

Методы оптимизации

Лекция 4

31 марта 2023 г.

МЕТОДЫ МНОГОМЕРНОЙ МИНИМИЗАЦИИ НУЛЕВОГО ПОРЯДКА

Во многих задачах минимизируемая функция задаётся с помощью некоторого алгоритма вычисления её значений в произвольной точке. Вид алгоритма может быть неизвестен (например, вычисление значений функции производится либо с помощью модели, либо на реальном объекте) или он может быть столь сложен, что аналитическое вычисление градиента слишком громоздко. Во всех этих случаях единственная информация, которой мы располагаем, — значения $f(x)$. Методы, использующие только эту информацию, называются методами нулевого порядка (часто говорят также о прямых методах, методах поиска или о методах без вычисления производных).

Наиболее прямолинейная стратегия в такой ситуации заключается в использовании значений функции для конечно-разностной аппроксимации производных. Более экономный способ связан с учётом значений функции в предыдущих точках. Наконец, известен ряд специфических методов нулевого порядка, не имеющих аналогов среди методов первого или второго порядков.

Для оценки градиента функции $f : \mathbb{R}^n \rightarrow \mathbb{R}^1$ в точке x составим конечно-разностные отношения

$$\Delta_1 = \alpha^{-1}[f(x + \alpha y) - f(x)], \quad \Delta_2 = (2\alpha)^{-1}[f(x + \alpha y) - f(x - \alpha y)]$$

где $y \in \mathbb{R}^n$ — произвольный вектор.

Если f дифференцируема в x , то $|\Delta_1 - (\nabla f(x), y)| \rightarrow 0$ при $\alpha \rightarrow 0$.

Если ∇f удовлетворяет условию Липшица с константой L

в окрестности точки x , то при достаточно малых α

$$|\Delta_1 - (\nabla f(x), y)| \leq L\alpha\|y\|^2/2.$$

Если f дважды дифференцируема и ∇f удовлетворяет условию

Липшица с константой L в окрестности точки x , то при достаточно

$$\text{малых } \alpha \quad |\Delta_2 - (\nabla f(x), y)| \leq L\alpha^2\|y\|^3/6.$$

Если f квадратична, то при любом α : $\Delta_2 = (\nabla f(x), y)$.

Таким образом, разностные отношения Δ_1 и Δ_2 могут служить приближением для линейной аппроксимации $f(x)$.

Рассмотрим методы вида

$$x^{k+i} = x^k - \gamma_k s^k$$

где $\gamma_k \geq 0$ — длина шага, а s^k вычисляется по одной из двух формул

$$s^k = \sum_{i=1}^m \alpha_k^{-1} [f(x^k + \alpha_k h^i) - f(x^k)] h^i,$$

$$s^k = \sum_{i=1}^m (2\alpha_k)^{-1} [f(x^k + \alpha_k h^i) - f(x^k - \alpha_k h^i)] h^i$$

Здесь h^i , $i = 1, \dots, m$, — векторы, задающие направления пробных шагов, α_k — длина пробного шага.

Выбирая различные h^i и m , получим те или иные алгоритмы.

Разностный аналог градиентного метода: $m = n$, $h^i = e_i$
 $i = 1, \dots, n$, где e_i — координатные орты, у которых i -я координата равна единице, а остальные — равны нулю.

Иначе говоря, пробные шаги делаются по координатным осям, так что исходный метод в координатной записи принимает вид

$$x_i^{k+1} = x_i^k - (\gamma_k / \alpha_k) [f(x^k + \alpha_k e_i) - f(x^k)] .$$

В соответствии с полученными выше оценками

$$s^k = \sum_{i=1}^m (\nabla f(x^k), e_i) e_i + \varepsilon^k = \nabla f(x^k) + \varepsilon^k,$$

где остаточный член ε^k может быть оценён в зависимости от гладкости $f(x)$.

Метод покоординатного спуска: $m = 1$, $h^i = e_i$ при $i = (k \bmod n)$,
и $h^i = 0$ при $i \neq (k \bmod n)$.

Шаги делаются по координатным осям, выбираемым в циклическом порядке:

$$x_i^{k+1} = \begin{cases} x_i^k - (\gamma_k / \alpha_k) [f(x^k + \alpha_k e_i) - f(x^k)], & i = (k \bmod n), \\ x_i^k & \text{в противном случае} \end{cases}$$

При этом $s^k = \nabla f(x^k)_j e_j + \varepsilon^k$.

Метод случайного покоординатного спуска: $m = 1$, $h = e_j$, где j принимает значения $1, \dots, n$ с равной вероятностью. Шаг делается, как и выше, по координатным осям, но они выбираются в случайном порядке.

Метод случайного поиска: $m = 1$, h — случайный вектор, равномерно распределённый на единичной сфере. Здесь движение производится по случайному направлению, а знак и величина шага определяются разностным отношением:

$$x^{k+1} = x^k - (\gamma_k / \alpha_k) [f(x^k + \alpha_k h) - f(x^k)] h.$$

Алгоритм с парными пробами. В случайном направлении по обе стороны от исходной точки производятся пробные шаги и осуществляется шаг в направлении меньшего значения целевой функции.

Алгоритм с возвратом при неудачном шаге.

Если значение функции в новой точке больше или равно значению функции в исходной точке, т. е. случайная проба оказалась неудачной, то остаются в исходном состоянии, после чего снова делается шаг в новом случайном направлении.

Алгоритм.

1. Задать начальную точку x^0 и параметры алгоритма $\alpha_0 > 0$, $\lambda \in (0, 1)$, $\varepsilon > 0$. Положить $k := 0$.
2. Положить $i := (k \bmod n) + 1$ и $p^k := e_i$.
3. Вычислить значение функции $f(x)$ в точках x^k и $x^k + \alpha_k p^k$. Если выполнено неравенство $f(x^k) > f(x^k + \alpha_k p^k)$, то $x^{k+1} := x^k + \alpha_k p^k$, $\alpha_{k+1} := \alpha_k$, $k := k + 1$ и перейти к п. 2.
4. Вычислить значение функции $f(x)$ в точке $x^k - \alpha_k p^k$. Если выполнено неравенство $f(x^k) > f(x^k - \alpha_k p^k)$, то $x^{k+1} := x^k - \alpha_k p^k$, $\alpha_{k+1} := \alpha_k$, $k := k + 1$ и перейти к п. 2.
5. Принять $x^{k+1} := x^k$. Если $i = n$, то $\alpha_{k+1} := \lambda \alpha_k$, иначе $\alpha_{k+1} := \alpha_k$.
6. Если $i = n$ и $k \geq n$, то проверить условие останова $\|x^k - x^{k-n}\| < \varepsilon$. В случае выполнения, расчёт закончить и принять $x^* := x^k$. Иначе $k := k + 1$ и перейти к п. 2.

СЕТОЧНЫЙ ПОИСК

В методе Хука—Дживса поиск фактически осуществляется по узлам некоторой сетки.

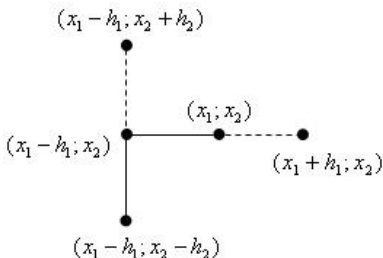
Метод является модификацией метода покоординатного спуска.

Идея метода заключается в том, что периодически поиск проводится в дополнительных направлениях помимо координатных, что может ускорить сходимость.

По существу, метод Хука—Дживса представляет собой комбинацию «исследующего» поиска с циклическим изменением переменных и «поиска по образцу» с использованием определённых эвристических правил.

Исследующий поиск ориентирован на выявление характера локального поведения целевой функции и определение дополнительного направления поиска, в котором происходит смещение при выполнении фазы поиска по образцу.

Поиск начинается в некоторой исходной точке x^0 . Если значение целевой функции в пробной точке не превышает значения функции в исходной точке, то шаг поиска рассматривается как успешный. В противном случае необходимо вернуться в предыдущую точку и сделать шаг в противоположном направлении с последующей проверкой значения целевой функции. После перебора всех n координат исследующий поиск завершается. Полученную в результате точку z^{n+1} называют базовой.

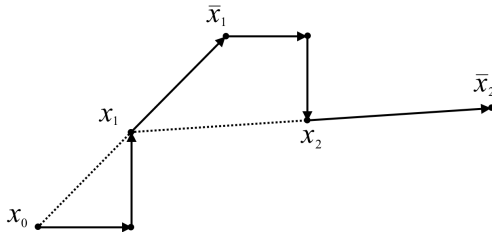


Поиск по образцу заключается в реализации единственного шага из полученной базовой точки вдоль прямой, соединяющей эту точку с предыдущей базовой точкой.

Новая точка образца определяется в соответствии с формулой

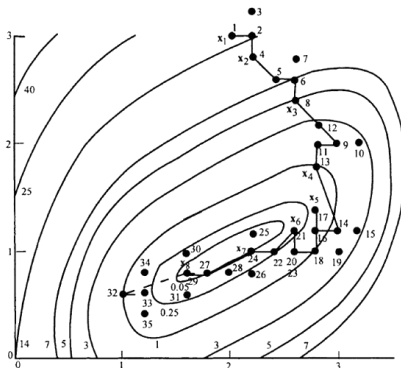
$$\bar{x}^{k+1} = x^{k+1} + \lambda(x^{k+1} - x^k),$$

где $\lambda > 0$ — параметр алгоритма, обычно выбирающийся равным 1.



Точка \bar{x}^{k+1} фиксируется в качестве следующей базовой точки, и вновь проводится исследующий поиск.

Если в результате получается точка с меньшим значением целевой функции, чем в точке \bar{x}^{k+1} то она рассматривается как новая базовая точка. Если исследующий поиск неудачен, необходимо вернуться в точку \bar{x}^{k+1} и уменьшить величину шага путём введения некоторого множителя и возобновить исследующий поиск. Поиск завершается, когда величина шага становится достаточно малой.



Алгоритм.

1. Задать: начальную точку x^0 , шаги перемещения h_i по каждому координатному направлению $i = 1, \dots, n$, точность вычисления минимума $\varepsilon > 0$, коэффициент убывания шага $\alpha > 0$, длина шага при поиске по образцу λ . Инициализировать счётчик итераций $k := 0$ счётчик циклического покоординатного спуска $i := 1$, и промежуточную точку для покоординатного спуска $z^1 := x^0$.
2. Провести исследовательский поиск в прямом направлении. Если $f(z^i + h_i e_i) < f(z^i)$, то $z^{i+1} = z^i + h_i e_i$ и перейти к п. 4. В противном случае перейти к следующему шагу.
3. Провести исследовательский поиск в обратном направлении. Если $f(z^i - h_i e_i) < f(z^i)$, то $z^{i+1} = z^i - h_i e_i$. Иначе положить $z^{i+1} = z^i$ и перейти к следующему шагу.

Алгоритм.

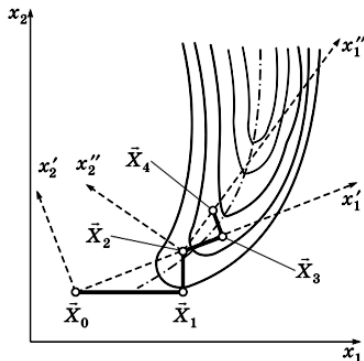
4. Продолжение покоординатного спуска. Если $i < n$, то $i := i + 1$ и перейти к п. 2. Иначе выполнить проверку $f(z^{n+1}) < f(x^k)$, в случае если неравенство выполнено то перейти к п. 5, иначе перейти к п. 6.

5. Произвести шаг поиска по образцу. Установить $x^{k+1} := z^{n+1}$ и $z^1 := x^{k+1} + \lambda(x^{k+1} - x^k)$. Положить $k := k + 1$, $i := 1$ и перейти к п. 2.

6. Проверка завершения. Если $\max_{1 \leq i \leq n} h_i < \varepsilon$, то поиск прекратить и положить $x^* := x^k$. В противном случае уменьшить шаг $h_i := h_i / \alpha$, $i = 1, \dots, n$. Установить $x^{k+1} := x^k$, $k := k + 1$, $i := 1$ и перейти к п. 2.

Существует ряд модификаций метода сеточного поиска путём введения системы ортогональных направлений поиска, ориентация которой изменяется на каждой итерации.

Например, Розенброком разработан метод, основанный на непрерывном изменении множества векторов, используемых при исследующем поиске, с помощью процедуры ортогонализации.



МЕТОДЫ НЕЛОКАЛЬНОЙ ЛИНЕЙНОЙ АППРОКСИМАЦИИ

В конечно-разностном градиентном методе пробные и рабочие шаги были разделены — точки $x^k + \alpha_k e_i$ служили только для оценки градиента в x^k ; в x^{k+1} вся работа проводится заново. Можно поступить и иначе, и строить линейную аппроксимацию по набору точек, расположенных достаточно далеко.

Типичным примером служит так называемый симплексный метод.

Симплексный метод оптимизации, впервые предложенный Спендли, Хекстом и Хемсвортом в 1962 г., совмещает процессы изучения поверхности отклика (зависимость $f(x)$ в $(n + 1)$ -мерном пространстве) и движения по ней.

(не путать с симплекс-методом в линейном программировании!).

Определение

Симплекс — это выпуклая оболочка $n + 1$ точки, которые предполагаются аффинно независимыми (то есть не лежат в подпространстве размерности $n - 1$).

Эти точки называются вершинами симплекса. Число n называется размерностью симплекса.

Определение

Правильным n -мерным симплексом является совокупность $n + 1$ равноудалённых друг от друга точек в n -мерном пространстве.

Например, в случае двух переменных правильным симплексом является равносторонний треугольник. В трёхмерном пространстве правильный симплекс представляет собой тетраэдр.

Если $x^0 = (x_1, \dots, x_n)$ — начальное приближение к точке минимума, то в качестве начального симплекса можно выбрать симплекс с вершинами

$$\begin{aligned} x^0 &= (x_1, \dots, x_n), \\ x^1 &= (x_1 + r, x_2 + s, x_3 + s, \dots, x_n + s) \\ x^2 &= (x_1 + s, x_2 + r, x_3 + s, \dots, x_n + s) \\ x^3 &= (x_1 + s, x_2 + s, x_3 + r, \dots, x_n + s) \\ &\vdots \\ x^n &= (x_1 + s, x_2 + s, x_3 + s, \dots, x_n + r) \end{aligned}$$

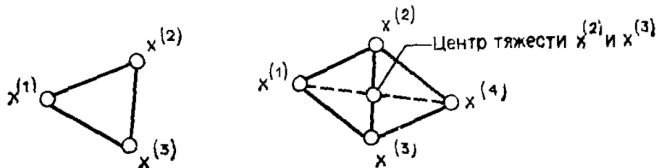
где

$$r = \frac{\alpha(\sqrt{n+1}-1+n)}{n\sqrt{2}}, \quad s = \frac{\alpha(\sqrt{n+1}-1)}{n\sqrt{2}}$$

Указанный симплекс является правильным, с длиной ребра α .

Симплекс обладает тем важным для его дальнейшего применения свойством, что в результате отбрасывания нескольких его вершин можно, используя оставшиеся вершины, получить новый симплекс с помощью добавления нескольких новых вершин. Это свойство лежит в основе построения итеративных алгоритмов оптимизации, использующих смещение симплекса по поверхности отклика.

Симплекс можно построить на любой грани начального симплекса путём переноса выбранной вершины на надлежащее расстояние вдоль прямой, проведённой через центр тяжести остальных вершин начального симплекса.



Построение нового симплекса

Пусть выбраны $n + 1$ точек x^0, x^1, \dots, x^n , образующие вершины правильного симплекса. Вычислим значения $f(x)$ в вершинах и найдём ту, для которой $f(x)$ максимальна: $j = \arg \max_{1 \leq i \leq n} f(x^i)$.

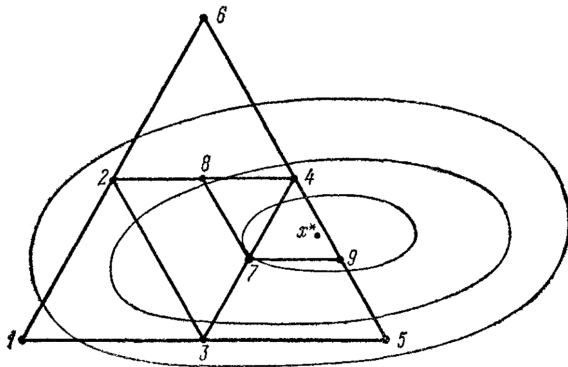
Если таких вершин несколько, то может быть взята любая из них.

Построим новый симплекс, отличающийся от старого лишь одной вершиной; x^j заменяется на x^{n+1} :

$$x^{n+1} = \frac{2}{n}(x^0 + \dots + x^{j-1} + x^{j+1} + \dots + x^n) - x^j,$$

т. е. x^{n+1} симметрично с x^j относительно грани, противолежащей x^j .

Если окажется, что в новом симплексе максимум достигается в x^{n+1} , то возвращаемся к исходному симплексу, заменив x^j на вершину, в которой значение $f(x)$ максимально среди оставшихся вершин и т. д. Если какая-либо точка сохраняется в $n + 1$ последовательном симплексе, то последний симплекс сокращается вдвое подобным преобразованием с центром в этой вершине.



Простейший вариант симплексного метода

Метод обладает рядом очевидных преимуществ

1. Расчёты и логическая структура метода отличаются сравнительной простотой, и, следовательно, соответствующая программа оказывается относительно короткой.
2. Уровень требований к объёму памяти невысокий.
3. Используется сравнительно небольшое число заранее установленных параметров: масштабный множитель α (длина ребра начального симплекса), коэффициент уменьшения множителя и параметры окончания поиска.
4. Алгоритм оказывается эффективным даже в тех случаях, когда ошибка вычисления значений целевой функции велика, поскольку при его реализации оперируют наибольшими значениями функции в вершинах, а не наименьшими.

Алгоритм обладает также рядом существенных недостатков.

1. Могут возникать трудности, связанные с масштабированием, поскольку все координаты вершин симплекса зависят от одного и того же масштабного множителя α . Чтобы обойти трудности такого рода, в практических задачах следует промасштабировать все переменные с тем, чтобы их значения были сравнимыми по величине.
2. Алгоритм работает слишком медленно, так как полученная на предыдущих итерациях информация не используется для ускорения поиска.
3. Не существует простого способа расширения симплекса, не требующего пересчёта значений целевой функции во всех точках образца. Таким образом, если α по какой-либо причине уменьшается (например, если встречается область с узким «оврагом»), то поиск должен продолжаться с уменьшенной величиной шага.

Ранее был описан лишь простейший вариант симплексного метода. Существует много его модификаций, в которых симплекс не обязательно правильный, а величина шага и условия дробления могут быть иными.

Нелдером и Мидом предложен способ изменения скорости движения симплекса путём его расширения или сжатия в зависимости от того, удачен был шаг или нет.

Этот метод также известен как *метод деформируемого многогранника*.

Опишем этот метод, являющийся одним из самых известных симплексных методов.

На N -м шаге для текущего симплекса S_N , который задаётся $(n + 1)$ вершинами x^0, x^1, \dots, x^n , оцениваются значения $y^i = f(x^i)$, $i = 1, \dots, n + 1$ и выбираются вершины x^h, x^l с максимальным и минимальным значениями целевой функции $f(x)$ соответственно:

$$y^h = \max_i (f(x^i)) = \max_i (y^i)$$

$$x^h = \arg \max_i (f(x^i))$$

$$y^l = \min_i (f(x^i)) = \min_i (y^i)$$

$$x^l = \arg \min_i (f(x^i))$$

1. *Отражение*. Оно осуществляется согласно соотношению

$$x^* := \bar{x} + \alpha(\bar{x} - x^h),$$

где $\alpha > 0$ — коэффициент отражения; \bar{x} — центр n вершин x^i , $i = 1, \dots, n+1$, $i \neq h$.

Если $y^l \leq y^* \leq y^h$, где $y^* = f(x^*)$, то x^h заменяется на x^* и переход к п. 1 с новым симплексом.

2. *Растяжение*. Если $y^* < y^l$, то есть если с помощью отражения был найден новый минимум, тогда вектор $x^* - \bar{x}$ удлинняется в соответствии с соотношением

$$x^{**} := \bar{x} + \gamma(x^* - \bar{x}),$$

где $\gamma > 1$ — коэффициент растяжения. Если $y^{**} < y^l$, где $y^{**} = f(x^{**})$, то x^h заменяется на x^{**} и процедура продолжается снова с п. 1 с новым симплексом. В противном случае x^h заменяется на x^* , и также осуществляется переход к п. 1.

3. *Сжатие*. Если после отражения $y^* \geq y^i$ для всех $i \neq h$, то есть при замене вершины x^h на x^* максимальное значение функции останется равным y^* , тогда в качестве x^h возьмем старую точку x^h или x^* в зависимости от того, где меньше значение функции, и осуществляем сжатие вектора $x^h - \bar{x}$ в соответствии с формулой

$$x^{**} := \bar{x} + \beta(x^h - \bar{x}),$$

где $0 < \beta < 1$ — коэффициент сжатия.

Если $y^{**} < y^h$, то x^h заменяют на x^{**} и переходят к п. 1.

4. *Редукция*. Если $y^{**} \geq y^h$, то все векторы $x^i - x^l$ ($i = 1, \dots, n + 1$) уменьшаются в $1/\delta$ раз в соответствии с формулой

$$x^i := x^i + \delta(x^i - x^l), \quad i = 1, \dots, n + 1,$$

где $0 < \delta < 1$ — коэффициент редукции.

Затем возвращаются к п. 1 и продолжается поиск минимума с новым симплексом.

В качестве критерия останова Нелдер и Мид предложили использовать условие

$$\left[\frac{1}{n} \sum_{i=1}^{n+1} (y^i - \bar{y})^2 \right]^{1/2} \leq \varepsilon$$

где $\varepsilon > 0$ — заданное малое число. Этот критерий предпочтительнее критерия, связанного с вариациями независимых переменных, так как в оврагах с острым дном симплекс может сжиматься до весьма малых размеров.

Кроме того, можно использовать ещё два критерия останова:

$$y^h - y^l \leq \varepsilon$$

$$\left[\frac{1}{n} \sum_{i=1}^{n+1} (y^i - \bar{y})^2 \right]^{1/2} \leq \varepsilon \left[\sum_{i=1}^{n+1} (x^i - \bar{x})^2 \right]^{1/2}$$

Проверка на достижение минимума осуществляется каждый раз, когда осуществляется возврат к п. 1. Выбор того или иного критерия прекращения поиска диктуется спецификой решаемой задачи.

Стратегия поиска определяется значениями четырёх коэффициентов — α , γ , β , δ и в меньшей степени — формой исходного симплекса.

В качестве удовлетворительных значений этих параметров при решении задачи безусловной оптимизации Нелдер и Мид рекомендовали $\alpha = 1$, $\gamma = 2$, $\beta = 0,5$ и $\delta = 0,5$.



Вопросы?