

Многомерная оптимизация

Звонарев Никита

Слушатели: 4-й курс, специальность СМ-СМ

Санкт-Петербургский Государственный Университет
Кафедра Статистического моделирования



Санкт-Петербург – 2019 г.

$\mathcal{D} \subset \mathbb{R}^n$ — область, $f(\mathbf{x})$ — целевая функция, $f : \mathcal{D} \rightarrow \mathbb{R}$.

Задача:

$$\mathbf{x}^* = \arg \min_{\mathbf{x} \in \mathcal{D}} f(\mathbf{x}).$$

Теперь $n > 1$ (да, n вместо k).

Идея: свести задачу к одномерной оптимизации. (Как?)

Подход: итеративный, каждая итерация — в два этапа:

- 1 Поиск направления оптимизации ($\mathbf{x}_k$ — текущая точка, направление — $\mathbf{p}_k$ (как найти?)).
- 2 Шаг по направлению $\mathbf{p}_k$ либо одномерная оптимизация:
 $\alpha_k^* = \arg \min_{0 \leq \alpha \leq 1} f(\mathbf{x}_k + \alpha \mathbf{p}_k), \mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k^* \mathbf{p}_k.$

$\mathcal{D} \subset \mathbb{R}^n$ — область, $f(\mathbf{x})$ — целевая функция, $f : \mathcal{D} \rightarrow \mathbb{R}$.

Задача:

$$\mathbf{x}^* = \arg \min_{\mathbf{x} \in \mathcal{D}} f(\mathbf{x}).$$

Теперь $n > 1$ (да, n вместо k).

Идея: свести задачу к одномерной оптимизации. (Как?)

Подход: итеративный, каждая итерация — в два этапа:

- 1 Поиск направления оптимизации ($\mathbf{x}_k$ — текущая точка, направление — $\mathbf{p}_k$ (как найти?)).
- 2 Шаг по направлению $\mathbf{p}_k$ либо одномерная оптимизация:
 $\alpha_k^* = \arg \min_{0 \leq \alpha \leq 1} f(\mathbf{x}_k + \alpha \mathbf{p}_k)$, $\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k^* \mathbf{p}_k$.

Он же *coordinate descent*. Самое простое, что можно придумать.

В терминах $\mathbf{p}_k$: $\mathbf{p}_k = \mathbf{e}_{(k \bmod n)+1}$, $\mathbf{e}_i \in \mathbb{R}^n$ — i -й орт.

Схема итерации та же:

$$\alpha_k^* = \arg \min_{\alpha \in \mathbb{R}} f(\mathbf{x}_k + \alpha \mathbf{p}_k), \quad \mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k^* \mathbf{p}_k.$$

Что выполнено: $f(\mathbf{x}_{k+1}) \leq f(\mathbf{x}_k)$.

Вопрос: это гарантирует сходимость к локальному минимуму?

Он же *coordinate descent*. Самое простое, что можно придумать.

В терминах $\mathbf{p}_k$: $\mathbf{p}_k = \mathbf{e}_{(k \bmod n)+1}$, $\mathbf{e}_i \in \mathbb{R}^n$ — i -й орт.

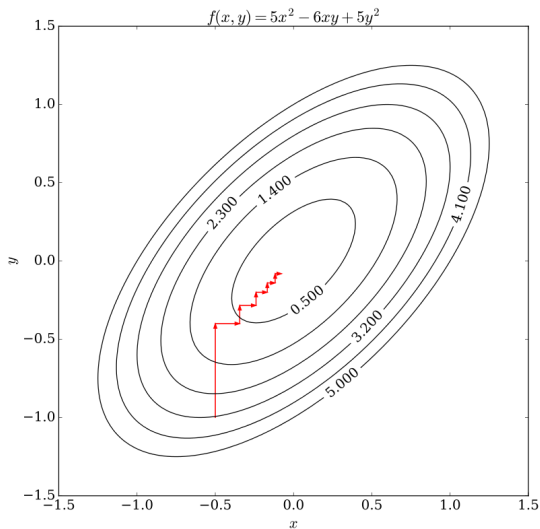
Схема итерации та же:

$$\alpha_k^* = \arg \min_{\alpha \in \mathbb{R}} f(\mathbf{x}_k + \alpha \mathbf{p}_k), \quad \mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k^* \mathbf{p}_k.$$

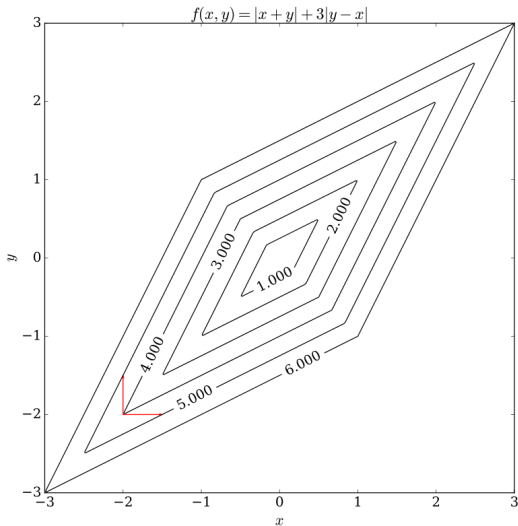
Что выполнено: $f(\mathbf{x}_{k+1}) \leq f(\mathbf{x}_k)$.

Вопрос: это гарантирует сходимость к локальному минимуму?

Покоординатный спуск: пример, когда есть сходимость



Покоординатный спуск: пример, когда нет сходимости



Ой! Метод может “застрять” там, где нет локального минимума.
На гладких функциях может сходиться медленно.

Получается, в нём нет смысла?

Идею используют. Например, вместо спуска по одной координате разбивают координаты на два набора, $n_1 + n_2 = n$, и чередуют их. На итерации — многомерная оптимизация по подмножеству координат.

Зачем так делают? Когда напрямую $f(x)$ оптимизировать слишком сложно.

Применения: статистика (важный частный случай — *ЕМ-алгоритм*),
...

Он же *gradient descent*. f — дифференцируемая

$\mathbf{p}_k$ — направление *антиградиента*: $\mathbf{p}_k = -f'(\mathbf{x}_k)$.

Схема итерации та же:

$$\alpha_k^* = \arg \min_{0 \leq \alpha < \infty} f(\mathbf{x}_k + \alpha \mathbf{p}_k), \quad \mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k^* \mathbf{p}_k.$$

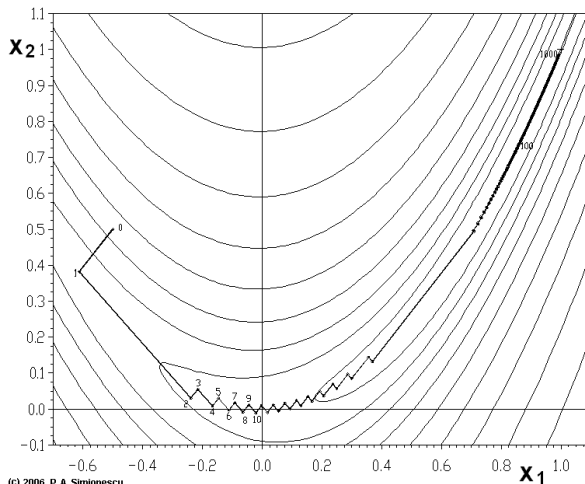
Можно выбрать такие малые α_k^* , что будет выполнено:

$$f(\mathbf{x}_{k+1}) \leq f(\mathbf{x}_k).$$

Сходимость выполняется в некоторых специфических случаях (выпуклость f , липшицевость f' , ...).

Градиентный спуск: пример медленной сходимости

Rosenbrock function: $f(x_1, x_2) = (1 - x_1)^2 + 100(x_2 - x_1^2)^2$.



Градиентный спуск: применения

Применяют, когда более сложные методы слишком дороги по времени/памяти. В этом случае применяют не поиск по направлению, а шаг на определенное расстояние.

В статистике (точнее, машинном обучении) применяют вариант градиентного спуска к целевым функциям вида:

$$f(\mathbf{x}) = \frac{1}{N} \sum_1^N f_i(\mathbf{x}).$$

f_i — это

Вместо того, чтобы на каждой итерации оптимизировать f , случайным образом разыгрывается индекс i , и вычисляется шаг по антиградиенту $-f'_i$ с угасающим весом. Получили метод стохастического градиента (*Stochastic gradient descent, SGD*).

Модификации: AdaGrad, RMSProp, Adam, ... (2010-е)

Метод сопряжённых градиентов, квадратичная функция

Он же *Conjugated gradient*, CG.

Хорошее название: никаких градиентов в нём нет.

Рассмотрим частный случай: $\arg \min_{\mathbf{x} \in \mathbb{R}^n} f(\mathbf{x}) = \frac{1}{2} \mathbf{x}^T \mathbf{A} \mathbf{x} - \mathbf{b}^T \mathbf{x}$, $\mathbf{A}$ — симметричная, положительно определенная матрица.

Эквивалентная задача: решение СЛАУ $\mathbf{A} \mathbf{x} = \mathbf{b}$.

Обозначим $r(\mathbf{x}) = \mathbf{A} \mathbf{x} - \mathbf{b}$.

Рассмотрим систему из n сопряжённых (т.е. ортогональных с $\mathbf{A}$) векторов $\mathbf{p}_1, \dots, \mathbf{p}_n$: $\mathbf{p}_i^T \mathbf{A} \mathbf{p}_j = 0$, $i \neq j$, $\mathbf{p}_i \neq 0$.

Возьмём произвольное $\mathbf{x}_0$, и найдём решение системы в виде разложения по векторам $\mathbf{p}_k$, то есть, $\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{p}_k$. α_k можно найти, взяв минимум по направлению:

$$\alpha_k = -\frac{r_k^T \mathbf{p}_k}{\mathbf{p}_k^T \mathbf{A} \mathbf{p}_k}, \quad r_k = r(\mathbf{x}_k).$$

Метод сопряжённых градиентов, квадратичная функция

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{p}_k.$$

$$\alpha_k = -\frac{\mathbf{r}_k^T \mathbf{p}_k}{\mathbf{p}_k^T \mathbf{A} \mathbf{p}_k}, \quad r_k = r(\mathbf{x}_k).$$

Теорема

Для любого $\mathbf{x}_0 \in \mathbb{R}^n$ последовательность $\mathbf{x}_k$ сходится к решению $\mathbf{x}^*$ в худшем случае за n шагов.

Вопрос: как найти векторы $\mathbf{p}_k$?

$$\mathbf{p}_k = -\mathbf{r}_k + \beta_k \mathbf{p}_{k-1},$$

$$\beta_k = \frac{\mathbf{r}_k^T \mathbf{A} \mathbf{p}_{k-1}}{\mathbf{p}_{k-1}^T \mathbf{A} \mathbf{p}_{k-1}}.$$

Nocedal, Jorge, Wright, S. Numerical Optimization.

Метод сопряжённых градиентов, алгоритм

$$\mathbf{r}_0 = \mathbf{Ax}_0 - \mathbf{b}, \mathbf{p}_0 = -\mathbf{r}_0, k = 0.$$

Пока $\mathbf{r}_k \neq 0$:

- $\alpha_k = \frac{\mathbf{r}_k^T \mathbf{r}_k}{\mathbf{p}_k^T \mathbf{A} \mathbf{p}_k}$
- $\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{p}_k$
- $\mathbf{r}_{k+1} = \mathbf{r}_k + \alpha_k \mathbf{A} \mathbf{p}_k$
- $\beta_{k+1} = \frac{\mathbf{r}_{k+1}^T \mathbf{r}_{k+1}}{\mathbf{r}_k^T \mathbf{r}_k}$
- $\mathbf{p}_{k+1} = -\mathbf{r}_{k+1} + \beta_{k+1} \mathbf{p}_k$
- Положить $k = k + 1$

Метод сопряжённых градиентов для произвольной функции

$f_i = f(\mathbf{x}_i)$, $\nabla f_i = f'(\mathbf{x}_i)$. f — дифференцируемая.

Метод Fletcher-Reeves:

$\mathbf{p}_0 = -\nabla f_0$, $k = 0$.

Пока не выполнен критерий остановки:

- $\alpha_k = \arg \min_{0 \leq \alpha < \infty} f(\mathbf{x}_k + \alpha \mathbf{p}_k)$
- $\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{p}_k$
- Вычислить ∇f_{k+1}
- $\beta_{k+1}^{FR} = \frac{\nabla f_{k+1}^T \nabla f_{k+1}}{\nabla f_k^T \nabla f_k}$
- $\mathbf{p}_{k+1} = -\nabla f_{k+1} + \beta_{k+1}^{FR} \mathbf{p}_k$
- Положить $k = k + 1$

Метод Polak-Ribiere:

$$\beta_{k+1}^{PR} = \frac{\nabla f_{k+1}^T (\nabla f_{k+1} - \nabla f_k)}{\nabla f_k^T \nabla f_k}.$$

Многомерный метод Ньютона

$$\mathbf{p}_k = -\nabla^2 f_k^{-1} \nabla f_k, \quad f_k = f(\mathbf{x}_k).$$

$\nabla^2 f_k$ — матрица Гессе (матрица вторых производных).

Теорема

Предположим, что f дважды дифференцируема, гессиан $\nabla^2 f(\mathbf{x})$ — липшицев в окрестности точки $\mathbf{x}^*$, в которой выполнены достаточные условия локального минимума. Тогда

- 1 Если $\mathbf{x}_0$ достаточно близко к $\mathbf{x}^*$, то последовательность сойдётся к $\mathbf{x}^*$.
- 2 Скорость сходимости $\mathbf{x}_k$ к $\mathbf{x}^*$ квадратичная.
- 3 $\|\nabla f_k\|$ сходится квадратично к 0.

Достаточное условие строгого локального минимума в точке $\mathbf{x}^*$: $\nabla^2 f_k$ — непрерывна в открытой окрестности $\mathbf{x}^*$, $\nabla f(\mathbf{x}^*) = 0$, $\nabla^2 f(\mathbf{x}^*)$ положительно определена.

Липшицевость: $\|\nabla^2 f(\mathbf{x}_1) - \nabla^2 f(\mathbf{x}_2)\| \leq M \|\mathbf{x}_1 - \mathbf{x}_2\|$ для любых $\mathbf{x}_1, \mathbf{x}_2$ в некоторой окрестности $\mathbf{x}^*$.

Многомерный метод Ньютона

$$\mathbf{p}_k = -\nabla^2 f_k^{-1} \nabla f_k, \quad f_k = f(\mathbf{x}_k).$$

Вопрос: сколько итераций требуется методу для сходимости на квадратичной функции?

Как вычислять градиент/матрицу Гессе: честно/численно (обсуждали).

Можно искать минимум по направлению $\mathbf{p}_k$, либо задать длину шага: константа ($\alpha_k = 1$) — не самое лучшее решение.

Backtracking: $\alpha_k^{(0)} = 1, \alpha_k^{(j+1)} = \alpha_k^{(j)}/2$.

Если $f(\mathbf{x}_k) \geq f(\mathbf{x}_k + \alpha_k^{(j)} \mathbf{p}_k)$, то $\alpha_k = \alpha_k^{(j)}$.

Когда можно обойтись без вычисления матрицы Гессе?

Пример: пусть $f = \|g(\mathbf{x}) - \mathbf{b}\|^2, g: \mathbb{R}^n \rightarrow \mathbb{R}^M, \mathbf{b} \in \mathbb{R}^M$.

Метод Гаусса-Ньютона: $\mathbf{p}_k = -(\nabla g(\mathbf{x}_k))^+(g(\mathbf{x}_k) - \mathbf{b}), \nabla g(\mathbf{x}_k) \in \mathbb{R}^{M \times n}$ — якобиан, $\mathbf{F}^+ = (\mathbf{F}^T \mathbf{F})^{-1} \mathbf{F}^T$ — псевдообращение Мура-Пенроуза.

На практике: методы *BFGS*, *L-BFGS-B* — используют приближение гессиана.

Многомерный метод Ньютона

$$\mathbf{p}_k = -\nabla^2 f_k^{-1} \nabla f_k, \quad f_k = f(\mathbf{x}_k).$$

Вопрос: сколько итераций требуется методу для сходимости на квадратичной функции?

Как вычислять градиент/матрицу Гессе: честно/численно (обсуждали).

Можно искать минимум по направлению $\mathbf{p}_k$, либо задать длину шага: константа ($\alpha_k = 1$) — не самое лучшее решение.

Backtracking: $\alpha_k^{(0)} = 1, \alpha_k^{(j+1)} = \alpha_k^{(j)}/2$.

Если $f(\mathbf{x}_k) \geq f(\mathbf{x}_k + \alpha_k^{(j)} \mathbf{p}_k)$, то $\alpha_k = \alpha_k^{(j)}$.

Когда можно обойтись без вычисления матрицы Гессе?

Пример: пусть $f = \|g(\mathbf{x}) - \mathbf{b}\|^2, g: \mathbb{R}^n \rightarrow \mathbb{R}^M, \mathbf{b} \in \mathbb{R}^M$.

Метод Гаусса-Ньютона: $\mathbf{p}_k = -(\nabla g(\mathbf{x}_k))^+(g(\mathbf{x}_k) - \mathbf{b}), \nabla g(\mathbf{x}_k) \in \mathbb{R}^{M \times n}$ — якобиан, $\mathbf{F}^+ = (\mathbf{F}^T \mathbf{F})^{-1} \mathbf{F}^T$ — псевдообращение Мура-Пенроуза.

На практике: методы *BFGS*, *L-BFGS-B* — используют приближение гессиана.

Многомерный метод Ньютона

$$\mathbf{p}_k = -\nabla^2 f_k^{-1} \nabla f_k, \quad f_k = f(\mathbf{x}_k).$$

Вопрос: сколько итераций требуется методу для сходимости на квадратичной функции?

Как вычислять градиент/матрицу Гессе: честно/численно (обсуждали).

Можно искать минимум по направлению $\mathbf{p}_k$, либо задать длину шага: константа ($\alpha_k = 1$) — не самое лучшее решение.

Backtracking: $\alpha_k^{(0)} = 1, \alpha_k^{(j+1)} = \alpha_k^{(j)}/2$.

Если $f(\mathbf{x}_k) \geq f(\mathbf{x}_k + \alpha_k^{(j)} \mathbf{p}_k)$, то $\alpha_k = \alpha_k^{(j)}$.

Когда можно обойтись без вычисления матрицы Гессе?

Пример: пусть $f = \|g(\mathbf{x}) - \mathbf{b}\|^2, g: \mathbb{R}^n \rightarrow \mathbb{R}^M, \mathbf{b} \in \mathbb{R}^M$.

Метод Гаусса-Ньютона: $\mathbf{p}_k = -(\nabla g(\mathbf{x}_k))^+(g(\mathbf{x}_k) - \mathbf{b}), \nabla g(\mathbf{x}_k) \in \mathbb{R}^{M \times n}$ — якобиан, $\mathbf{F}^+ = (\mathbf{F}^T \mathbf{F})^{-1} \mathbf{F}^T$ — псевдообращение Мура-Пенроуза.

На практике: методы *BFGS*, *L-BFGS-B* — используют приближение гессиана.

Возможные критерии остановки

- $\|\nabla f(\mathbf{x}_k)\| < \varepsilon$
- $\|\mathbf{x}_k - \mathbf{x}_{k-1}\| < \varepsilon$
- $\left\| \frac{f(\mathbf{x}_k) - f(\mathbf{x}_{k-1})}{f(\mathbf{x}_k)} \right\| < \varepsilon$

+ ограничение на число итераций! (чтобы метод не работал бесконечно долго, если он “застрял”).

Вопрос: а если мы используем численное вычисление градиента, то надо ли нам тогда обратить внимание на ε ?

Методы на выбор для семестрового задания

- 1 Покоординатный спуск
- 2 Градиентный спуск
- 3 Сопряженные градиенты Fletcher-Reeves
- 4 Сопряженные градиенты Polak-Ribiere
- 5 Ньютон (поиск по направлению)
- 6 Ньютон (backtracking)
- 7 Nelder-Mead (метод деформируемого многогранника)