

## 1 Линейный МНК

Задача линейного МНК (метода наименьших квадратов) в терминах линейной алгебры записывается следующим образом: требуется найти вектор параметров  $\mathbf{x}_*$ , минимизирующий функцию  $\Phi(\mathbf{x})$  для данной матрицы  $A$  и вектора  $\mathbf{y}$ :

$$\Phi(\mathbf{x}) = \frac{1}{2} \|\mathbf{y} - A\mathbf{x}\|^2,$$

где  $\|\dots\|$  обозначает норму, а двойка в знаменателе введена для удобства вычислений, как будет видно вскоре она пропадает при дифференцировании этого выражения.

В случае физического эксперимента,  $\mathbf{y}$  — это результаты измерений, а  $A\mathbf{x}$  — это предсказание линейной модели для наших измерений. Обратите внимание, что здесь  $\mathbf{x}$  — это неизвестные параметры модели, которые мы хотим найти из эксперимента. С другой стороны матрица  $A$  известна и определяется параметрами эксперимента.

Приравняв производную от  $\Phi(x)$  по компонентам  $\mathbf{x}$  к нулю мы получим систему линейных уравнений, решение которой даёт искомые значения  $\mathbf{x}$ :

$$0 = \frac{\partial \Phi(x)}{\partial x_i} = A_{ji} A_{jk} x_k - A_{ji} y_j. \quad (1)$$

Решение этой системы:

$$\mathbf{x}_* = (A^T A)^{-1} A^T \mathbf{y}. \quad (2)$$

## 2 Нелинейный МНК

Задача нелинейного МНК ставится похожим образом, но в этот раз линейная комбинация  $A\mathbf{x}$  заменяется на векторную функцию  $\mathbf{f}(\mathbf{x})$ :

$$\Phi(\mathbf{x}) = \frac{1}{2} \|\mathbf{y} - \mathbf{f}(\mathbf{x})\|^2 \quad (3)$$

Снова найдём производную этого выражения:

$$\frac{\partial \Phi(x)}{\partial x_i} = \frac{\partial f_j}{\partial x_i} (f_j - y_j), \quad (4)$$

$$\nabla \Phi(x) = J^T (\mathbf{f}(\mathbf{x}) - \mathbf{y}), \quad (5)$$

где якобиан  $J_{ij}(\mathbf{x}) = \partial f_i / \partial x_j$  — известные значения производной функции  $\mathbf{f}$  в точке  $\mathbf{x}$ .

## 3 Градиентный спуск

Можно предположить, что направление, противоположное градиенту  $\nabla \Phi(\mathbf{x})$ , задаёт направление, в котором стоит искать минимум функции  $\Phi(\mathbf{x})$ . Это

даёт нам следующий способ поиска  $\mathbf{x}_*$ . Пусть начальное приближение решения — это  $\mathbf{x}^{(0)}$ , тогда каждое следующее приближение  $\mathbf{x}^{(k)}$  может быть получено следующим образом:

$$\mathbf{x}^{(k)} = \mathbf{x}^{(k-1)} - \frac{\nabla \Phi(\mathbf{x}^{(k-1)})}{\|\nabla \Phi(\mathbf{x}^{(k-1)})\|} \times s^{(k)}, \quad (6)$$

где  $s^{(k)}$  — размер шага, его выбор может зависеть как от  $k$  (обычно шаг уменьшают в процессе итераций), так и от длины  $\|\nabla \Phi(\mathbf{x}^{(k-1)})\|$ .

## 4 Метод Гаусса—Ньютона

В некотором смысле, скалярное произведение  $(\nabla \Phi(\mathbf{x}), \mathbf{x})$  со значением градиента  $\nabla \Phi(\mathbf{x})$ , даваемым уравнением (5), является линейной аппроксимацией функции  $\Phi(\mathbf{x})$  в точке  $\mathbf{x}$ . У такой линейной функции минимум не определён, поэтому распространены методы оптимизации, в которых на каждом итерационном шаге  $k$  функция  $\Phi(\mathbf{x}^{(k-1)})$  аппроксимируется квадратичной функцией, у которой можно найти минимум и использовать его в качестве следующего приближения решения  $\mathbf{x}^{(k)}$ . Построим один из возможных вариантов такой квадратичной аппроксимации.

Разложим функцию  $\mathbf{f}(\mathbf{x}^k)$  в ряд Тейлора вокруг точки  $\mathbf{x}^{(k-1)}$ :

$$\mathbf{f}(\mathbf{x}^{(k)}) = \mathbf{f}(\mathbf{x}^{(k-1)}) - J(\mathbf{x}^{(k-1)})(\mathbf{x}^{(k)} - \mathbf{x}^{(k-1)}) + o(\|\mathbf{x}^{(k)} - \mathbf{x}^{(k-1)}\|). \quad (7)$$

Введём обозначение  $\Delta \mathbf{x} \equiv \mathbf{x}^{(k)} - \mathbf{x}^{(k-1)}$  и подставим полученное выражение в определение  $\Phi(\mathbf{x})$  (3), пренебрегая  $o(\|\Delta \mathbf{x}\|)$ :

$$\begin{aligned} \Phi(\mathbf{x}^{(k)}) &\simeq \frac{1}{2} \|\mathbf{y} - \mathbf{f}(\mathbf{x}^{(k-1)}) + J(\mathbf{x}^{(k-1)})\Delta \mathbf{x}\|^2 \\ &= \frac{1}{2} \|\mathbf{y} - \mathbf{f}\|^2 + (J^T(\mathbf{y} - \mathbf{f}), \Delta \mathbf{x}) + \frac{1}{2} (J^T J \Delta \mathbf{x}, \Delta \mathbf{x}). \end{aligned} \quad (8)$$

Перед нами квадратичная форма относительно вектора  $\Delta \mathbf{x}$ . Так как  $J^T J$  положительно определена (докажите сами), то эта квадратичная форма имеет минимум. Этот минимум может быть найден по следующей формуле:

$$J^T J \Delta \mathbf{x}_* = J^T \mathbf{y}. \quad (9)$$

На практике, значение  $\Delta \mathbf{x}_*$  часто умножают на константу  $K : 0 < K \leq 1$ :

$$\mathbf{x}^{(k)} = \mathbf{x}^{(k-1)} + K \Delta \mathbf{x}_*. \quad (10)$$

Обратим внимание, что формула (8) похожа на разложение  $\Phi(\mathbf{x})$  в ряд Тейлора до второго порядка. Однако, в ряд Тейлора входит дополнительное слагаемое, пропорциональное вторым производным функции  $\mathbf{f}(\mathbf{x})$ . Если функция  $\mathbf{f}(\mathbf{x})$  сильно нелинейна, то при полном разложении в ряд Тейлора  $\Phi(\mathbf{x})$  до второго порядка малости может быть нарушено условие положительной определенности квадратичной формы.

## 5 Метод Левенберга—Марквардта

Квадратичное представление метода Гаусса—Ньютона (8) формально справедливо лишь в малой области вокруг точки разложения  $x^{(k-1)}$ . Это значит, что предсказываемое квадратичной формой значение  $\Phi(x^{(k)})$  может сильно отличаться от реального значения при больших  $\Delta\mathbf{x}$ . Таким образом, сложное поведение функции, например седловина или резкое изменение производных, может оказаться серьёзным препятствием к сходимости. Одним из вариантов решения проблемы является модификация уравнения (9) следующим образом:

$$J^T J \Delta\mathbf{x}_* + \lambda^{(k)} I \Delta\mathbf{x}_* = J^T \mathbf{y}, \quad (11)$$

где  $I$  — единичная матрица, а  $\lambda^{(k)}$  — безразмерный положительный коэффициент, который изменяется согласно определённому алгоритму. Видно, что величина  $\Delta$  монотонно убывает с  $\lambda$ , что позволяет регулировать шаг правильно подбирая этот коэффициент. Отметим, что это уравнение совпадает с уравнением для поиска условного экстремума в области  $\|\Delta\mathbf{x}_*\| < \Delta(\lambda)$ .

В начале итераций выбирается начальное значение  $\lambda^{(0)} > 0$ , обычно берётся значение меньше единицы. Затем, на каждом итерационном шаге  $k$  рассчитывается два значения  $\Phi$  — для  $\Delta\mathbf{x}$ , получаемого при  $\lambda = \lambda^{(k-1)}$  и  $\lambda = \lambda^{(k-1)}/\nu$ , где  $\nu > 1$  — постоянный безразмерный коэффициент. Обозначим соответствующие значения  $\Phi$  как  $\Phi(\lambda^{(k-1)})$  и  $\Phi(\lambda^{(k-1)}/\nu)$ , а значение  $\Phi$  на предыдущем итерационном шаге  $k-1$  как  $\Phi^{(k-1)}$ . Тогда значение  $\mathbf{x}^{(k)} = x^{(k-1)} + \Delta\mathbf{x}$  ищется из решения (11), причём коэффициент  $\lambda^{(k)}$  выбирается согласно следующему алгоритму:

1. Если  $\Phi(\lambda^{(k-1)}/\nu) \leq \Phi$  значит бóльший шаг улучшил приближение, и  $\lambda^{(k)} = \lambda^{(k-1)}/\nu$
2. Если  $\Phi(\lambda^{(k-1)}/\nu) > \Phi$  и  $\Phi(\lambda^{(k-1)}/\nu) \leq \Phi(\lambda^{(k-1)})$ , то старый шаг дал лучший результат, и  $\lambda^{(k)} = \lambda^{(k-1)}$
3. Если  $\Phi(\lambda^{(k-1)}/\nu) > \Phi$  и  $\Phi(\lambda^{(k-1)}/\nu) > \Phi(\lambda^{(k-1)})$ , то результат ухудшился при обоих размерах шага и следует увеличивать  $\lambda$  в  $\hat{\nu}$  (это число не обязательно равно  $\nu$ ) столько  $w$  раз, сколько нужно чтоб  $\Phi(\lambda^{(k-1)}\hat{\nu}^w) \leq \Phi$ . После этого  $\lambda^{(k)} = \lambda^{(k-1)}\hat{\nu}^w$ .

Предложенный метод называется методом Левенберга—Марквардта и является эвристическим. Это значит, что можно придумать много правил по которым модифицируется  $\lambda$ , например, её увеличение в шаге 3 может производиться аддитивно, а не мультипликативно.

## 6 Метод сопряженных градиентов

Метод Левенберга—Марквардта, как и подобные ему методы, требует нахождения минимума квадратичной функции на каждом итерационном шаге. Если размерность решаемой задачи велика ( $n \gtrsim 20-30$ ), то нахождение

минимума квадратичной формы типа (8) с помощью обращения матрицы является вычислительно сложной задачей и требует использования  $O(n^2)$  памяти для хранения различных матриц. Метод сопряженных градиентов позволяет решить эту задачу эффективнее.

Пусть дана квадратичная функция  $Q$ :

$$Q(\mathbf{x}) = \frac{1}{2}(\mathbf{x}, H\mathbf{x}) + (\mathbf{c}, \mathbf{x}), \quad (12)$$

где  $H$  и  $\mathbf{c}$  — заданная положительно определенная матрица и заданный вектор.

Для нахождения минимума этой функции можно воспользоваться итерационным алгоритмом, состоящим из количества шагов  $n$  равного размерности  $\mathbf{x}$ . На первом шаге выбирается произвольное значение  $\mathbf{x}^{(0)}$ , а также задаются значения векторов  $\mathbf{g}^{(0)} = H\mathbf{x}^{(0)} + \mathbf{c}$  и  $\mathbf{p}^{(0)} = -\mathbf{g}^{(0)}$ . Затем для всех  $i$  от 0 до  $n - 1$  выполняется:

$$\begin{aligned} \alpha^{(i)} &= \frac{\|\mathbf{g}^{(i)}\|^2}{(p^{(i)}, Hp^{(i)})}, \\ \mathbf{x}^{(i+1)} &= \mathbf{x}^{(i)} + \alpha^{(i)}\mathbf{p}^{(i)}, \\ \mathbf{g}^{(i+1)} &= \mathbf{g}^{(i)} + \alpha^{(i)}H\mathbf{p}^{(i)}, \\ \beta^{(i)} &= \frac{\|\mathbf{g}^{(i+1)}\|^2}{\|\mathbf{g}^{(i)}\|^2}, \\ \mathbf{p}^{(i+1)} &= -\mathbf{g}^{(i+1)} + \beta^{(i)}\mathbf{p}^{(i)}. \end{aligned} \quad (13)$$

Обратим внимание на особенности этого метода:

- Этот метод основан на построении крыловских подпространств и является самым быстрым известным методом нахождения минимума  $Q(\mathbf{x})$ .
- Прост в реализации.
- Метод сходится к точному решению.
- Каждая последовательная итерация метода улучшает приближение найденное решение, поэтому выполнение метода можно прервать на шаге  $i < n - 1$  для получения приближенного решения.
- С другой стороны, из-за ошибок округления на практике принято использовать  $2n$  шагов алгоритма для достижения точного решения.
- Если значения  $H$  заданы программно, то требуется всего  $O(n)$  памяти для реализации алгоритма. Например, в случае решения задачи оптимизации,  $H$  может быть матрицей Гессе или  $J^T J$  и значения компонент матрицы могут быть вычислены программно.
- Этот алгоритм возможно видоизменить для работы с не положительно определенными  $H$ .

Отметим, что метод сопряженных градиентов используется и для решения задачи линейного МНК, если её размерность велика.