# Методы оптимизации высокого порядка

Киселев Егор, Б05-931

9 декабря 2021 г.

## 1 Описание методов

В методах второго порядка при поиске минимума используют информацию о функции и её производных до второго порядка включительно. К этой группе относят метод Ньютона и его модификации.

### Метод Ньютона

В основе метода лежит квадратичная аппроксимация функции $f(\mathbf{x})$ в точке $\mathbf{x}_k$, которую можно получить, отбрасывая в рядах Тейлора члены третьего и более высокого порядков:

$$f(\mathbf{x}) \approx f(\mathbf{x}_k) + \nabla^T f(\mathbf{x}_k) \cdot (\mathbf{x} - \mathbf{x}_k) + \frac{1}{2}(\mathbf{x} - \mathbf{x}_k)^T \cdot \nabla^2 f(\mathbf{x}_k) \cdot (\mathbf{x} - \mathbf{x}_k).$$

Заменим $\mathbf{x}$ на $\mathbf{x}_{k+1}$ и обозначим $\Delta\mathbf{x}_k = \mathbf{x}_{k+1} - \mathbf{x}_k$:

$$f(\mathbf{x}_{k+1}) \approx f(\mathbf{x}_k) + \nabla^T f(\mathbf{x}_k) \cdot \Delta\mathbf{x}_k + \frac{1}{2}(\Delta\mathbf{x}_k)^T \cdot \nabla^2 f(\mathbf{x}_k) \cdot \Delta\mathbf{x}_k.$$

Минимум функции $f(\mathbf{x})$ в направлении $\Delta\mathbf{x}_k$ определяется дифференцированием $f(\mathbf{x})$ по каждой из компонент $\Delta\mathbf{x}$ и приравниванием к нулю полученных выражений:

$$\nabla f(\mathbf{x}_k) + \nabla^2 f(\mathbf{x}_k) \cdot \Delta\mathbf{x}_k = \mathbf{0},$$

Откуда следует формула шага:

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \left(\nabla^2 f(\mathbf{x}_k)\right)^{-1} \cdot \nabla f(\mathbf{x}_k).$$

Заметим, что если $f(\mathbf{x})$ — квадратичная функция (выпуклая вниз), то для достижения минимума достаточно одного шага, но в общем случае, когда требуется несколько шагов, итерационная формула с длиной шага $\alpha_k$ имеет такой вид:

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \alpha_k \left(\nabla^2 f(\mathbf{x}_k)\right)^{-1} \cdot \nabla f(\mathbf{x}_k).$$

Убывание (сходимость) метода Ньютона гарантируется (в предположении, что $f(\mathbf{x})$ дважды дифференцируема), при условии положительной определенности $\left(\nabla^2 f(\mathbf{x}_k)\right)^{-1}$:

$$\left(\nabla^2 f(\mathbf{x}_k)\right)^{-1} \succ 0 \Rightarrow -\nabla^T f(\mathbf{x}_k) \cdot \Delta\mathbf{x}_k = \left(\nabla f(\mathbf{x}_k)\right)^T \cdot \left(\nabla^2 f(\mathbf{x}_k)\right)^{-1} \cdot \nabla f(\mathbf{x}_k) > 0.$$

Из плюсов данного метода можно выделить:

- Квадратичная сходимость,
- Высокая точность решения.

Из минусов:

- $O(n^2)$ памяти для хранения гессиана,
- Асимптотика $O(n^3)$,
- Гессиан может оказаться вырожденным.

Наибольшие проблемы с эффективностью в большинстве случаев вызывает вычисление обратного гессиана на каждом шаге. Кроме того, необходимое условие сходимости метода — положительная определённость гессиана, что не всегда правда. Чтобы решить эти проблемы можно вычислять приближенный обратный гессиан. Есть несколько способов такого приближения, на которых основаны *Квазиньютоновские методы*. Рассмотрим самый популярный из них — BFGS.

### BFGS

Алгоритм назван в честь его авторов Broyden, Fletcher, Goldfarb, Shanno. Суть метода состоит в вычислении гессиана исходя из его значения на предыдущем шаге:

$$H_{k+1} = H_k - \frac{H_k \Delta \mathbf{x}_k (\Delta \mathbf{x}_k)^T H_k^T}{(\Delta \mathbf{x}_k)^T H_k \Delta \mathbf{x}_k} + \frac{\Delta \mathbf{y}_k (\Delta \mathbf{y}_k)^T}{(\Delta \mathbf{y}_k)^T \Delta \mathbf{x}_k},$$

где $\Delta \mathbf{y}_k = \nabla f(\mathbf{x}_{k+1}) - \nabla f(\mathbf{x}_k)$. Но поскольку вычислять обратную матрицу тоже затруднительно, то используется формула для обратного гессиана:

$$H_{k+1}^{-1} = \left(I - \rho_k \Delta \mathbf{x}_k (\Delta \mathbf{y}_k)^T\right) H_k^{-1} \left(I - \rho_k \Delta \mathbf{y}_k (\Delta \mathbf{x}_k)^T\right) + \rho_k \Delta \mathbf{x}_k (\Delta \mathbf{x}_k)^T,$$

где $\rho_k = \frac{1}{(\Delta \mathbf{y}_k)^T \Delta \mathbf{x}_k}$. В качестве начального приближения гессиана можно брать любую невырожденную, хорошо обусловленную матрицу. Зачастую используют единичную матрицу.

Коэффициент $\alpha_k$ находим используя линейный поиск, где $\alpha_k$ удовлетворяет условиям Вольфе:

$$f(\mathbf{x}_k + \alpha_k \mathbf{s}_k) \leq f(\mathbf{x}_k) + c_1 \alpha_k \left(\nabla f(\mathbf{x}_k)\right)^T \mathbf{s}_k$$
$$\left(\nabla f(\mathbf{x}_k + \alpha_k \mathbf{s}_k)\right)^T \mathbf{s}_k \geq c_2 \left(\nabla f(\mathbf{x}_k)\right)^T \mathbf{s}_k,$$

Константы $c_1$ и $c_2$ выбирают следующим образом: $0 \leq c_1 \leq c_2 \leq 1$. В большинстве реализаций: $c_1 = 0.0001$ и $_2 = 0.9$.

Фактически мы находим такое $\alpha_k$ при котором значение функции $f(\mathbf{x}_k + \alpha_k \mathbf{s}_k)$ минимально. Тогда мы сделаем шаг в минимальную на выбранном направлении точку.

# 2 Сравнение методов

Методы первого порядка используют информацию лишь о первых производных целевой функции. В окрестностях точки локального минимума частные производные малы, и поэтому проходится делать большее число шагов для попадания в окрестность экстремума. Ускорить этот процесс помогает использование вторых производных функции, которые в окрестности минимума больше нуля. Градиентные методы обладают линейной скоростью сходимости, в то время как метод Ньютона обладает квадратичной скоростью сходимости. Тем не менее, применение метода Ньютона оказывается очень эффективным при условии, что выполняются необходимые и достаточные условия его сходимости (дважды дифференцируемость целевой функции и положительная определённость обратного гессиана). Однако само исследование необходимых и достаточных условий сходимости метода в некоторых случаях может быть достаточно сложной задачей.

### Отчет

В практической части представлены реализации метода Ньютона и BFGS. Проверено, что на квадратичной функции метод Ньютона сходится за один шаг. Для сравнительных тестов использована функция Розенброка, она хороша в особенности для сравнения методов второго порядка с методами более низкого порядка, поскольку в параболической окрестности её минимума градиенты близки к нулю и после попадания в эту "низину"градиентным методам очень тяжело дойти до точки минимума, так как их шаги устремляются к нулю, а методы второго порядка не затухают благодаря гессиану. Протестировав на ней три алгоритма (Нелдера-Мида, градиентный спуск и Ньютона), замечаем, что градиентный метод не дошел до минимума с нужной точностью за разумное количество итераций, в то время как методы Недлера-Мида и Ньютона хорошо сошлись, поскольку первый не зависит от градиента, а второй помимо градиента использует гессиан. Кроме того заметим, что метод Ньютона пришел точно в нужную точку, сделав достаточно большой шаг. Это говорит о том, что использование производных второго порядка, позволяет лучше выбирать направление и длину шага.

Перейдем теперь к BFGS. Сравнив его поведение с работой метода Ньютона, видим, что он требует значительно больше итераций в силу неточности оценки гессиана. Но сравнивая методы на многомерной функции Розенброка с достаточно большим числом измерений (100), получаем значительное ускорение по времени приблизительно в десять раз.

# 3 Список использованной литературы


- Stephen Boyd, Lieven Vandenberghe, Convex Optimization
  `https://web.stanford.edu/~boyd/cvxbook/bv_cvxbook.pdf`
- Лемешко Б. Ю., Методы оптимизации
  `https://ami.nstu.ru/~headrd/seminar/publik_html/MO_conspect.pdf`
- Кочегурова Е. А. , Теория и методы оптимизации
  `https://portal.tpu.ru/SHARED/k/KOCHEG/study/Tab1/Lection_MO_2018.pdf`

- Кононюк А. Е., Основы теории оптимизации
  http://ecat.diit.edu.ua/ft/Optimization2_1.pdf
- https://fmin.xyz/
- https://habr.com/ru/post/561128/
- https://habr.com/ru/post/333356/