Метод Ньютона и квазиньютоновские методы

Математические методы распознавания образов, весна 2024. Конспект составил Алексеев Илья.

Сравнительная таблица методов, которые изучим на этом занятии (d — число весов модели):

	Число операций	Память	Скорость сходимости
GD	$O(d)$	$O(d)$	линейная
Newton	$O(d^3)$	$O(d^2)$	квадратичная
Newton-CG	$O(dn)$	$O(d)$	сверхлинейная
SR1	$O(d^2)$	$O(d^2)$	сверхлинейная
BFGS	$O(d^2)$	$O(d^2)$	сверхлинейная
L-BFGS	$O(dm)$	$O(dm)$	линейная

Примечание:

- В эту таблицу не включены слагаемые и множители, возникающие из-за вызова оракула
- n — число итераций в методе сопряжённых градиентов
- m — размер очереди
- в колонке "Скорость сходимости" указана гарантированная доказанная скорость, на практике L-BFGS может показывать сверхлинейную сходимость

Метод Ньютона

По-прежнему решаем безусловную задачу

$$\min_x f(x),$$

где f — дважды непрерывно дифференцируема и $f''(x) \succ 0$.

Разложим $f(x)$ в ряд Тейлора до второго слагаемого и получим квадратичную аппроксимацию:

$$f(x) \approx f(x_0) + f'(x_0)^T(x - x_0) + \frac{1}{2}(x - x_0)^T f''(x_0)(x - x_0).$$

Найдём минимум такой функции:

$$f'(x_0) + f''(x_0)(x - x_0) = 0$$
$$x - x_0 = -[f''(x_0)]^{-1} f'(x_0) = \Delta x_{\text{nt}}$$

Будем использовать **шаг Ньютона** Δx_{nt} в качестве шага в спуске:

$$x_{k+1} = x_k - f''(x_k)^{-1} f'(x_k).$$

Чтобы не обращать гессиан, решаем линейную систему $f''(x_k)h_k = -f'(x_k)$ с использованием матричных разложений.

Геометрический смысл гессиана

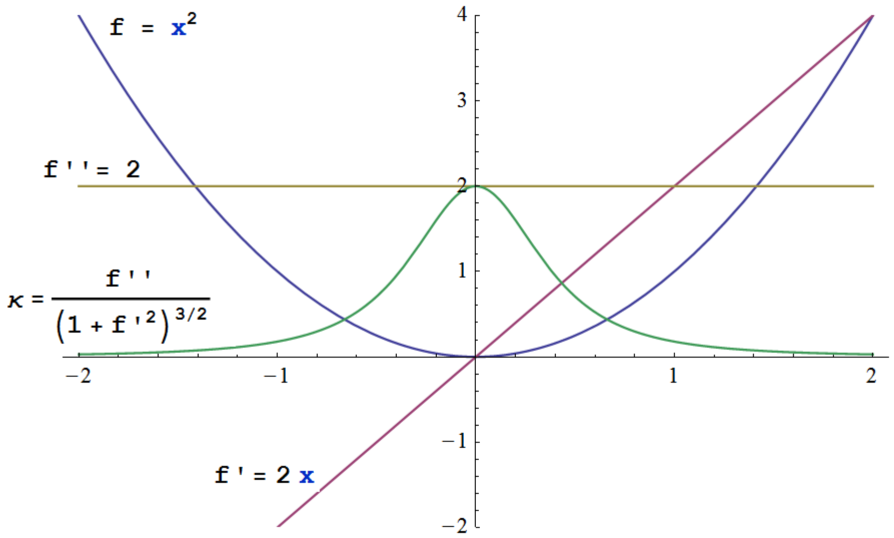
Гессиан имеет прямую связь с выпуклостью и кривизной функции в данной точке:

- Положительная определённость гессиана означает выпуклость функции
- С помощью гессиана можно вычислить коэффициент кривизны κ функции в любой точке

Например, для двумерной функции:

$$\kappa = \frac{\det f''}{(1 + (f_x)^2 + (f_y)^2)^2}$$

Пример для одномерной функции:



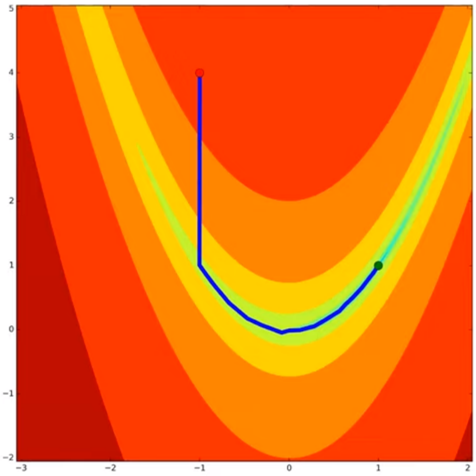
В шаге Ньютона гессиан позволяет совершить шаг, оптимальный с точки зрения квадратичной аппроксимации.

Сравнение с градиентным спуском

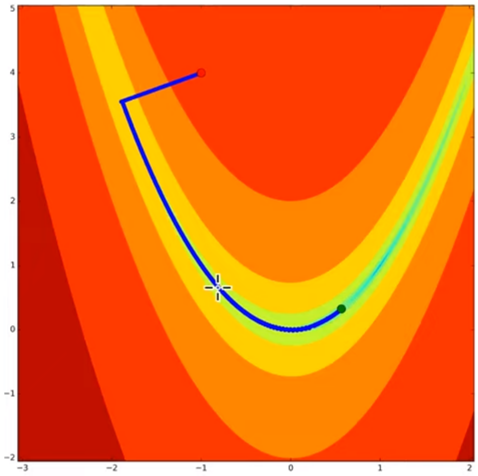
Что если вместо вычисления гессиана взять скалярную матрицу? Это будет эквивалентно предположению, что оптимизируемый функционал имеет одинаковую кривизну во всех направлениях. При этом шаг обновления превратится в шаг градиентного спуска:

$$f''(x) := \alpha I \Rightarrow x_{k+1} = x_k - \frac{1}{\alpha} f'(x_k).$$

Это одно из объяснений, почему градиентный спуск медленно сходится для плохо шкалированных функций.



Newton: 30 итераций



GD: 1000 итераций

Демпфированный метод Ньютона

Локальная сходимость метода Ньютона

Говорят, что присутствует **локальная сходимость**, если сходимость зависит от x_0. До сих пор мы имели дело с градиентными методами, которые сходились из любой точки. Но метод Ньютона может расходиться или осциллировать в зависимости от x_0.

Достаточно рассмотреть пример:

$$\varphi(t) = \sqrt{1+t^2}.$$

Минимум этой функции достигается в точке $t = 0$. Запишем метод Ньютона для этой функции:

$$\varphi'(t) = \frac{t}{\sqrt{1+t^2}}, \ \varphi''(t) = \frac{1}{(1+t^2)^{3/2}} \Rightarrow$$
$$t_{k+1} = t_k - \frac{\varphi'(t_k)}{\varphi''(t_k)} = t_k - t_k(1+t_k^2) = -t_k^3.$$

Видим, что метод сходится только в области $|t_0| < 1$.

Локальная сверхлинейная сходимость метода Ньютона

Допустим, в окрестности x_k находится искомый x^*. Тогда справедлива цепочка равенств:

$$0 = f'(x^*) = f'(x_k) + f''(x_k)(x^* - x_k) + o(\|x^* - x_k\|)$$
$$\Rightarrow -f'(x_k) = f''(x_k)(x^* - x_k) + o(\|x^* - x_k\|).$$

Множим обе части на обратный гессиан:

$$-[f''(x_k)]^{-1} f'(x_k) = x^* - x_k + o(\|x^* - x_k\|)$$
$$x_k - x^* - f''(x_k)^{-1} f'(x_k) = o(\|x^* - x_k\|).$$

Поскольку $x_{k+1} - x_k = -f''(x_k)^{-1} f'(x_k)$, то

$$x_{k+1} - x^* = o(\|x^* - x_k\|).$$

Получили определение сверхлинейной сходимости. В ходе более скрупулёзных умозаключений можно выяснить, что метод Ньютона имеет **квадратичную скорость сходимости**.

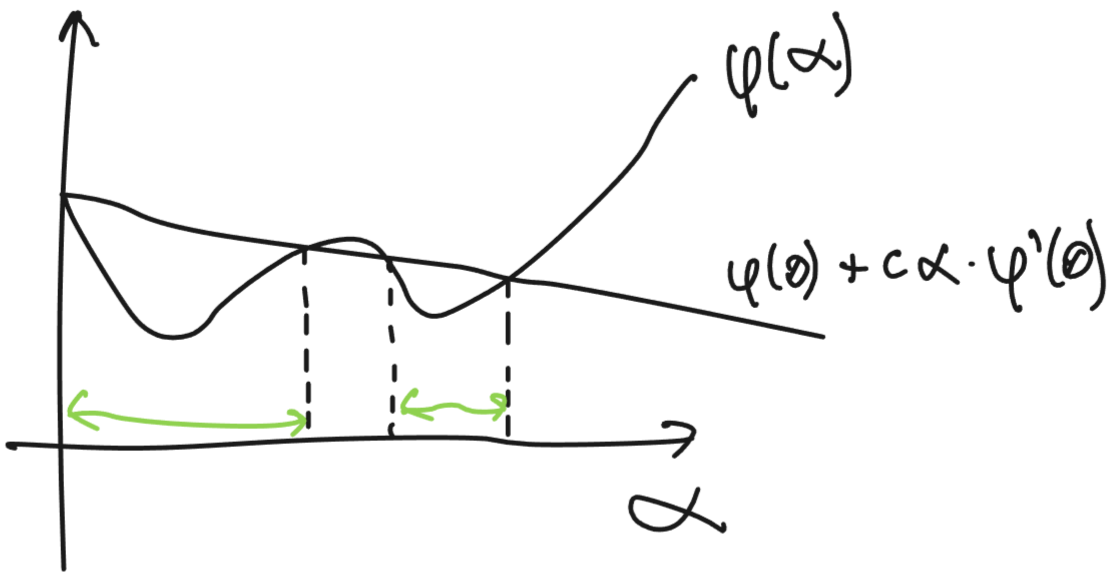
Неточная одномерная оптимизация

Небольшое погружение в одномерную оптимизацию. Пусть $\varphi(\alpha)$ — одномерная функция, $\varphi'(0) < 0$. Решаем задачу неточной оптимизации $\min_\alpha \varphi(\alpha)$.

Скажем, что размер шага α удовлетворяет **правилу Армихо**, если

$$\varphi(\alpha) \leqslant \varphi(0) + c\alpha \cdot \varphi'(0),$$

где $c \in (0, 1)$. Такое правило задает области, в которых $\varphi(\alpha)$ гарантированно меньше, чем $\varphi(0)$.



Демпфированный метод Ньютона

Пусть теперь $\varphi(\alpha) = f(x + \alpha h)$ — значение оптимизируемого функционала вдоль направления h. Тогда метод Ньютона можно разбить на две фазы:

- На первой фазе x_k далеко от оптимума настолько, что метод находится вне области сверхлинейной сходимости. В этот момент можно искать шаги $\alpha < 1$ по правилу Армихо.
- Во второй фазе x_k близок к оптимуму настолько, что метод находится в области сверхлинейной сходимости. Тут можно брать $\alpha = 1$.

Чтобы найти α, удовлетворяющее правилу Армихо, можно использовать процедуру **бэктрекинга**:

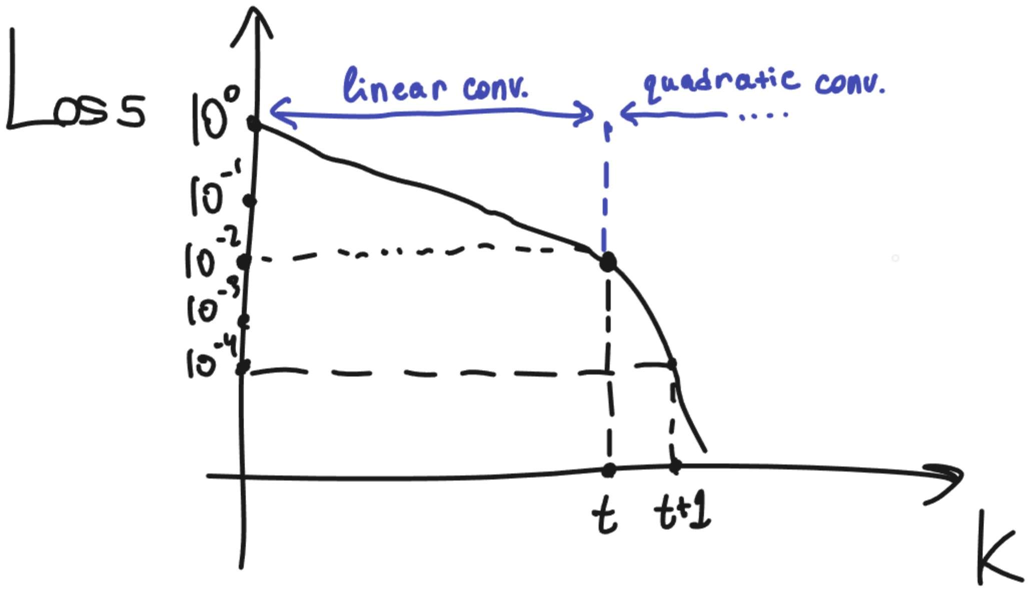
1. Инициализируем $\alpha_0 > 0,\ \rho \in (0, 1)$
2. Пока не выполнено правило Армихо:
    1. Обновляем $\alpha_{k+1} = \rho\alpha_k$
    2. (Если $\alpha_{k+1} \leqslant \varepsilon$, то поиск не удался)

Алгоритм **демпфированного метода Ньютона**:

1. Инициализируем $x_0 \in \mathbb{R}^n$
2. Пока не выполнен критерий останова:
    1. Вычислить $h_k = f''(x_k)^{-1} f'(x_k)$
    2. Бэктрекинг по α_k и правилу Армихо
    3. Обновление $x_{k+1} = x_k - \alpha_k f''(x_k)^{-1} f'(x_k)$

Для метода Ньютона всегда берем $\alpha_0 = 1$. Тогда демпфированный метод Ньютона обретает глобальную сходимость, потому что если x_0 не принадлежит области локальной сходимости, то благодаря правилу Армихо на первых итерациях метод не расходится, а медленно движется к области сходимости.

Кривая сходимости зачастую имеет следующий вид: она состоит из линейного или сублинейного участка в начале и сверхлинейного или квадратичного участка в конце.



Безгессианный метод Ньютона

Из-за медленной итерации и больших накладок по памяти метод Ньютона не используют. Что если вычислять шаг Ньютона $f''(x_k)^{-1}f'(x_k)$ приближённо. Если применить ряд трюков, то можно найти шаг Ньютона с достаточной точностью, проведя существенно меньшее количество вычислений.

Умножение "гессиан–вектор"

Первый трюк состоит в том, чтобы искать произведение гессиана на вектор без вычисления и хранения самого гессиана. Рассмотрим пример логистической регрессии:

$$f(w) = \sum_{i=1}^{\ell} \mathcal{L}(x_i, y_i, w), \quad \mathcal{L}(x, y, w) = -\log \sigma(yw^T x), \quad x_i, w \in \mathbb{R}^d.$$

Гессиан оптимизируемого функционала записывается так:

$$f''(w) = X^T B X, \quad B = \mathrm{diag}(\sigma \circ (1 - \sigma)), \quad \sigma = (\sigma_1, \ldots, \sigma_\ell)^T \in \mathbb{R}^\ell, \quad \sigma_i = \sigma(y_i w^T x_i).$$

Тогда для произвольного $h \in \mathbb{R}^d$ произведение $f''(w)h$ можно вычислить за $O(\ell d)$ по следующей схеме:

1. $z_1 := Xh$, стоимость $O(\ell d)$
2. $z_2 := \sigma \circ (1 - \sigma) \circ z_1$, стоимость $O(\ell)$
3. $f''(w)h := X^T z_2$, стоимость $O(\ell d)$

Неточное решение СЛАУ

Решение СЛАУ эффективно выполнять с помощью матричных разложений. Но такие методы требуют явно хранить матрицу в памяти. Существуют методы, которые не требуют хранения матрицы в память, а обращаются к ней только посредством умножения на вектор. Если применить для этой цели метод сопряжённых градиентов, то получится безгессианный метод, который в scipy называют `Newton-CG`.

Квазиньютоновские методы

Концептуальная схема

Идея безгессианных методов Ньютона в том, чтобы приближённо искать шаг спуска посредством неточного решения СЛАУ с гессианом. Идея квазиньютоновских методов другая: что если на каждой итерации приближать сам гессиан $B_k \approx f''(x_k)$ или обратный гессиан $H_k \approx f''(x_k)^{-1}$, не используя при этом оракул второго порядка. Речь о следующей концептуальной схеме (пример для H_k):

1. Инициализировать H_0 (например $H_0 := I$)
2. Сделать шаг спуска $x_1 := x_0 - f'(x_0)$
3. Вычислить $H_1 := \mathrm{Update}(H_0, f'(x_1))$, сложность $O(n^2)$
4. Сделать шаг спуска $x_2 := x_1 - H_1 f'(x_1)$, сложность $O(n^2)$
5. Вычислить $H_2 := \mathrm{Update}(H_1, f'(x_2))$
6. И так далее

Secant Equation

Пока будем рассуждать в терминах $B_k = H_k^{-1}$ и на время забудем про H_k. Итак, мы используем квадратичную модель как в методе Ньютона:

$$x_{k+1} = \arg\min_h m(h), \quad m(h) = f(x_k) + f'(x_k)^T h + \frac{1}{2} h^T B_k h.$$

Во всех квазиньютоновских методах на B_k накладывается **условие секущей**. Оно заключается в выполнении следующего:

1. Градиент аппроксимирующей модели должен совпадать с градиентом исходной функции в текущей точке: $m'(0) = f'(x_k)$.
2. Градиенты в предыдущей точке тоже должны совпадать: $m'(x_{k-1} - x_k) = f'(x_{k-1})$.

В одномерном случае эти два условия эквивалентны тому, что мы аппроксимируем производную $f'(x)$ прямой, проходящей через x_0, x_1.

$$f'(x_k) + B_k(x_{k-1} - x_k) = f'(x_{k-1}) \iff B_k(x_{k-1} - x_k) = f'(x_{k-1}) - f'(x_k).$$

Обозначим $s_{k-1} = x_{k-1} - x_k$ и $y_{k-1} = f'(x_{k-1}) - f'(x_k)$. Тогда условие секущей эквивалентно $B_k s_{k-1} = y_{k-1}$.

В этой системе n уравнений и $n(n+1)/2$ неизвестных — т.е. решений целое множество. Однако решение не всегда существует. Например, если $s_0^T y_0 < 0$, то $s_0^T B_1 s_0 < 0$, т.е. нарушено $B_1 \succ 0$. Значит, $s_0^T y_0 \geqslant 0$ — ОДЗ нашей задачи (curvature condition).

Symmetric Rank-one (SR1)

Идея метода SR1 в том, чтобы использовать обновление вида $B_{k+1} := B_k + \alpha_k v_k v_k^T$. Это одноранговое обновление, сохраняющее свойство симметричности. Если подставить это правило обновления в уравнение секущей, то можно вывести такую формулу:

$$B_{k+1} := B_k + \frac{(y_k - B_k s_k)(y_k - B_k s_k)^T}{(y_k - B_k s_k)^T s_k}.$$

По тождеству Вудбери:

$$H_{k+1} := H_k + \frac{(s_k - H_k y_k)(s_k - H_k y_k)^T}{(s_k - H_k y_k)^T y_k}.$$

С помощью этой формулы нужно явно посчитать и поместить в память матрицу H_{k+1}. Вычисление шага Ньютона можно произвести по следующей схеме:

1. $z_1 := (s_k - H_k y_k)^T f'(x_{k+1})$, стоимость $O(n)$
2. $z_2 := (s_k - H_k y_k) z_1$, стоимость $O(n)$
3. $z_3 := z_1 / (s_k - H_k y_k)^T y_k$, стоимость $O(n)$
4. $H_{k+1} f'(x_{k+1}) := H_k f'(x_{k+1}) + z_3$, стоимость $O(n^2)$

BFGS

Идея в том, чтобы искать формулу обновления в виде оптимизационной задачи для H_{k+1}:

$$\begin{aligned} \min_H \ & \|W^{-1}(H - H_k)W^{-T}\|_F^2, \\ \text{s.t. } & H y_k = s_k, \\ & H \succ 0. \end{aligned}$$

Где $W \in \mathbb{R}^{n \times n}$ — невырожденная и $WW^T y_k = s_k$ (здесь используется так называемая взвешенная норма Фробениуса). Она имеет аналитическое решение:

$$H_{k+1} = (I - \rho_k s_k y_k^T) H_k (I - \rho_k y_k s_k^T) + \rho_k s_k s_k^T.$$

Подсчет шага Ньютона по этой формуле тоже занимает $O(n^2)$. Немного более подробный вывод BFGS можно посмотреть здесь.

L-BFGS

Алгоритм называют Limited Memory BFGS, потому что он не хранит гессиан, а хранит две очереди из m последних векторов: $y_{k-1}, \dots, y_{k-m}$ и $s_{k-1}, \dots, s_{k-m}$. Обычно берут $m = 10$. С помощью этой очереди можно выполнить приближённый шаг обновления BFGS. Сложность по памяти и времени $O(nm)$.