

# Метод BFGS

## (Broyden Fletcher Goldfarb Shanno)

### Оглавление

Введение.....	2
Условия Вольфе.....	3
Квазиньютоновские методы.....	4
Метод BFGS .....	6

Авторы: Михнев Денис и Грошева Екатерина

## Введение

Метод BFGS является итерационным методом численной оптимизации и относится к классу квазиньютоновских методов. Буквы в его названия складываются из первых букв фамилий исследователей, что работали над ним, а именно: Broyden Fletcher Goldfarb Shanno. Квазиньютоновскими методами называют те, где в отличие от ньютоновских напрямую не вычисляется гессиан функции. Вместо вычисления частных производных второго порядка, гессиан вычисляется из ранее сделанных шагов.

Существует несколько модификаций метода:

**L-BFGS** (ограниченное использование памяти) — используется в случае большого количества неизвестных.

**L-BFGS-B** — модификация с ограниченным использованием памяти в многомерном кубе.

Метод эффективен и устойчив, поэтому зачастую применяется в функциях оптимизации. Например, в SciPy, популярной библиотеки для языка python, в функции optimize по умолчанию применяется BFGS, L-BFGS-B.

Каждая итерация может быть совершена со стоимостью  $O(n^2)$  ( плюс стоимость вычисления функции и оценки градиента). Помимо главного преимущества, связанного с отсутствием вычисления производных второго порядка, здесь также нет таких операций, как решение линейных систем или сложных математических операций. Нет операций порядка  $O(n^3)$ . Данный алгоритм устойчив и имеет сверхлинейную сходимость, чего достаточно для большинства практических задач. Несмотря на то, что методы Ньютона сходятся гораздо быстрее (квадратично), стоимость каждой итерации выше, поскольку необходимо решать линейные системы.

Формула BFGS имеет также имеет самокорректирующиеся свойства. Если матрица не смогла верно оценить кривизну функции и если эта плохая оценка замедляет алгоритм, то аппроксимация гессиана стремится исправить ситуацию за несколько шагов. Самокорректирующие свойства алгоритма работают только в том случае, если реализован соответствующий линейный поиск (соблюдены условия Вольфе).

## Условия Вольфе

(Впервые опубликованы Филипом Вольфе в 1969 году)

Пусть решается задача оптимизации  $\min_x f(x)$  и уже имеется приближение решения задачи  $x_k$  и пусть каким-либо методом мы нашли направление  $p_k$ , в котором будем искать новое приближение решения  $x_{k+1}$ . Тогда  $x_{k+1} = x_k + a_k p_k$ , где  $a_k$  удовлетворяет условиям Вольфе:

$$\begin{aligned} f(x_k + a_k p_k) &\leq f(x_k) + c_1 a_k \nabla f_k^T p_k, \\ \nabla f(x_k + a_k p_k)^T p_k &\geq c_2 \nabla f_k^T p_k \end{aligned}$$

Константы выбираются следующим образом:  $0 < c_1 < c_2 < 1$ . Обычно константа  $c_1$  выбирается достаточно маленькой (в окрестности 0), что означает, что функция после совершения шага должна уменьшиться, в то время как  $c_2$  выбирается значительно большей (в окрестности 1), что, в свою очередь, означает, что проекция градиента в новом приближении должна либо изменить направление, либо уменьшиться.

### Усиленные условия Вольфе

$$\begin{aligned} f(x_k + a_k p_k) &\leq f(x_k) + c_1 a_k \nabla f_k^T p_k, \\ |\nabla f(x_k + a_k p_k)^T p_k| &\leq c_2 |\nabla f_k^T p_k| \end{aligned}$$

Такой вариант предполагает, что новое приближение лежит в окрестности локального минимума функции:  $\phi(a) = f(x_k + a_k p_k)$ . Второе неравенство изменено таким образом, чтобы проекция градиента должна уменьшиться по модулю. Таким образом исключаются точки, которые находятся далеко от стационарных точек функции  $\phi$ . Константы подбираются так же, как и в условиях Вольфе.

### Свойства Алгоритма

Можно показать, что если  $p_k$  - направление убывания ограниченной снизу и непрерывно дифференцируемой функции  $f$ , каждый шаг  $a_k$  удовлетворяет условиям Вольфе, а градиент функции  $f$  непрерывен по Липшицу:

□  $x, \tilde{x} \in N \quad \|\nabla f(x) - \nabla f(\tilde{x})\| \leq L \|x - \tilde{x}\|$ , то

$$\sum_{k \geq 0} \cos^2 \theta_k \|\nabla f_k\|^2 < \infty, \text{ где } \cos^2 \theta_k = - \frac{\nabla f_k^T p_k}{\|\nabla f_k\| \|p_k\|}.$$

Отсюда следует, что  $\cos^2 \theta_k \|f_k\|^2 \rightarrow 0$  при  $k \rightarrow \infty$ , что означает, что алгоритм сходится.

## Квазиньютоновские Методы

Методы оптимизации такого типа основаны на накоплении информации о кривизне целевой функции по наблюдениям за изменением градиента, чем принципиально отличаются от ньютоновских методов. Класс квазиньютоновских методов исключает явное формирование матрицы Гессе, заменяя её некоторым приближением.

### Квазиньютоновское условие

Разложим градиент  $f(x_k)$  исходной функции в ряд Тейлора в окрестности точки очередного приближения  $x_k$  по степеням следующего шага алгоритма  $s_k$ :  $f(x_k + s_k) \approx f(x_k) + F(x_k)s_k$ . Тогда оценка матрицы Гессе  $H_{k+1}$  должна удовлетворять равенству:  $H_{k+1} s_k = y_k$ , где  $y_k = f(x_k + s_k) - f(x_k)$ .

### Начальное приближение и направление поиска

На каждой итерации с помощью  $H_k$  определяется следующее направление поиска  $p_k$ , и матрица  $H$  обновляется с учётом вновь полученной информации о кривизне:  $H_k p_k = -f(x_k)$ ,  $H_{k+1} = H_k + U_k$ , где  $U_k$  – матрица, характеризующая поправку, вносимую на очередном шаге.

В качестве начального приближения  $H_0$  кладут единичную матрицу. Первое направление  $p_0$  будет в точности совпадать с направлением наискорейшего спуска.

### Поправка единичного ранга

Один шаг алгоритма даёт информацию о кривизне вдоль одного направления, поэтому ранг матрицы  $U_k$  полагают малым, и даже единичным:  $H_{k+1} = H_k + UV^T$ , где  $U$  и  $V$  некоторые вектора. Тогда квазиньютоновское условие примет вид:  $(H_k + UV^T)s_k = y_k$ ,  $U(V^T s_k) = y_k - H_k s_k$ .

Полагая, что предыдущая матрица на очередном шаге квазиньютоновскому условию не удовлетворяет и что вектор  $U$  не ортогонален  $s_k$ , получают выражение для  $U$  и  $H_{k+1}$ :  $U = \frac{1}{U^T s_k} (y_k - H_k s_k) U^T$ .

Из соображений симметричности матрицы Гессе, вектор  $U$  берут коллинеарным  $U$ :  $H_{k+1} = H_k + \frac{1}{(y_k - H_k s_k)^T s_k} (y_k - H_k s_k)(y_k - H_k s_k)^T$ .

Полученное уравнение называется симметричной формулой ранга один.

### Поправки ранга два

Один из способов конструирования поправок ранга два заключается в построении сходящейся последовательности матриц  $H^{(j)}$ . В качестве начального значения  $H^{(0)}$  берут  $H_k$ ,  $H^{(1)}$  вычисляют по формуле:  $H^{(1)} = H^{(0)} + \frac{1}{U^T s_k} (y_k - H^{(0)} s_k) U^T$ . После чего её симметризируют:  $H^{(2)} = \frac{H^{(1)} + H^{(1)T}}{2}$ .

Однако, полученная матрица больше не удовлетворяет квазиньютоновскому условию. Чтобы это исправить, процедуру повторяют. В результате на  $j$  – м шаге:  $H^{(2j+1)} = H^{(2j)} + \frac{1}{U^T s_k} (y_k - H^{(2j)} s_k) U^T$ ,  
 $H^{(2j+2)} = \frac{H^{(2j+1)} + H^{(2j+1)T}}{2}$ .

Предел этой последовательности равен:  $H_{k+1} = H_k + \frac{1}{U^T s_k} [(y_k - H_k s_k) U^T + U (y_k - H_k s_k)^T] - \frac{(y_k - H_k s_k)^T s_k}{(U^T s_k)^2} U U^T$ .

При выборе различных  $U$  (не ортогональных  $s_k$ ) получают различные формулы пересчета матрицы  $H$ :

- 1)  $U = y_k - H_k s_k$  приводит к симметричной формуле ранга один;
- 2)  $U = s_k$  приводит к симметричной формуле Пауэлла-Бройдена (PSB)
- 3)  $U = y_k$  приводит к симметричной формуле Девидона – Флетчера – Пауэлла (DFP):  $H_{k+1} = H_k - \frac{1}{s_k^T H_k s_k} H_k s_k s_k^T H_k + \frac{1}{y_k^T s_k} y_k y_k^T + (s_k^T H_k s_k) w_k w_k^T$ ,  
 где  $w_k = \frac{1}{y_k^T s_k} y_k - \frac{1}{s_k^T H_k s_k} H_k s_k$ .

Нетрудно проверить, что  $w_k$  ортогонален  $s_k$ . Таким образом, добавление слагаемого  $w_k w_k^T$  не нарушит ни квазиньютоновского условия, ни условия симметричности. Поэтому проводился ряд теоретических исследований, подвергавших последнее слагаемое масштабированию на предмет получения наилучшего приближения. В результате была принята точка зрения, что наилучшим вариантом является отвечающий полному отсутствию последнего слагаемого. Этот вариант пересчёта известен под именем формулы Бройдена — Флетчера — Гольдфарба — Шанно (BFGS):  $H_{k+1} = H_k - \frac{1}{s_k^T H_k s_k} H_k s_k s_k^T H_k + \frac{1}{y_k^T s_k} y_k y_k^T$ .

# Метод BFGS

Рассмотрим алгоритм данного метода.

Пусть задана некоторая функция  $f(x, y)$ . Решается задача оптимизации:  
 $\min f(x, y)$ .

Данная функция:

- является не выпуклой функцией
- имеет непрерывные вторые производные

## Шаг 1

Инициализируем начальную точку  $x_0$ ;

Задаем точно поиска  $\varepsilon > 0$ ;

Определяем начальное приближение  $H_0 = B_0^{-1}$ , где  $B_0^{-1}$  – гессиан функции.

*(Гессиан функции - симметрическая квадратичная форма, описывающая поведение функции во втором порядке. Для функции  $f$ , дважды дифференцируемой в точке  $x \in R^n$   $H(x) = \sum_{i=1}^n \sum_{j=1}^n a_{ij} x_i x_j$ , где  $a_{ij} = \frac{\partial^2 f}{\partial x_i \partial x_j}$  и функция  $f$  задана на  $n$ -мерном вещественном пространстве  $R^n$  с координатами  $x_1, \dots, x_n$ . Гессиан – квадратичная форма, заданная на касательном пространстве, не меняющаяся при линейных преобразованиях переменных. Гессианом также часто называют и определитель матрицы  $(a_{ij})$ .)*

Общей формулы для выбора начального приближения  $H_0$ , которая хорошо работала бы во всех случаях, не существует. В качестве начального приближения можно взять гессиан функции, вычисленный в начальной точке  $x_0$ , либо хорошо обусловленную, невырожденную матрицу. На практике чаще всего используют единичную.

## Шаг 2

Находим точку, в направлении которой будем производить поиск. Определим ее следующим образом:  $p_k = -H_k * \nabla f_k$ .

## Шаг 3

Вычислим через рекуррентное соотношение следующую точку:

$x_{k+1} = x_k + a_k * p_k$ , где коэффициент  $a_k$  находим, используя линейный поиск. Данный коэффициент удовлетворяет условиям Вольфе. В большинстве реализаций константы выбираются такие:  $c_1 = 0.0001$  и  $c_2 = 0.9$ . Фактически мы находим такое  $a_k$ , при котором значение функции  $f(x_k + a_k * p_k)$  минимально.

## Шаг 4

Определяем вектора:

$$s_k = x_{k+1} - x_k$$

$$y_k = \nabla f_{k+1} - \nabla f_k,$$

где  $s_k$  – шаг алгоритма на итерации,  $y_k$  – изменение градиента на итерации.

### Шаг 5

Обновляем гессиан функции, согласно следующей формуле:

$$H_{k+1} = (I - p_k * s_k * y_k^T) H_k (I - p_k * y_k * s_k^T) + p * s_k * s_k^T, \text{ где } p_k = \frac{1}{y_k^T s_k}$$

и  $I$  – единичная матрица.

Замечание: выражение вида  $y_k * s_k^T$  является внешним произведением двух векторов. Пусть определены два вектора  $U$  и  $V$ , тогда их внешнее произведение эквивалентно матричному произведению  $UV^T$ :

$$UV^T = \begin{pmatrix} U_1 \\ U_2 \end{pmatrix} (V_1 \quad V_2) = \begin{pmatrix} U_1 V_1 & U_1 V_2 \\ U_2 V_1 & U_2 V_2 \end{pmatrix}$$

### Шаг 6

Алгоритм продолжает выполняться до тех пор, пока истинно неравенство:  
 $|\nabla f_k| > \varepsilon$