АЛГОРИТМ ОПИСАННЫХ ЭЛЛИПСОИДОВ ДЛЯ ЗАДАЧИ О НАИЛУЧШЕМ ЛИНЕЙНОМ КЛАССИФИКАТОРЕ

Стецюк П.И.
stetsyukp@gmail.com

Институт кибернетики им. В.М. Глушкова, Киев

VII Международная научно-техническая конференция
"Информационно-компьютерные технологии – 2014"
29–30 мая 2014 г., г. Житомир

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.

Основные результаты на основе МЭ

1979 **Хачиян Л.** построил первый полиномиальный алгоритм решения задачи ЛП с рациональными коэффициентами

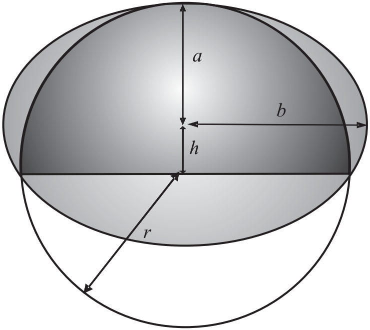
1981 **Grötchel M., Lóvasz L., Schrijver A.** разработали полиномиальные алгоритмы для ряда задач дискретной оптимизации

XI симпозиум по мат. программированию (1982)

Метод эллипсоидов и результаты на его основе 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.

Геометрия метода эллипсоидов



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.

Что такое метод эллипсоидов?

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 точка x^* (например, точка минимума выпуклой функции $f(x)$), убывает со скоростью геометрической прогрессии со знаменателем $q(n)$.

О двух вариантах метода эллипсоидов

В методе эллипсоидов Юдина-Немировского-Шора

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

В приближенном методе эллипсоидов [4]

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

Если $n = 1$, то $q(1) = 2 - \sqrt{2} \approx 0.5858$.

4. Стецюк П.И. *Приближенный метод эллипсоидов* // Кибернетика и системный анализ. – 2003. – № 3. – С. 141–146.

octave-функция emshor (ellipsoid method shor)

```
# octave-code for emshor (version 1.0, 13.09.2013)      #row00
# находит точку минимума выпуклой функции f(x)          #row00a
% Входные параметры:                                    #row00b
% calcfg - имя функции calcfg(x): вычисляет f и g(1:n) #row00c
% x  - начальная точка x0(1:n) (на выходе портится)     #row00d
% rad - радиус шара с центром в точке x0                #row00e
% epsf - точность останова по значению функции          #row00f
% maxitn - максимальное количество итераций             #row00g
% Выходные параметры:                                   #row00h
% xr -- точка минимума (рекордная)                      #row00i
% fr -- значение функции в точке xr                     #row00j
% ist -- код останова (1 = epsf, 4 = maxitn)            #row00k
function [xr,fr,ist]=emshor(calcfg,x,rad,epsf,maxitn); #row01
dn=float(length(x)); beta=sqrt((dn-1.d0)/(dn+1.d0));   #row02
radn=rad; B=eye(length(x)); fr=inf; ist=4;              #row03
for (itn = 0:maxitn)                                    #row04
   [f, g1] = calcfg(x); g=B'*g1; dg=norm(g);            #row05
   if (f < fr)  fr = f; xr = x; itr=itn; endif          #row06
   if(radn*dg < epsf) ist = 1;  return; endif           #row07
   xi=(1.d0/dg)*g; dx = B * xi; hs=radn/(dn+1.d0);      #row08
   x -= hs * dx;  B += (beta - 1) * B * xi * xi';       #row09
   radn=radn/sqrt(1.d0-1.d0/dn)/sqrt(1.d0+1.d0/dn);     #row10
   printf("itn %4d itr %4d f %16.8e fr %16.8e\n",       #row11
          itn,itr,f,fr);
endfor                                                  #row12
endfunction                                             #row13
```

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 $\|x_0 - x^*\| \leq r$, 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)

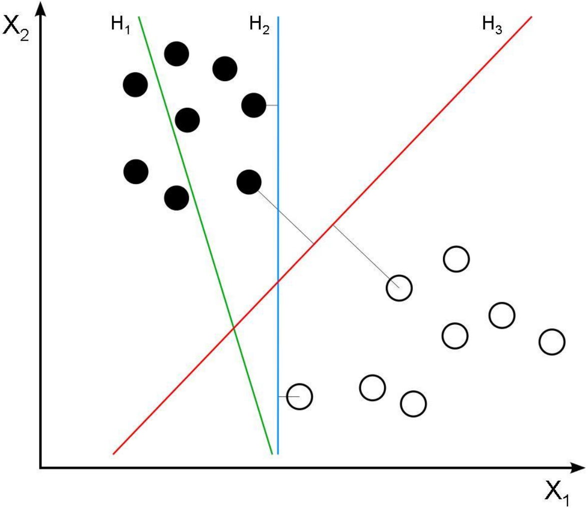
```
# octave-code for emshor (version 1.0, 13.09.2013)      #row00
# находит точку минимума выпуклой функции f(x)           #row00a
% Входные параметры:                                     #row00b
% calcfg - имя функции calcfg(x): вычисляет f и g(1:n)   #row00c
% x  - начальная точка x0(1:n) (на выходе портится)      #row00d
% rad - радиус шара с центром в точке x0                 #row00e
% epsf - точность останова по значению функции           #row00f
% maxitn - максимальное количество итераций              #row00g
% Выходные параметры:                                    #row00h
% xr -- точка минимума (рекордная)                       #row00i
% fr -- значение функции в точке xr                      #row00j
% ist -- код останова (1 = epsf, 4 = maxitn)             #row00k
```

octave-функция emshor (код программы)

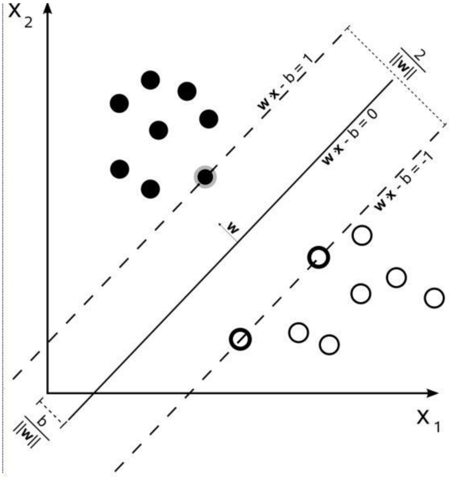
```
function [xr,fr,ist]=emshor(calcfg,x,rad,epsf,maxitn); #row01
dn=float(length(x)); beta=sqrt((dn-1.d0)/(dn+1.d0));   #row02
radn=rad; B=eye(length(x)); fr=inf; ist=4;             #row03
for (itn = 0:maxitn)                                   #row04
   [f, g1] = calcfg(x); g=B'*g1; dg=norm(g);           #row05
   if (f < fr)  fr = f; xr = x; itr=itn; endif         #row06
   if(radn*dg < epsf) ist = 1;  return; endif          #row07
   xi=(1.d0/dg)*g; dx = B * xi; hs=radn/(dn+1.d0);     #row08
   x -= hs * dx;  B += (beta - 1) * B * xi * xi';      #row09
   radn=radn/sqrt(1.d0-1.d0/dn)/sqrt(1.d0+1.d0/dn);    #row10
   printf("itn %4d itr %4d f %16.8e fr %16.8e\n",      #row11
          itn,itr,f,fr);
endfor                                                 #row12
endfunction                                            #row13
```

Содержание

Примеры линейных классификаторов H_2 и H_3



Пример наилучшего линейного классификатора



Наилучший линейный классификатор (НЛК)

Задача о наилучшем линейном классификаторе состоит в нахождении гиперплоскости с максимальным зазором, которая разделяет два множества точек $X = \{x_1, \ldots, x_{m_1}\}$ и $Y = \{y_1, \ldots, y_{m_2}\}$ евклидового пространства E^n.

Предполагается, что множества X и Y разделимы линейной гиперплоскостью $a^T z + b = 0$ (линейный классификатор).

Задача выпуклого программирования для НЛК

найти $$-2d^* = f^* = f(a^*, b^*) = \min_{(a,b)\in E^{n+1}} f(a,b) \qquad (1)$$

при ограничении $$\sum_{i=1}^{n} a_i^2 \le 1, \qquad (2)$$

где $$f(a,b) = \max\{\max_{i=1,\dots,m_1}\{-a^T x_i - b\}, \max_{j=1,\dots,m_2}\{a^T y_i + b\}.$$

Здесь $2d^*$ – максимальная ширина полосы, которая разделяет множества X и Y. Решение a^* и b^* задает гиперплоскость $(a^*)^T z + b^* = 0$, проходящую через середину полосы.

Безусловная задача для НЛК

Задача (1),(2) сводится к задаче минимизации негладкой выпуклой функции

$$F_P(a, b) = f(a, b) + P \max\{0, \sum_{i=1}^{n} a_i^2 - 1\} \qquad (4)$$

для которой при $P > d^*$ [*] ее минимум совпадает с решением задачи (1),(2).

*. Стецюк П.І., Березовський О.А., Журбенко М.Г., Кропотов Д.О. *Методи негладкої оптимізації в спеціальних задачах класифікації.* - Препр. / НАН України. Ін-т кібернетики ім. В.М. Глушкова; 2009-1. - 29 с.

О методах решения

Для нахождения минимума функции $F_P(a, b)$ можно использовать алгоритмы минимизации негладких выпуклых функций.
Если $n = 2 \div 10$, то целесообразно использовать методы эллипсоидов. Их сходимость не зависит от количества точек $m = m_1 + m_2$.

Ниже опишем такой алгоритм и сделаем это для произвольного $\alpha > 1$ – коэффициента растяжения пространства. Для удобства неизвестные (a, b) обозначим одним вектором $u \in E^{n+1}$.

Установка начальных значений

Инициализация

Положим штраф $P = \min\limits_{i,j} \|x_i - y_j\|$, стартовую точку $u_0 = 0$ и начальный радиус $r_0 = \max\limits_{i,j}\{\|x_i\|, \|y_j\|\} + 1$.

Положим $B_0 = I_{n+1}$, где I_{n+1} – единичная матрица размером $(n+1) \times (n+1)$.

Перейдем к первой итерации со значениями u_0, r_0 и B_0.

Собственно сам алгоритм

Пусть на k-й итерации найдены $u_k \in E^{n+1}$, r_k и B_k.
Для перехода к $(k+1)$-й итерации выполняем

Шаг 1. Вычислим $F_P(u_k)$ и $g(u_k) = \partial F_P(u_k)$. Если $r_k \| B_k^T \| \le \varepsilon$, то ОСТАНОВ $u^* = u_k$. Иначе переходим к шагу 2.

Шаг 2. Вычислим очередную точку

$$u_{k+1} = u_k - h_k B_k \xi_k, \text{ где } h_k = \frac{1}{2}\left(1 - \frac{1}{\alpha^2}\right), \; \xi_k = \frac{B_k^T g(u_k)}{\|B_k^T g(u_k)\|}.$$

Шаг 3. Пересчитаем матрицу B_{k+1} и радиус r_{k+1}

$$B_{k+1} = B_k + (\beta - 1)(B_k \xi_k)\xi_k^T, \; \beta = 1/\alpha, \; r_{k+1} = (\alpha + \beta) \times r_k / 2.$$

Шаг 4. Переходим к $(k+1)$-й итерации с x_{k+1}, r_{k+1} и B_{k+1}.

О сходимости алгоритма

Теорема

При любом α – таком, что $\alpha + 1/\alpha \leq 2\sqrt[n+1]{\alpha}$, алгоритм описанных эллипсоидов сходится к $u^{*}{=}(a^{*}, b^{*})$ – оптимальному решению задачи (1),(2).

Благодарности

Работа выполнена при финансовой поддержке НАН Украины, проект 07-01-14 (У)

Вопросы?

СПАСИБО ЗА ВНИМАНИЕ!