

Продвинутые методы многомерной оптимизации

Рассмотрим задачу безусловной оптимизации в многомерном пространстве:

$$f(\mathbf{x}) \rightarrow \min_{\mathbf{x} \in \mathbb{R}^N}.$$

Ранее для решения этой задачи были рассмотрены методы покоординатного и градиентного спуска, метод Ньютона, а также комбинированные методы. Во всех этих подходах очередное направление оптимизации \mathbf{d}_k вычисляется только с использованием информации от оракула в текущей точке \mathbf{x}_k . Таким образом, траектория оптимизации никак не учитывается¹. В методах оптимизации, рассматриваемых ниже, очередное направление оптимизации \mathbf{d}_k существенно зависит от траектории оптимизации и, в частности, зависит от предыдущих направлений $\mathbf{d}_{k-1}, \dots, \mathbf{d}_0$.

Решение СЛАУ с помощью метода сопряженных градиентов

Рассмотрим задачу минимизации квадратичной функции с положительно-определенным гессианом:

$$f(\mathbf{x}) = \frac{1}{2} \mathbf{x}^T A \mathbf{x} - \mathbf{x}^T \mathbf{b} \rightarrow \min_{\mathbf{x}}, \quad A = A^T \succ 0. \quad (1)$$

Приравнявая к нулю градиент f , получаем:

$$\nabla f(\mathbf{x}) = A \mathbf{x} - \mathbf{b} = \mathbf{0} \Rightarrow A \mathbf{x} = \mathbf{b}.$$

Таким образом, задача минимизации f эквивалентна решению СЛАУ $A \mathbf{x} = \mathbf{b}$.

Назовем набор векторов $\{\mathbf{d}_i\}_{i=0}^k$ **сопряженным относительно положительно-определенной матрицы A** , если выполнено условие:

$$\mathbf{d}_i^T A \mathbf{d}_j = 0 \quad \forall i \neq j.$$

Примером сопряженных направлений является набор собственных векторов матрицы A . Действительно, в этом случае $\mathbf{d}_i^T A \mathbf{d}_j = \lambda_i \mathbf{d}_i^T \mathbf{d}_j = 0$. Последнее условие выполнено, т.к. собственные вектора, отвечающие различным собственным значениям, ортогональны, а среди собственных векторов с одинаковым собственным значением всегда можно выбрать ортогональный набор.

Набор сопряженных векторов является линейно-независимым². Рассмотрим нетривиальную линейную комбинацию сопряженных векторов:

$$\sum_{i=0}^k \alpha_i \mathbf{d}_i = \mathbf{0}, \quad \exists \alpha_i \neq 0.$$

Домножим слева это уравнение на $\mathbf{d}_j^T A$ для некоторого j . В результате получим:

$$\sum_{i=0}^k \alpha_i \mathbf{d}_j^T A \mathbf{d}_i = \alpha_j \mathbf{d}_j^T A \mathbf{d}_j = 0 \Rightarrow \alpha_j = 0.$$

Последнее условие следует из того, что матрица A является строго положительно-определенной. Из линейной независимости вытекает, что сопряженный набор $\{\mathbf{d}_i\}_{i=0}^{N-1}$ является базисом всего пространства \mathbb{R}^N . В частности, решение СЛАУ \mathbf{x}_* можно разложить по этому базису с некоторыми коэффициентами α_i :

$$\mathbf{x}_* = \sum_{i=0}^{N-1} \alpha_i \mathbf{d}_i. \quad (2)$$

Домножая это уравнение слева на $\mathbf{d}_j^T A$, получаем:

$$\mathbf{d}_j^T A \mathbf{x}_* = \sum_{i=0}^{N-1} \alpha_i \mathbf{d}_j^T A \mathbf{d}_i = \alpha_j \mathbf{d}_j^T A \mathbf{d}_j \Rightarrow \alpha_j = \frac{\mathbf{d}_j^T A \mathbf{x}_*}{\mathbf{d}_j^T A \mathbf{d}_j} = \frac{\mathbf{d}_j^T \mathbf{b}}{\mathbf{d}_j^T A \mathbf{d}_j}. \quad (3)$$

¹это не совсем так при использовании адаптивной коррекции длины шага

²На самом деле набор сопряженных векторов можно рассматривать как ортогональный базис относительно скалярного произведения $(\mathbf{x}, \mathbf{y}) = \mathbf{x}^T A \mathbf{y}$. Отсюда становится очевидным линейная независимость векторов, а также то, что матрица A должна быть положительно-определенной (иначе не будет скалярного произведения).

Последнее равенство вытекает из того, что \mathbf{x}_* – решение СЛАУ, т.е. $A\mathbf{x}_* = \mathbf{b}$. Здесь становится очевидным преимущество сопряженного набора векторов относительно других базисов для поиска \mathbf{x}_* : для сопряженного набора коэффициенты разложения α_j вычисляются аналитически. Поиск \mathbf{x}_* по формуле (2) с коэффициентами (3) можно представить в виде итерационного процесса:

$$\mathbf{x}_0 = \mathbf{0}, \quad \mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{d}_k, \quad \alpha_k = \frac{\mathbf{d}_k^T \mathbf{b}}{\mathbf{d}_k^T A \mathbf{d}_k}, \quad k = 0, \dots, N.$$

В результате $\mathbf{x}_{N+1} = \mathbf{x}_*$.

Теперь получим аналогичный итерационный процесс для поиска \mathbf{x}_* , который начинается с произвольного ненулевого вектора \mathbf{x}_0 . Для этого разложим вектор $\mathbf{x}_* - \mathbf{x}_0$ по сопряженному базису:

$$\mathbf{x}_* - \mathbf{x}_0 = \sum_{i=0}^{N-1} \alpha_i \mathbf{d}_i.$$

Домножая это равенство на $\mathbf{d}_j^T A$ слева, получаем:

$$\mathbf{d}_j^T A(\mathbf{x}_* - \mathbf{x}_0) = \alpha_j \mathbf{d}_j^T A \mathbf{d}_j \Rightarrow \alpha_j = \frac{\mathbf{d}_j^T A(\mathbf{x}_* - \mathbf{x}_0)}{\mathbf{d}_j^T A \mathbf{d}_j} = \frac{\mathbf{d}_j^T (\mathbf{b} - A\mathbf{x}_0)}{\mathbf{d}_j^T A \mathbf{d}_j} = -\frac{\mathbf{d}_j^T \mathbf{g}_0}{\mathbf{d}_j^T A \mathbf{d}_j} = -\frac{\mathbf{d}_j^T \mathbf{g}_j}{\mathbf{d}_j^T A \mathbf{d}_j}. \quad (4)$$

Здесь и далее через \mathbf{g}_j обозначается градиент функции f в точке \mathbf{x}_j . Для квадратичной функции $\mathbf{g}_j = A\mathbf{x}_j - \mathbf{b}$. Докажем последний переход в формуле (4):

$$\mathbf{d}_j^T (\mathbf{g}_j - \mathbf{g}_0) = \mathbf{d}_j^T (A\mathbf{x}_j - \mathbf{b} - A\mathbf{x}_0 + \mathbf{b}) = \mathbf{d}_j^T A \left(\sum_{i=0}^{j-1} \alpha_i \mathbf{d}_i \right) = \sum_{i=0}^{j-1} \alpha_i \mathbf{d}_j^T A \mathbf{d}_i = 0 \Rightarrow \mathbf{d}_j^T \mathbf{g}_j = \mathbf{d}_j^T \mathbf{g}_0. \quad (5)$$

Таким образом, получаем следующий итерационный процесс для поиска \mathbf{x}_* из произвольного начального приближения \mathbf{x}_0 :

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{d}_k, \quad \alpha_k = -\frac{\mathbf{d}_k^T \mathbf{g}_k}{\mathbf{d}_k^T A \mathbf{d}_k}, \quad k = 0, \dots, N. \quad (6)$$

Можно показать, что данный итерационный процесс является процессом наискорейшего спуска для функции f вдоль сопряженных направлений $\{\mathbf{d}_i\}_{i=0}^{N-1}$. Для этого достаточно убедиться в том, что коэффициент α_j , вычисляемый по формуле (4), доставляет минимум по α функции $f(\mathbf{x}_j + \alpha \mathbf{d}_j)$. Действительно,

$$\begin{aligned} \left. \frac{\partial}{\partial \alpha} f(\mathbf{x}_j + \alpha \mathbf{d}_j) \right|_{\alpha=\alpha_j} &= \nabla f(\mathbf{x}_j + \alpha_j \mathbf{d}_j)^T \mathbf{d}_j = \mathbf{g}_{j+1}^T \mathbf{d}_j = (A\mathbf{x}_j + \alpha_j A \mathbf{d}_j - \mathbf{b})^T \mathbf{d}_j = \\ &= (A\mathbf{x}_j - \mathbf{b})^T \mathbf{d}_j + \alpha_j \mathbf{d}_j^T A \mathbf{d}_j = \mathbf{g}_j^T \mathbf{d}_j - \mathbf{d}_j^T \mathbf{g}_j = 0. \end{aligned} \quad (7)$$

Одновременно здесь было доказано, что $\mathbf{g}_{j+1}^T \mathbf{d}_j = 0$. Покажем, что $\mathbf{g}_{j+1}^T \mathbf{d}_i = 0 \forall i \leq j$. Действительно,

$$\mathbf{g}_{j+1}^T \mathbf{d}_i = (A\mathbf{x}_{j+1} - \mathbf{b})^T \mathbf{d}_i = (A(\mathbf{x}_0 + \sum_{i=0}^j \alpha_i \mathbf{d}_i) - \mathbf{b})^T \mathbf{d}_i = (A\mathbf{x}_0 - \mathbf{b})^T \mathbf{d}_i + \alpha_i \mathbf{d}_i^T A \mathbf{d}_i = \mathbf{g}_0^T \mathbf{d}_i - \mathbf{g}_i^T \mathbf{d}_i = 0.$$

Последнее равенство следует из (5). Условие ортогональности градиента \mathbf{g}_{j+1} всем предыдущим направлениям оптимизации означает, что \mathbf{x}_{j+1} доставляет минимум функции f в линейной оболочке $\mathcal{L}(\mathbf{d}_0, \dots, \mathbf{d}_j)$. Более того, можно показать, что точка $\mathbf{x}_j + \alpha \mathbf{d}_j$ минимизирует функцию f в линейной оболочке $\mathcal{L}(\mathbf{d}_0, \dots, \mathbf{d}_{j-1})$ для любого α . Действительно,

$$\nabla f(\mathbf{x}_j + \alpha \mathbf{d}_j)^T \mathbf{d}_i = (A(\mathbf{x}_j + \alpha \mathbf{d}_j) - \mathbf{b})^T \mathbf{d}_i = (A\mathbf{x}_j - \mathbf{b})^T \mathbf{d}_i + \alpha \mathbf{d}_j^T A \mathbf{d}_i = \mathbf{g}_j^T \mathbf{d}_i = 0. \quad \forall i < j.$$

Таким образом, при движении вдоль очередного сопряженного направления оптимизацию по предыдущим направлениям проводить не нужно. Это одна из отличительных особенностей метода сопряженных направлений.

Для проведения итерационного процесса (6) необходимо указать полный набор сопряженных направлений для матрицы A . Рассмотрим следующую схему генерации \mathbf{d}_k :

$$\mathbf{d}_0 = -\mathbf{g}_0, \quad \mathbf{d}_{k+1} = -\mathbf{g}_{k+1} + \beta_k \mathbf{d}_k, \quad \beta_k = \frac{\mathbf{g}_{k+1}^T A \mathbf{d}_k}{\mathbf{d}_k^T A \mathbf{d}_k}, \quad k = 0, \dots, N-1. \quad (8)$$

Верна следующая последовательность рассуждений:

$$\begin{aligned} \mathbf{d}_0 &= -\mathbf{g}_0 \in \mathcal{L}(\mathbf{g}_0), \\ \mathbf{g}_1 &= \mathbf{g}_0 + \alpha_0 A \mathbf{d}_0 = \mathbf{g}_0 - \alpha_0 A \mathbf{g}_0 \in \mathcal{L}(\mathbf{g}_0, A \mathbf{g}_0), \\ \mathbf{d}_1 &= -\mathbf{g}_1 + \beta_0 \mathbf{d}_0 \in \mathcal{L}(\mathbf{g}_0, A \mathbf{g}_0), \\ \mathbf{g}_2 &= \mathbf{g}_1 + \alpha_1 A \mathbf{d}_1 \in \mathcal{L}(\mathbf{g}_0, A \mathbf{g}_0, A^2 \mathbf{g}_0), \\ \mathbf{d}_2 &= -\mathbf{g}_2 + \beta_1 \mathbf{d}_1 \in \mathcal{L}(\mathbf{g}_0, A \mathbf{g}_0, A^2 \mathbf{g}_0), \\ &\dots \end{aligned}$$

В результате набор $\{\mathbf{d}_0, \dots, \mathbf{d}_i\}$, генерируемый по схеме (8), и набор $\{\mathbf{g}_0, \dots, \mathbf{g}_i\}$ принадлежат одному и тому же линейному пространству $\mathcal{L}(\mathbf{g}_0, A\mathbf{g}_0, \dots, A^i\mathbf{g}_0)$ ³. Следовательно, вектор $A\mathbf{d}_i$ можно представить в базисе $\{\mathbf{d}_0, \dots, \mathbf{d}_k\}$, $k > i$, а вектор \mathbf{g}_i можно представить в базисе $\{\mathbf{d}_0, \dots, \mathbf{d}_i\}$:

$$A\mathbf{d}_i = \sum_{j=0}^k a_j \mathbf{d}_j, \quad (9)$$

$$\mathbf{g}_i = \sum_{j=0}^i b_j \mathbf{d}_j. \quad (10)$$

Покажем теперь, что схема (8) позволяет построить набор сопряженных направлений для матрицы A . Кроме того, покажем, что при ее использовании $\mathbf{g}_k^T \mathbf{d}_i = 0 \ \forall i < k$ и $\mathbf{g}_k^T \mathbf{g}_i = 0 \ \forall i < k$. Проведем доказательство по индукции. Пусть известно, что

$$\begin{aligned} \mathbf{g}_k^T \mathbf{d}_i &= 0 \ \forall i < k, \\ \mathbf{d}_k^T A\mathbf{d}_i &= 0 \ \forall i < k, \\ \mathbf{g}_k^T \mathbf{g}_i &= 0 \ \forall i < k. \end{aligned}$$

Докажем, что аналогичные утверждения верны для $k+1$. Из рассуждений (7) следует, что $\mathbf{g}_{k+1}^T \mathbf{d}_k = 0 \ \forall k$. Далее

$$\mathbf{g}_{k+1}^T \mathbf{d}_i = (\mathbf{g}_k + \alpha_k A\mathbf{d}_k)^T \mathbf{d}_i = \mathbf{g}_k^T \mathbf{d}_i + \alpha_k \mathbf{d}_k^T A\mathbf{d}_i = 0.$$

Последнее утверждение выполняется, исходя из предположения индукции. Покажем теперь, что $\mathbf{d}_{k+1}^T A\mathbf{d}_i = 0 \ \forall i \leq k$. Используя (8), получим, что

$$\mathbf{d}_{k+1}^T A\mathbf{d}_k = (-\mathbf{g}_{k+1} + \beta_k \mathbf{d}_k)^T A\mathbf{d}_k = -\mathbf{g}_{k+1}^T A\mathbf{d}_k + \mathbf{g}_{k+1}^T A\mathbf{d}_k = 0 \ \forall k.$$

Далее с помощью (9) заключаем, что для $i < k$

$$\mathbf{d}_{k+1}^T A\mathbf{d}_i = (-\mathbf{g}_{k+1} + \beta_k \mathbf{d}_k)^T A\mathbf{d}_i = -\mathbf{g}_{k+1}^T A\mathbf{d}_i = -\mathbf{g}_{k+1}^T \left(\sum_{j=0}^k a_j \mathbf{d}_j \right) = -\sum_{j=0}^k a_j \mathbf{g}_{k+1}^T \mathbf{d}_j = 0.$$

Осталось показать, что $\mathbf{g}_{k+1}^T \mathbf{g}_i = 0 \ \forall i < k$. Используя (10), получаем

$$\mathbf{g}_{k+1}^T \mathbf{g}_i = \mathbf{g}_{k+1}^T \left(\sum_{j=0}^i b_j \mathbf{d}_j \right) = \sum_{j=0}^i b_j \mathbf{g}_{k+1}^T \mathbf{d}_j = 0.$$

В результате получаем, что набор $\{\mathbf{d}_0, \dots, \mathbf{d}_{N-1}\}$ действительно является сопряженным относительно матрицы A . Заметим, что в доказательстве сопряженности существенно используется тот факт, что первое направление выбирается в соответствии с антиградиентом. При другом выборе \mathbf{d}_0 итерационный процесс (8) не приводит к набору сопряженных направлений.

С помощью доказанных выше свойств $\mathbf{g}_k^T \mathbf{d}_i = 0 \ \forall i < k$ и $\mathbf{g}_k^T \mathbf{g}_i = 0 \ \forall i < k$ можно несколько упростить выражения для α_k и β_k в итерационных процессах (6), (8):

$$\begin{aligned} \alpha_k &= -\frac{\mathbf{g}_k^T \mathbf{d}_k}{\mathbf{d}_k^T A\mathbf{d}_k} = -\frac{\mathbf{g}_k^T (-\mathbf{g}_k + \beta_{k-1} \mathbf{d}_{k-1})}{\mathbf{d}_k^T A\mathbf{d}_k} = \frac{\mathbf{g}_k^T \mathbf{g}_k}{\mathbf{d}_k^T A\mathbf{d}_k}, \\ \beta_k &= \frac{\mathbf{g}_{k+1}^T A\mathbf{d}_k}{\mathbf{d}_k^T A\mathbf{d}_k} = \frac{\mathbf{g}_{k+1}^T (\mathbf{g}_{k+1} - \mathbf{g}_k)}{\mathbf{d}_k^T A\mathbf{d}_k \alpha_k} = \frac{\mathbf{g}_{k+1}^T \mathbf{g}_{k+1}}{\mathbf{g}_k^T \mathbf{g}_k}. \end{aligned} \quad (11)$$

В результате получаем итоговый алгоритм решения СЛАУ $A\mathbf{x} = \mathbf{b}$, который получил название метода сопряженных градиентов:

1. Задаем начальное приближение \mathbf{x}_0 и требуемую точность ε ;
2. Инициализация $\mathbf{d}_0 = -\mathbf{g}_0$, $k = 0$;
3. $\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{d}_k$, где $\alpha_k = \frac{\mathbf{g}_k^T \mathbf{g}_k}{\mathbf{d}_k^T A\mathbf{d}_k}$;
4. Если $\alpha_k \|\mathbf{d}_k\| < \varepsilon$, то стоп;

³Пространства такого вида известны в линейной алгебре как пространства Крылова

$$5. \mathbf{d}_{k+1} = -\mathbf{g}_{k+1} + \beta_k \mathbf{d}_k, \text{ где } \beta_k = \frac{\mathbf{g}_{k+1}^T \mathbf{g}_{k+1}}{\mathbf{g}_k^T \mathbf{g}_k};$$

6. $k = k + 1$ и переход к шагу 3.

Данный алгоритм гарантированно сходится к решению за N шагов, где N – размерность пространства решения. На каждом шаге итерационного процесса требуется проводить только одно умножение матрицы A на вектор \mathbf{d}_k (вектор $A\mathbf{x}_{k+1}$ при этом вычисляется как $A\mathbf{x}_k + \alpha_k A\mathbf{d}_k$). Остальные операции в алгоритме являются векторными: скалярное произведение двух векторов и сумма двух векторов. На рис. 1 показан пример применения этого алгоритма. Для сравнения показана также траектория метода наискорейшего спуска.

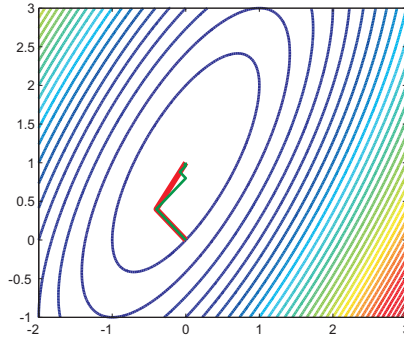


Рис. 1: Пример работы метода наискорейшего спуска (зеленая траектория) и метода сопряженных градиентов (красная траектория) для оптимизации квадратичной функции.

Основные преимущества метода сопряженных градиентов связаны с пространствами большой размерности. При $N \gg 1$ методы решения СЛАУ, основанные на матричных разложениях, например, разложении Холецкого или QR-разложении, перестают быть применимыми в силу необходимости итерационного пересчета матриц, размер которых совпадает с размером матрицы A . Напротив, в методе сопряженных градиентов все операции производятся только для векторов размерности N , а самая сложная операция в алгоритме – это умножение матрицы A на очередной вектор \mathbf{d}_k . Для ряда матриц специального вида, например, разреженных матриц или матриц, представляющих базис Фурье, такое умножение может быть проведено эффективнее общего случая.

Метод сопряженных градиентов для минимизации произвольной функции

Рассмотрим теперь задачу оптимизации произвольной гладкой функции f . Тогда в схеме метода сопряженных градиентов матрица A заменяется на гессиан H_k в текущей точке \mathbf{x}_k . Практически данный подход превращается в метод Ньютона, в котором квадратичная аппроксимация функции минимизируется с помощью сопряженных градиентов. Поэтому метод сопряженных градиентов имеет квадратичную скорость сходимости в малой окрестности оптимального решения.

Для того, чтобы застраховаться от возможной неадекватности квадратичного приближения функции f в текущей точке, в методе сопряженных градиентов предлагается решать одномерную задачу оптимизации при движении вдоль очередного направления \mathbf{d}_k :

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{d}_k, \quad \alpha_k = \arg \min_{\alpha} f(\mathbf{x}_k + \alpha \mathbf{d}_k).$$

Заметим, что в этом случае отпадает необходимость вычисления гессиана, и метод превращается в метод оптимизации первого порядка. При использовании формулы (11) для β_k соответствующий метод сопряженных градиентов получил название Флетчера-Ривса (Fletcher-Reeves). В методе Полака-Рибье (Polak-Ribiere) предлагается использовать другую формулу для β_k :

$$\beta_k = \frac{\mathbf{g}_{k+1}^T \mathbf{g}_{k+1} - \mathbf{g}_{k+1}^T \mathbf{g}_k}{\mathbf{g}_k^T \mathbf{g}_k}.$$

Для случая квадратичной функции последняя формула переходит в формулу (11), т.к. $\mathbf{g}_{k+1}^T \mathbf{g}_k = 0$. Однако, для произвольной функции она позволяет добиться более устойчивой и качественной работы метода.

В методе сопряженных градиентов направление \mathbf{d}_{k+1} зависит от \mathbf{d}_k , а, значит, неявно зависит от всех предыдущих направлений $\mathbf{d}_0, \dots, \mathbf{d}_k$. Для повышения устойчивости работы метода для неквадратичной функции предлагается устанавливать $\beta_k = 0$ после каждой N -ой итерации, т.е. периодически «очищать» предысторию, в которой могут накапливаться неудачные направления. Обнуление β_k соответствует выбору направления оптимизации по антиградиенту. На рис. 2,а показан пример применения метода сопряженных градиентов без использования обнуления. В результате в ходе итераций метод «проскочил» оптимальную точку $[1, 1]^T$ за счет

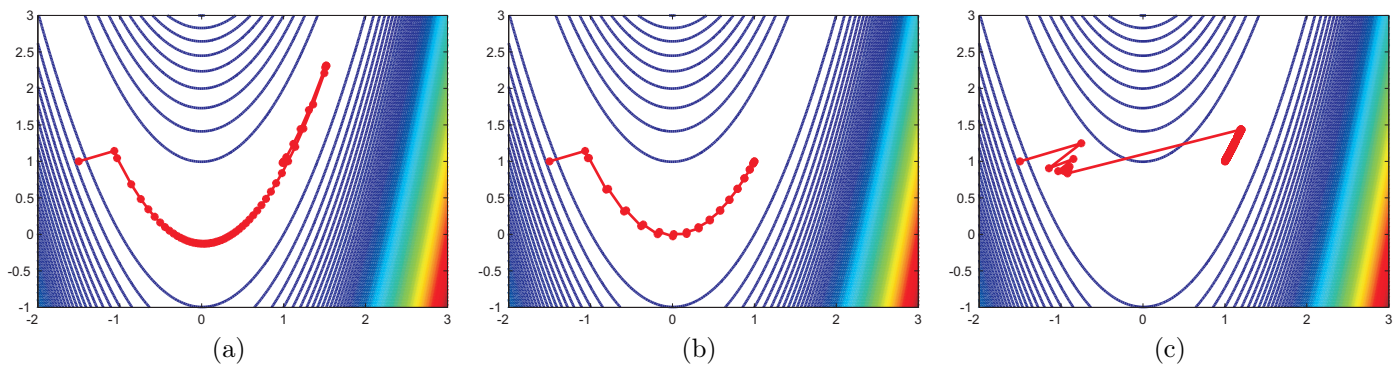


Рис. 2: Примеры работы метода сопряженных градиентов для функции Розенблока. Случай (а): без использования обнуления β_k , всего 64 итерации, случай (b): с использованием обнуления, всего 36 итераций, случай (с): использование backtracking, всего 1260 итераций.

сильной зависимости от предыдущих направлений. Напротив, использование обнуления (см. рис. 2,b) позволило успешно обнаружить минимум за меньшее число итераций.

Как было отмечено выше, в методе сопряженных градиентов очередное направление оптимизации выбирается таким образом, чтобы при движении вдоль него сохранить минимум по всем предыдущим направлениям. За счет этого удается избежать ступенчатого поведения, характерного для покоординатного и градиентного спуска. Однако, такой подход неявно предполагает, что вдоль очередного направления проводится точная оптимизация, т.к. в дальнейшем возврат к этому направлению не запланирован. На рис. 2,с показан пример работы метода, в котором на этапе одномерной оптимизации используется backtracking. В результате метод начинает работать крайне неустойчиво, а сходимость достигается только за 1260 итераций. В реальности метод сопряженных градиентов может устойчиво сходиться и при использовании неточной одномерной оптимизации. Однако, ее использование связано с определенными рисками.

Квази-ньютоновские методы

В квази-ньютоновских методах оптимизации пересчет осуществляется по формуле

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \alpha_k S_k \mathbf{g}_k. \quad (12)$$

Здесь S_k – некоторая положительно-определенная матрица. Условие положительной определенности S_k гарантирует возможность уменьшения функции f вдоль направления $\mathbf{d}_k = -S_k \mathbf{g}_k$. Действительно,

$$\left. \frac{\partial}{\partial \alpha} f(\mathbf{x}_k - \alpha S_k \mathbf{g}_k) \right|_{\alpha=0} = -\mathbf{g}_k^T S_k \mathbf{g}_k < 0.$$

Если $S_k = I$, то (12) переходит в градиентный спуск. Если $S_k = H_k^{-1}$, то (12) переходит в метод Ньютона.

По аналогии с методом сопряженных градиентов, рассмотрим сначала квази-ньютоновские методы для случая минимизации квадратичной функции (1). Введем обозначения: $\delta_k = \mathbf{x}_{k+1} - \mathbf{x}_k = -\alpha_k S_k \mathbf{g}_k$, $\gamma_k = \mathbf{g}_{k+1} - \mathbf{g}_k$. Для квадратичной функции $\gamma_k = \mathbf{g}_{k+1} - \mathbf{g}_k = A \delta_k \forall k$. В результате

$$[\gamma_0 \ \gamma_1 \ \dots \ \gamma_{N-1}] = A[\delta_0 \ \delta_1 \ \dots \ \delta_{N-1}].$$

Пусть в первый момент времени задано некоторое начальное приближение \mathbf{x}_0 . Положим $S_0 = I^4$ и будем пересчитывать S_k по правилу $S_{k+1} = S_k + C_k$, где C_k – некоторая матрица. Проведем N шагов вида (12) и получим набор $\{\delta_0, \dots, \delta_{N-1}\}$, набор $\{\gamma_0, \dots, \gamma_{N-1}\}$ и набор матриц S_1, S_2, \dots, S_N . Если оказывается, что

$$S_N [\gamma_0 \ \gamma_1 \ \dots \ \gamma_{N-1}] = [\delta_0 \ \delta_1 \ \dots \ \delta_{N-1}],$$

то матрица $S_N = A^{-1}$ и следующий шаг по схеме (12) соответствует шагу Ньютона, т.е. $\mathbf{x}_{N+1} = \mathbf{x}_*$.

По аналогии с методом сопряженных градиентов, в котором существуют разные формулы для β_k и, соответственно, разные итерационные схемы пересчета \mathbf{d}_k , для квази-ньютоновских методов также предложено несколько способов выбора C_k . Однако, все эти способы гарантируют выполнение следующих свойств:

1. Метод сходится за N шагов для квадратичной функции;

⁴это соответствует использованию направления антиградиента

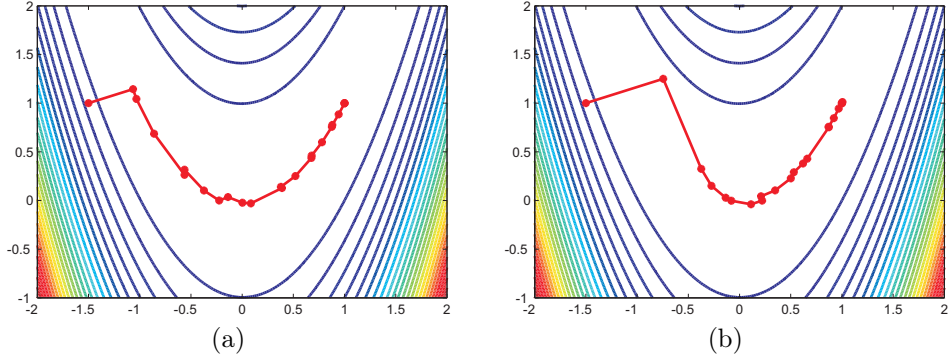


Рис. 3: Примеры работы метода L-BFGS для функции Розенблока. Случай (a): точная одномерная оптимизация, случай (b): использование метода Флетчера.

2. Для квадратичной функции набор векторов $\{\delta_0, \dots, \delta_{N-1}\}$ образует сопряженный базис относительно матрицы A ;
3. Матрица S_k всегда остается положительно-определенной.

Последнее свойство важно для принципиальной возможности уменьшения f на каждой итерации.

Рассмотрим несколько схем выбора C_k :

- [Схема DFP \(Davidon-Fletcher-Powell\)](#).

$$S_{k+1} = S_k + \frac{\delta_k \delta_k^T}{\delta_k^T \gamma_k} - \frac{S_k \gamma_k \gamma_k^T S_k}{\gamma_k^T S_k \gamma_k}.$$

- [Схема BFGS \(Broyden-Fletcher-Goldfarb-Shanno\)](#).

$$S_{k+1} = S_k + \left(1 + \frac{\gamma_k^T S_k \gamma_k}{\gamma_k^T \delta_k}\right) \frac{\delta_k \delta_k^T}{\gamma_k^T \delta_k} - \frac{\delta_k \gamma_k^T S_k + S_k \gamma_k \delta_k^T}{\gamma_k^T \delta_k}.$$

- [Схема L-BGFS \(Limited Memory BFGS\)](#). Формула пересчета аналогична BFGS, но для $S_k = I$:

$$S_{k+1} = I + \left(1 + \frac{\gamma_k^T \gamma_k}{\gamma_k^T \delta_k}\right) \frac{\delta_k \delta_k^T}{\gamma_k^T \delta_k} - \frac{\delta_k \gamma_k^T + \gamma_k \delta_k^T}{\gamma_k^T \delta_k}.$$

Подставляя эту формулу в (12), получаем:

$$\begin{aligned} \mathbf{d}_{k+1} &= -S_{k+1} \mathbf{g}_{k+1} = -\mathbf{g}_{k+1} + A_k \delta_k + B_k \gamma_k, \\ A_k &= -\frac{\mathbf{g}_{k+1}^T \delta_k}{\gamma_k^T \delta_k} \left(1 + \frac{\gamma_k^T \gamma_k}{\gamma_k^T \delta_k}\right) + \frac{\mathbf{g}_{k+1}^T \gamma_k}{\gamma_k^T \delta_k}, \\ B_k &= -\frac{\mathbf{g}_{k+1}^T \delta_k}{\gamma_k^T \delta_k}. \end{aligned}$$

Заметим, что все квази-ньютонские методы являются методами первого порядка. В случае квадратичной функции направления $\delta_0, \dots, \delta_{N-1}$ являются сопряженными относительно A . Поэтому в этом случае все квази-ньютонские методы эквивалентны методу сопряженных градиентов. При этом существенным моментом является использование направления антиградиента в первый момент времени ($S_0 = I$). При использовании другой матрицы S_0 генерируемые направления оптимизации $\delta_0, \dots, \delta_{N-1}$ теряют свойства сопряженности, а сам метод – свойство гарантированной сходимости за N итераций.

На рис. 3,а показан пример работы метода L-BFGS для функции Розенблока. Заметим, что в отличие от метода сопряженных градиентов, в квази-ньютонских методах нет необходимости «очищать» предысторию и периодически приравнивать $S_k = I$. Кроме того, квази-ньютонские методы являются более толерантными к использованию неточной одномерной оптимизации (см. рис. 3,б).