqrt[n]{\alpha}$, то отношение $q(n) < 1$ и объем эллипсоида, в котором локализуется искомая точка x^*, убывает со скоростью геометрической прогрессии со знаменателем $q(n)$.

В методе эллипсоидов Юдина-Немировского-Шора

$$q(n) \leq 1 - \frac{1}{2n} \quad \text{и реализуется при} \quad \alpha = \sqrt{\frac{n+1}{n-1}}.$$

Содержание

Задача и критерий останова

Решается такая задача:

для выпуклой функции $f(x)$, $x \in R^n$ найти точку x_ε^*, для которой $f(x_\varepsilon^*) - f^* \leq \varepsilon_f$, где $f^* = f(x^*)$.

Входной параметр метода:

$\varepsilon_f > 0$ – точность, с которой требуется найти $f_\varepsilon^* = f(x_\varepsilon^*)$.

Обозначения:

$g(x_k)$ – субградиент функции $f(x)$ в точке x_k.

Ицициализация в методе

Инициализация:

- Положим стартовую точку $x_0 \in R^n$ и радиус r_0 такой, что $\|x_0 - x^*\| \le r_0$.
- Введем в рассмотрение $n \times n$-матрицу B и положим $B_0 := I_n$, где I_n – единичная $n \times n$-матрица. Перейдем к первой итерации со значениями x_0, r_0 и B_0.

Пусть на k-й итерации найдены значения $x_k \in R^n$, r_k, B_k. Переход к $(k+1)$-й итерации состоит в выполнении такой последовательности действий.

Итерация метода

Шаг 1. Вычислим $f(x_k)$ и $g(x_k)$. Если $\|B_k^T g(x_k)\| r_k \le \varepsilon_f$, то "Останов: $k^* = k$ и $x_\varepsilon^* = x_k$". Иначе переходим к шагу 2.

Шаг 2. Вычислим очередную точку

$$x_{k+1} := x_k - h_k B_k \xi_k, \text{ где } \xi_k := \frac{B_k^T g(x_k)}{\left\| B_k^T g(x_k) \right\|}, \; h_k = \frac{1}{n+1} r_k.$$

Шаг 3. Вычислим

$$B_{k+1} := B_k + \left(\sqrt{\frac{n-1}{n+1}} - 1 \right) (B_k \xi_k)\, \xi_k^T \quad \text{и} \quad r_{k+1} := r_k \frac{n}{\sqrt{n^2-1}}.$$

Шаг 4. Переходим к итерации $(k+1)$ со значениями $x_{k+1}, r_{k+1}, B_{k+1}$.

Сходимость метода

Теорема.

Пусть $A_k = B_k^{-1}$. Последовательность точек $\{x_k\}_{k=0}^{k^*}$ удовлетворяет неравенству

$$\|A_k(x_k - x^*)\| \leq r_k, \quad k = 0, 1, 2, \dots, k^*$$

На каждой итерации $k>0$ уменьшение объема эллипсоида $\mathcal{E}_k = \{x \in R^n : \|A_k(x_k - x)\| \leq r_k\}$, локализующего точку x^*, есть величина постоянная и равна

$$q = \frac{vol(\mathcal{E}_k)}{vol(\mathcal{E}_{k-1})} = \sqrt{\frac{n-1}{n+1}} \left(\frac{n}{\sqrt{n^2-1}} \right)^n < \exp\left\{ -\frac{1}{2n} \right\} < 1.$$

Что следует из теоремы?

Для уменьшения в 10 раз объема эллипсоида, локализующего x^*, требуется K итераций, где

$$K = -\frac{\ln 10}{\ln q} \approx (2\ln 10)n \approx 4.6n,$$

т.е. чтобы на порядок улучшить отклонение найденного рекордного значения функции $f(x)$ от ее оптимального значения f^*, потребуется сделать $4.6n^2$ итераций.

Содержание

Octave-функция emshor (ellipsoid method shor)

```
# octave-code for emshor (version 1.1, 14.03.2019)        #row00
# шукає наближення до точки мінімуму опуклої функції f(x)#row00a
# Вхідні параметри:                                       #row00b
#   calcfg - ім'я функції для обчислення f та g           #row00c
#   x0 - стартова точка, x0(1:n)                          #row00d
#   rad - радіус кулі, що локалізує точку мінімуму        #row00e
#   epsf, maxitn - параметри зупинки (точн., макс. ітер.)#row00f
#   intp - інтервал друку (через кожні intp ітерацій)     #row00g
# Вихідні параметри:                                      #row00h
#   x - знайдене наближення до точки мінімуму, x(1:n)     #row00i
#   f - значення функції f в точці x                      #row00j
#   itn - кількість виконаних ітерацій                    #row00k
#   ist - код завершення (1 = epsf, 4 = maxitn)           #row00l

function [x,f,itn,ist] = emshor(calcfg,x0,rad,            #row01
                                epsf,maxitn,intp);
dn=double(length(x0)); beta=sqrt((dn-1.d0)/(dn+1.d0));   #row02
x=x0; radn=rad; B=eye(length(x));                         #row03
for (itn = 0:maxitn)                                      #row04
   [f, g1] = calcfg(x); g=B'*g1; dg=norm(g);              #row05
   if(radn*dg < epsf) ist = 1;  return; endif             #row06
   xi=(1.d0/dg)*g; dx = B * xi;                           #row07
   hs=radn/(dn+1.d0); x -= hs * dx;                       #row08
   B += (beta - 1) * B * xi * xi';                        #row09
   radn=radn/sqrt(1.d0-1.d0/dn)/sqrt(1.d0+1.d0/dn);       #row10
   if(mod(itn,intp)==0)                                   #row11
     printf("itn %4d  f %14.6e\n",itn,f);                 #row12
   endif                                                  #row13
endfor                                                    #row14
ist = 4;                                                  #row15
endfunction                                               #row16
```

Что находит программа emshor?

Пусть $f(x)$ – выпуклая функция, x^* – точка минимума, $f^* = f(x^*)$, x_0 – начальное приближение.

Теорема (о программе emshor)

Если x_0 такое, что $\|x_0 - x^*\| \leq rad$, то программа **emshor** заканчивает работу выполнением одного из условий:

1. найдена точка x_r – такая, что $f(x_r) - f^* \leq \varepsilon_f$ (ist=1),
2. **maxitn** итераций оказалось недостаточно (ist=4).

octave-функция emshor (комментарии)

```
# octave-code for emshor (version 1.1, 14.03.2019)      #row00
# шукає наближення до точки мінімуму опуклої функції   #row00a
# Вхідні параметри:                                      #row00b
#   calcfg - ім'я функції для обчислення f(x) та g(x)   #row00c
#   x0 - стартова точка, x0(1:n)                         #row00d
#   rad - радіус кулі, що локалізує точку мінімуму      #row00e
#   epsf, maxitn - параметри зупинки (точн.,макс.ітер.)#row00f
#   intp - інтервал друку (через кожні intp ітерацій)   #row00g
# Вихідні параметри:                                     #row00h
#   x - знайдене наближення до точки мінімуму, x(1:n)   #row00i
#   f - значення функції f в точці x                    #row00j
#   itn - кількість виконаних ітерацій                  #row00k
#   ist - код завершення (1 = epsf, 4 = maxitn)         #row00l
```

octave-функция emshor (код программы)

```
function [x,f,itn,ist] = emshor(calcfg,x0,rad,             #row01
                                epsf,maxitn,intp);
dn=double(length(x0)); beta=sqrt((dn-1.d0)/(dn+1.d0)); #row02
x=x0; radn=rad; B=eye(length(x));                      #row03
for (itn = 0:maxitn)                                    #row04
   [f, g1] = calcfg(x); g=B'*g1; dg=norm(g);            #row05
   if(radn*dg < epsf) ist = 1;  return; endif           #row06
   xi=(1.d0/dg)*g; dx = B * xi;                         #row07
   hs=radn/(dn+1.d0); x -= hs * dx;                     #row08
   B += (beta - 1) * B * xi * xi';                      #row09
   radn=radn/sqrt(1.d0-1.d0/dn)/sqrt(1.d0+1.d0/dn);     #row10
   if(mod(itn,intp)==0)                                 #row11
     printf("itn %4d  f %14.6e\n",itn,f);               #row12
   endif                                                #row13
endfor                                                  #row14
ist = 4;                                                #row15
endfunction                                             #row16
```

Заключение

Метод эллипсоидов можно успешно применять для нахождения x^*_ε, если количество переменных $n = 2 \div 10$.

Чтобы найти f^* с относительной точностью равной 10^{-10}, максимальное количество итераций следует из таблицы

n	itn	n	itn	n	itn
2	177	5	1144	8	2940
3	407	6	1651	9	3723
4	730	7	2249	10	4598

Вопросы?

СПАСИБО ЗА ВНИМАНИЕ!