УДК 519.85

В.Н. КУЗЬМЕНКО, Э.И. НЕНАХОВ

МЕТОД МИНИМИЗАЦИИ С ПРЕОБРАЗОВАНИЕМ ПРОСТРАНСТВА НА ОСНОВЕ КВАДРАТИЧНОЙ АППРОКСИМАЦИИ ФУНКЦИИ

Описываются метод минимизации гладких функций, основанный на локально-квадратичной оценке функции по разности ее градиентов. В случае квадратичной функции метод дает решение за конечное число шагов.

Введение. Существуют различные подходы в использовании методов, предназначенных для решения квадратичных задач, при минимизации не квадратичных, но гладких функций. Среди таких подходов отметим методы, использующие квадратичную регуляризацию оптимизационной задачи [1, 2], методы последовательного квадратичного программирования [3], методы, основанные на квадратичном приближении значений функции [4], методы, использующие значения вторых производных [5].

В настоящей работе рассматривается конечный метод минизации квадратичной функции и применение его процедур для минимизации гладких функций. Метод не вычисляет и не использует явно вторые производные, но делает их оценку, которая для квадратичной функции является точной. При решении произвольной гладкой задачи метод после ряда шагов обновляет аппроксимационную квадратичную матрицу, что позволяет изменять квадратичную аппроксимацию в зависимости от накопленной информации.

Рассмотрим квадратичную задачу

$$f(x) = \tfrac{1}{2}(x, Ax) + bx \to \min, \qquad (1)$$

где A – симметричная положительно определенная невырожденная матрица; b – вектор-строка; $(\cdot,\cdot)$ – знак скалярного произведения; $x \in X$. Рассматриваем также невырожденное преобразование пространства $x = Dx'$. Говоря о преобразовании пространства будем иметь ввиду соответствующую матрицу и наоборот.

В пространстве переменных $x' \in X'$ задача (1) имеет вид

$$\varphi(x') = \tfrac{1}{2}\left(x', D^* ADx'\right) + bDx' \to \min, \qquad (2)$$

где знаком * обозначено траспонирование.

Матрицу квадратичной функции в пространстве X' обозначим $A' = D^* AD.$

Для произвольной точки x_0 и произвольного вектора v значение $t = -(v, \nabla f(x_0))/(v, Av)$ задает сдвиг из точки x_0, при котором достигается минимум (1) функции $f(x)$ при перемещении вдоль вектора v. Значение t сохраняется при переходе в пространство X'.

Заметим также, что разность $w = \nabla f(x_0 + v) - \nabla f(x_0) = Av$ не зависит от точки x_0.

Построим преобразование D таким образом, чтобы выполнялись такие свойства:

а) вектор $v' = D^{-1}v$ – собственный для матрицы A', т. е. $A'v' = \lambda_{v'}v'$;

б) $A'e_{i_v} = e_{i_v}$, где e_{i_v} орт с единицей по координате i_v. То есть матрица A' имеет единичный столбец и строку i_v;

в) $\lambda_{v'} = 1$.

Преобразование D построим как композицию трех преобразований $D = HBZ,$ последовательно обеспечивающих выполнение свойств а), б) и в).

Преобразование H. Рассмотрим такое преобразование и его свойства $H = I - \bar{w} \cdot \bar{w}^*\left(I - \bar{v} \cdot \bar{v}^*\right)$. Символами $\bar{w}, \bar{v}$ будем обозначать нормированные векторы. Скалярное произведение векторов (a, b) будем записывать также, как $a^*b.$

1. Обратное преобразование H^{-1} определяется следующим образом. Преобразование H можно представить в виде $H = I - \bar{w}\left(\bar{w}^* - (\bar{w}, \bar{v})\bar{v}^*\right)$. Общая форма такого преобразования – это $B = I + a \cdot b^*$. Обратное к нему $B^{-1} = I - a \cdot b^* /(1 + a^*b)$ существует, если $a^*b \neq -1$. Следовательно $H^{-1} =$ $= I + \dfrac{\bar{w}\left(\bar{w}^* - (\bar{w}, \bar{v})\bar{v}^*\right)}{(\bar{w}, \bar{v})^2}$ и H^{-1} существует, если $(w, v) \neq 0$.

2. Транспонированное преобразование таково: $H^* = I - \left(\bar{w} - (\bar{w}, \bar{v})\bar{v}\right)\bar{w}^*$.

3. Для преобразования H выполняется: $Hv = v$ и, следовательно, $H^{-1}v = v$. То есть, вектор v – собственный для матриц H и H^{-1} с собственным значением 1.

4. Далее следует, что $(H^*AH)H^{-1}v = H^*Av = H^*w = (w,\bar{v})\bar{v} = \dfrac{(w,v)}{(v,v)}H^{-1}v$.

То есть вектор $v' = H^{-1}v$ в преобразованом пространстве будет собственным для матрицы $A' = H^*AH$ с собственным значением $\lambda_v = (w,v)/(v,v)$.

5. Если $v = -\nabla f(x_0)$, то $t = 1/\lambda_v$ – шаг из точки x_0 до минимума в направлении v. Заметим также, что $H^*w = \lambda_v v$.

6. Вектор w – собственный для матриц H и H^{-1} соответственно с собственными числами $\lambda_w^H = (\bar{w},\bar{v})^2$ и $\lambda_w^{H^{-1}} = 1/(\bar{w},\bar{v})^2$.

7. Если $v \neq 0$, то $(w,v) = (v,w) = (v,Av) > 0$ в силу положительной определенности матрицы A. Следовательно, матрица H^{-1} существует и существуют и не равны нулю вычисленные выше собственные числа.

Преобразование B. В качестве преобразования B возьмем преобразование Хаусхолдера (преобразование отражения). Это преобразование «зеркально» отражает пространство относительно некоторой гиперплоскости, которая задается вектором нормали ω и проходит через начало координат. Преобразование задается матрицей: $B = I - 2\bar{\omega}\bar{\omega}^*$. В нашем случае для того, чтобы сделать столбец и строку i_v новой матрицы, содержащими один ненулевой элемент i_v (т. е. диагональный элемент), «отразим» вектор $\bar{v}$ на орт e_{i_v}, и наоборот. В этом случае $\omega = e_{i_v} - \bar{v}$, а B имеет вид

$$B = I - \frac{(e_{i_v} - \bar{v})(e_{i_v} - \bar{v})^*}{1 - e_{i_v}^*\bar{v}}.$$

К общим свойствам преобразования Хаусхолдера относятся:

а) $B^* = B$; $B^*B = I$; $B\omega = -\omega$; $Bv = v$, если $(v,\omega) = 0$; $\det(B) = -1$.

Преобразование Z. Смысл преобразования Z заключается в том, чтобы сделать собственное число λ_v равным 1. Выше было показано, что $A'v' = \lambda_v v'$ после преобразования H, где $\lambda_v = (w,v)/(v,v)$. Поскольку вектор $\bar{v}$ теперь «отражен» в единичный орт e_{i_v}, то для получения единичного собственного значения в качестве матрицы Z достаточно взять диагональную матрицу со всеми элементами $z_{ii} = 1$, кроме элемента $z_{i_v i_v} = 1/\sqrt{\lambda_v} = \|v\|/\sqrt{(w,v)}$.

В результате непосредствено проверяем, что выполняются свойства а), б), в) для матрицы A', а именно:

а) $A'v' = (D^*AD)D^{-1}v = (Z^*B^*H^*AHBZ)(HBZ)^{-1}v = (HBZ)^{-1}v = v'$;

б) $A'e_{i_v} = (Z^*B^*H^*AHBZ)e_{i_v} = e_{i_v}$;

в) из а) следует, что $\lambda_{v'} = 1$.

Дополнительно отметим, что вектор v' в преобразованном пространстве X' содержит только одну ненулевую компоненту i_v со значением $\sqrt{(w,v)} = \sqrt{(v,Av)}$.

Рассмотрим следующий алгоритм решения квадратичной задачи (1).

Начальное состояние: задана начальная точка x_1 в исходном пространстве X. Положим $k=1,\ A_1 = A,\ D_0 = I,\ X_1 = X$.

1. В начале шага k имеем точку $x_k \in X_k$, квадратичную матрицу A_k и матрицу D_{k-1} текущего преобразования пространства X_k в пространство X_{k-1}.

2. Выбираем направление движения $v_k = -g^k(x_k)$, где $g^k(x_k)$ градиент квадратичной функции φ_k в пространстве X_k, и шаг t_k, минимизирующий функцию вдоль этого направления. Вычисляем следующую точку $x''_{k+1} = x_k + t_k v_k$.

3. Вычисляем вектор $w_k = g^k(x_k + v_k) - g^k(x_k) = A_k v_k$, матрицу D_k следующего преобразования пространств X_{k+1} в X_k: $D_k = H_k B_k Z_k$, где $H_k = I - \overline{w}_k \cdot \overline{w}_k^* \left(I - \overline{v}_k \cdot \overline{v}_k^*\right)$, $B_k = I - 2\overline{\omega}_k \overline{\omega}_k^*$, $\omega_k = e_k - \overline{v}_k$, неединичный элемент в Z_k равен $z_{kk} = \|v_k\| / \sqrt{(w_k, v_k)}$ и квадратичную матрицу в преобразованном пространстве $A_{k+1} = D_k^* A_k D_k$.

4. Вычисляем текущую точку $x_{k+1} = D_k^{-1} x''_{k+1}$ в пространстве X_{k+1}.

5. Если решение не найдено, то полагаем $k = k+1$ и переходим к шагу 1.

Свойства алгоритма.

1) $A_{k+1} v'_k = v'_k$, где $v'_k = D_k^{-1} v_k = -D_k^{-1} g^k(x_k)$, в силу построения D_k;

2) $v'_k \in X_{k+1}$ имеет одну не нулевую компоненту k;

3) $D_{k+1}^{-1} v'_k = v'_k$, так как выполняются свойства 1), 2) и $(v_{k+1}, v'_k) = 0$, где $v_{k+1} = -g^{k+1}(x_{k+1})$, а точка x_{k+1} получена путем минимизации по направлению v'_k. То есть преобразование D_{k+1}^{-1} не изменяет вектор v'_k;

4) $A_{k+2} v'_k = v'_k$, так как выполняются свойства 1), 2), 3);

5) на шаге $k+1$ для всех $v'_i,\ i = 1,\dots,k$ направлений движения предыдущих шагов, пересчитанных в пространство X_{k+1}, выполняется: а) $A_{k+1} v'_i = v'_i$, б) $(v'_i, v'_j) = 0$, $j \neq i,\ i, j = 1,\dots,k$, в) v'_i имеет одну не нулевую компоненту i;

6) матрица A_{k+1} имеет k первых единичных строк и столбцов. Матрица A_{n+1} становится единичной;

7) алгоритм находит минимум функции не более, чем за n шагов, где n – размерность пространства X;

8) для работы алгоритма нет необходимости знать матрицу и вектор квадратичной фунции в пространсте X_k. Достаточно уметь вычислять градиент $\nabla f(x)$ в исходном пространстве X и знать полное преобразование P_k пространства X_k в исходное X: $P_k = D_1 D_2 \ldots D_k$.

Свойство 5) доказывается по индукции. Действительно, для $k = 2$ это свойства 1) – 4).

Имеем $(g^{k+1}(x_{k+1}), v'_k) = 0$ в силу того, что x_{k+1} минимум по направлению v'_k; $g^{k+1}(x_{k+1}) = A_{k+1}(x_k + t_k v'_k) + b_{k+1} = (A_{k+1}x_k + b_{k+1}) + t_k v'_k = g^{k+1}(x_k) + t_k v'_k$.

Следовательно $(g^{k+1}(x_k) + t_k v'_k, v'_{k-1}) = 0$, так как x_k минимум по направлению v'_{k-1} и $(v'_k, v'_{k-1}) = 0.$ Далее $g^{k+1}(x_k)$ выражается через $g^{k+1}(x_{k-1})$ и v'_k, v'_{k-1} и так далее. Получаем, что $(v_{k+1}, v'_i) = 0$ для всех $i = 1, \ldots, k.$ Отсюда по аналогии с 3), 4) получаем, что $A_{k+2}v'_i = v'_i$, для $i = 1, \ldots, k.$ Для v'_{k+1} выполняется $(v'_{k+1}, v'_i) = 0,$ $i = 1, \ldots, k$ в силу 2).

Свойство 8) приведенного алгоритма позволяет использовать его для минимизации не только квадратичных функций, но и гладких выпуклых функций.

Пусть определена выпуклая гладкая функция $f(x)$ в пространстве $X = R^n$ такая, что $f(x) \to \infty$ при $\|x\| \to \infty$.

Рассмотрим следующий алгоритм нахождения минимума $f(x)$.

Начальное состояние: задана начальная точка x_0. Положим $k = 0,$ $P_0 = I.$

1. В начале шага k имеем текущую точку x_k в исходном пространстве X, матрицу текущего преобразования пространства P_k: $x = P_k x'.$ Положим $P_k = I,$ если k кратно n.

2. Вычисляем функцию $f(x_k)$ и градиент $\nabla f(x_k).$ Градиент в преобразованном пространстве X' вычисляется как $P_k^* \nabla f(x_k).$ Преобразование этого вектора обратно в пространство X дает вектор $s_k = P_k P_k^* \nabla f(x_k),$ вдоль которого будем искать минимум функции $f(\cdot)$ из точки x_k на текущей итерации.

3. Возьмем некий шаг длиной $\beta_k > 0$ в направлении противоположном градиенту $P_k^* \nabla f(x_k)$ в пространстве X': $v_k = -\beta_k P_k^* \nabla f(x_k),$ рассмотрим также такую разность градиентов: $w_k = P_k^* \big(\nabla f(x_k - \beta_k s_k) - \nabla f(x_k)\big)$ в X'.

4. Изменим преобразование пространства следующим образом: $P_{k+1} = P_k H_k B_k Z_k,$ где H_k, B_k, Z_k описаны выше.

5. Найдем $\alpha_k = \arg\min\limits_{\alpha} \{ f(x_k - \alpha s_k) \}$ и положим $x_{k+1} = x_k - \alpha_k s_k.$

6. Положим $k = k + 1.$ Если не выполняются критерий останова $\|\nabla f(x_k)\| < \varepsilon_g$, то перейдем на шаг 1.

При вычислительной реализации алгоритма необходимо уделять внимание численной устойчивости. Поэтому на некоторых итерация преобразования H_k и B_k могут не выполняться. А именно, H_k не выполняется, если $\left|\left(\overline{w}_k,\overline{v}_k\right)\right| \le \varepsilon$, B_k не выполняется, если $\left|1-\left(e_{i_k},\overline{v}_k\right)\right| \le \varepsilon$, что обусловлено сохранением устойчивости обратных преобразований.

Второй необходимый момент – это обновление матрицы P_k, что связано со «сбросом» накопленных ошибок вычислений и(или) необходимостью изменения квадратичной аппроксимации.

В.М. Кузьменко, Е.І. Ненахов

МЕТОД МІНІМІЗАЦІЇ З ПЕРЕТВОРЕННЯМ ПРОСТОРУ, ЩО ГРУНТУЄТЬСЯ НА КВАДРАТИЧНІЙ АПРОКСІМАЦІЇ ФУНКЦІЇ

Розглядається метод мінімізації гладких функцій, який грунтується на локально-квадратичній оцінці функції за різницею її градієнтів. У випадку квадратичної функції метод дає розв’язок за скінчену кількість кроків.

V.M. Kuzmenko, E.I. Nenakhov

THE MINIMIZATION METHOD WITH SPACE TRANSFORMATION BASED ON QUADRATIC APPROXIMATION OF FUNCTION

A method for smooth function minimization based on local quadratic estimation of the function using difference of its gradients is considered. In case of quadratic function the method gives solution after finite number of steps.

1. *Пшеничный Б.Н., Ненахов Э.И., Кузьменко В.Н.* Модифицированный метод отсечения для минимизации выпуклой функции // Кибернетика и системный анализ. – 1997. – № 6.– С. 142 – 149.
2. *Пшеничный Б.Н., Ненахов Э.И., Кузьменко В.Н.* Комбинированный метод решения общей задачи выпуклого программирования // Там же. – 1998. – № 4. – С. 121 – 134.
3. *Gill P.E., Murray W., Saunders M.A.* SNOPT: An SQP algorithm for large-scale constrained optimization // SIAM Journal on Optimization. – 2002. – Vol. 12. – P. 979 – 1006.
4. *Бойко В.В., Кузьменко В.Н.* Использование квадратичного приближения целевой функции в PNK-методе // Теорія оптимальних рішень. – К.: Ін-т кібернетики імені В.М. Глушкова НАН України, 2010. – С. 120 – 125.
5. *Fletcher R.* Practical Methods of Optimization // J. Wiley and Sons, Chichester, England. – 1987. – 276 p.

Получено 14.04.2015