r-Алгоритмы Шора и Octave-функция ralgb5a

Стецюк П.И., Фишер А.
stetsyukp@gmail.com, Andreas.Fischer@tu-dresden.de

Институт кибернетики имени В.М. Глушкова, Киев
Технический университет Дрездена, Германия

Міжнародна наукова конференція "Сучасна інформатика: проблеми, досягнення та перспективи розвитку", присвячена 60-річчю заснування Інституту кібернетики імені В.М.Глушкова НАН України
м. Київ, 13-15 грудня 2017 р.

План доклада

1. Идея r-алгоритмов
2. $r(\alpha)$-алгоритмы
3. Octave-функция ralgb5a
4. Тестовый пример

План доклада

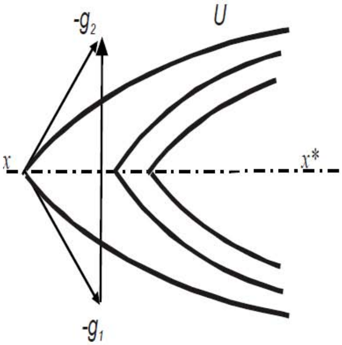
Почему r-алгоритмы?

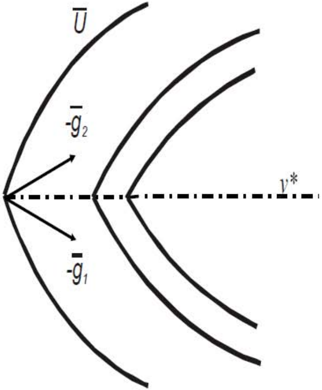
r-алгоритмы получили название от слова "разность".

Их называют r-алгоритмами Шора (1970), либо r-алгоритмами Шора–Журбенко (1971).

- Шор Н.З. Методы минимизации недифференцируемых функций и их приложения. – Докт. диссерт., Киев, 1970.
- Шор Н.З., Журбенко Н.Г. Метод минимизации, использующий операцию растяжения пространства в направлении разности двух последовательных градиентов // Кибернетика. – 1971. – № 3. – С. 51–59.

Идея r-алгоритмов (по рис. из книг Шора)





Как с помощью операции растяжения пространства

построить субградиентные алгоритмы – монотонные (почти монотонные) по минимизируемой функции?

Три статьи Шора (о сходимости r-алгоритмов)

- Шор Н.З., Журбенко Н.Г. Метод минимизации, использующий операцию растяжения пространства в направлении разности двух последовательных градиентов // Кибернетика. – 1971. – № 3. – С. 51–59.
- Шор Н.З. Исследование сходимости метода градиентного типа с растяжением пространства в направлении разности двух последовательных градиентов // Кибернетика. – 1975. – № 4. – С. 48-53.
- Шор Н.З. Монотонные модификации r-алгоритмов и их приложения // Кибернетика и системный анализ. – 2002. – № 6. – С. 74–96.

План доклада

Вычислительная схема

$r(\alpha)$-Алгоритмом называется процедура построения последовательностей $\{x_k\}_{k=0}^{\infty}$ и $\{B_k\}_{k=0}^{\infty}$ по правилу:

$$x_{k+1} = x_k - h_k B_k \xi_k, \;\; B_{k+1} = B_k R_\beta(\eta_k), \;\; k = 0, 1, 2, \dots, \qquad (1)$$

$$\text{где} \quad \xi_k = \frac{B_k^T g_f(x_k)}{\|B_k^T g_f(x_k)\|}, \quad h_k \geq h_k^\star = \operatorname*{argmin}_{h \geq 0} f(x_k - hB_k\xi_k), \qquad (2)$$

$$\eta_k = \frac{B_k^T r_k}{\|B_k^T r_k\|}, \;\; r_k = g_f(x_{k+1}) - g_f(x_k), \quad \beta = \frac{1}{\alpha} < 1. \qquad (3)$$

Здесь x_0 — начальное приближение, $B_0 = I_n$ – единичная $n \times n$-матрица, h_k – шаговый множитель, $\alpha > 1$ – коэффициент растяжения пространства, $R_\beta(\eta) = I_n + (\beta - 1)\eta\eta^T$ – оператор „сжатия“ пространства субградиентов в нормированном направлении η с коэффициентом $\beta < 1$, $g_f(x_k)$ и $g_f(x_{k+1})$ – произвольные субградиенты функции $f(x)$ в точках x_k и x_{k+1}.

Два способа регулировки шага

1. Величина h_k выбирается из условия $h_k = h_k^*$ $(h_k \approx h_k^*)$, что означает точный (приближенный) одномерный поиск минимума функции в направлении спуска.

2. Величина h_k настраивается (адаптируется) в процессе выполнения одномерного поиска минимума функции.

$r(\alpha)$-алгоритм с адаптивным шагом

оказался эффективным вариантом r-алгоритмов и на его основе разработано ряд программных реализаций (Журбенко Н.Г., Кунцевич А.В., Лиховид А.П. и др.)

Для $r(\alpha)$-алгоритма с адаптивным шагом разработаны Octave-функции: ralgb5 (2011, Стецюк П.И.), ralgb4 (2016), ralgb5a (август, 2017) – упрощенная версия ralgb5.

Стецюк П.И. Теория и программные реализации r-алгоритмов Шора // Кибернетика и системный анализ. – 2017. – № 5. – С. 43–57.

План доклада

Комментарии к Octave-функции ralgb5a

```
# ralgb5a for Shor's r-algorithm (P.Stetsyuk, August 18, 2017)

# Input parameters:
#    calcfg - name of the function calcfg(x) for calculation of f and g
#    x - the starting point, x0(1:n) (it is modified in the program)
#    alpha - the value of coefficient of space dilation
#    h0, q1 - parameters of the adaptive step adjustment
#    epsx, epsg, maxitn - stop parameters
#    intp - print information every intp iteration

# Output parameters:
#    xr - a minimum point, which was found by the program, xr(1:n)
#    fr - the value of the function f at the point xr
#    itn - the number of iterations used by the program
#    nfg - the number of function calcfg calls
#    istop - exit code (2 = epsg, 3 = epsx, 4 = maxitn, 5 = error)
```

Код octave-функции ralgb5a

```
function [xr,fr,itn,nfg,istop] = ralgb5a(calcfg,x,alpha,h0,q1,   # row001
                                 epsg,epsx,maxitn,intp);
itn = 0; B = eye(length(x)); hs = h0; lsa = 0; lsm = 0;         # row002
xr = x; [fr,g0] = calcfg(xr); nfg = 1;                          # row003
printf("itn %4d  f%15.6e  fr%15.6e  nfg %4d\n",itn,fr,fr,nfg);  # row004
if(norm(g0) < epsg) istop = 2;  return; endif                   # row005
for (itn = 1:maxitn)                                            # row006
   dx = B * (g1 = B' * g0)/norm(g1);                            # row007
   d = 1; ls = 0; ddx = 0;                                      # row008
   while (d > 0)                                                # row009
      x -= hs * dx; ddx += hs * norm(dx);                       # row010
      [f, g1] = calcfg(x); nfg ++;                              # row011
      if (f < fr)  fr = f; xr = x;  endif                       # row012
      if(norm(g1) < epsg) istop = 2;  return; endif             # row013
      ls ++; (mod(ls,3) == 0) && (hs *= 1.1);                   # row014
      if(ls > 500) istop = 5;  return; endif                    # row015
      d = dx' * g1;                                             # row016
   endwhile                                                     # row017
   (ls == 1) && (hs *= q1);  lsa=lsa+ls; lsm=max(lsm,ls);       # row018
   if(mod(itn,intp)==0)                                         # row019
     printf("itn %4d  f %14.6e  fr %14.6e", itn, f, fr);        # row020
     printf("  nfg %4d  lsa %3d  lsm %3d\n", nfg, lsa, lsm);    # row021
     lsa=0; lsm=0;                                              # row022
   endif                                                        # row023
   if(ddx < epsx) istop = 3;  return; endif                     # row024
   xi = (dg = B' * (g1 - g0) )/norm(dg);                        # row025
   B += (1 / alpha - 1) * B * xi * xi';                         # row026
   g0 = g1;                                                     # row027
endfor                                                          # row028
istop = 4;                                                      # row029
endfunction                                                     # row030
```

Выбор параметров для ralgb5a

При минимизации негладких функций рекомендуется:

$$\alpha = 2 \div 4,\ \ h_0 = 1.0,\ \ q_1 = 1.0.$$

Если известна оценка расстояния от начальной точки x_0 до точки минимума x^*, то начальный шаг h_0 целесообразно выбирать из условия $h_0 \approx \|x_0 - x^*\|$.

При минимизации гладких функций рекомендуется

$$q_1 = 0.8 \div 0.95.$$

При таком выборе параметров, как правило, среднее число шагов по направлению редко превосходит два, а за n итераций точность по функции улучшается в три-пять раз.

План доклада

Овражная кусочно-линейная функция

имеет следующий вид

$$f(x) = \sum_{i=1}^{100} (1.2)^{i-1}|x_i - 1|, \quad f^* = 0, \quad x^* = (1, 1, \dots, 1)^T.$$

Коэффициенты при $|x_i - 1|$, $i = 1, \dots, 100$, образуют геометрическую прогрессию с показателем $q = 1.2$, где минимальный коэффициент равен $(1.2)^0 = 1$, а максимальный коэффициент равен $(1.2)^{99} \approx 6.9015e^{+07}$.

Тестовый расчет

Код программы

```
function [f,g] = sabs(x)
global w
temp=x-ones(length(x),1); f=sum(abs(w.*temp)); g=w.*sign(temp);
endfunction
global w
n=100; temp=[0:(n-1)]'; w=1.2.**temp; # w(1,1), w(100,1),
x = zeros(n,1); alpha = 4.0, h0 = 10.0, q1 = 1.0,
epsx = 1.e-10, epsg = 1.e-12, maxitn = 5000, intp=500;
[xr,fr,itn,nfg,istop] = ralgb5a(@sabs,x,alpha,h0,q1,epsg,epsx,maxitn,intp);
printf("itn %4d  fr %23.15e  istop %d  nfg %4d\n", itn, fr, istop,nfg);
dx = norm(xr-ones(n,1)),
```

Протокол работы программы при **intp=500**

```
alpha =  4  h0 =  10  q1 =  1
epsx = 1.0000e-008  epsg = 1.0000e-012  maxitn =  5000
itn    0  f  4.140899e+008  fr  4.140899e+008  nfg    1
itn  500  f  1.718525e+003  fr  1.273433e+003  nfg  532  lsa 531  lsm   4
itn 1000  f  1.409472e+000  fr  1.192802e+000  nfg 1032  lsa 500  lsm   1
itn 1500  f  1.258921e-003  fr  1.258921e-003  nfg 1532  lsa 500  lsm   1
itn 2000  f  1.422859e-006  fr  1.224438e-006  nfg 2032  lsa 500  lsm   1
itn 2046  fr  6.340398755873688e-007  istop 3  nfg 2078
dx = 1.9497e-008
```

Заключение

Программа ralgb5a может быть использована для поиска точек минимума гладких и негладких выпуклых функций.

В дальнейшем ее планируется усовершенствовать так, чтобы ее можно было использовать для поиска точек равновесия, включая обобщенные равновесия Нэша, а также для решения систем нелинейных уравнений с ограничениями, вариационных неравенств и комплементарных задач.

Благодарности

Работа выполнена при частичной поддержке

Volkswagen Foundation, грант No 90 306

Вопросы?

СПАСИБО ЗА ВНИМАНИЕ!