Методы оптимизации

Лектор: Андреев Юрий Александрович

_scarleteagle **imkochelorov** **smvfe**

asparagus_densiflorus

Обусловленность (Холмы и овраги)

Градиентный спуск и поиск шага

В общем виде градиентный спуск выглядит так:

$$x_{k+1} = x_k + h(k)u,$$

где h — *learning rate* (шаг), u — вектор направления.

Причём $h(k)$ необязательно должен зависеть от k, номера итерации. Он может быть константен, так как градиент сам по себе уменьшается по мере приближения к минимуму.

Антиградиент является скорейшим направлением убывания, поэтому обычно выбирается именно он:

$$x_{k+1} = x_k - h(k)\nabla f(x_k)$$

> Есть вариант спуска, при котором мы нормируем антиградиент (тогда вектор направления действительно будет иметь единичную норму):
>
> $$x_{k+1} = x_k - h(k)\frac{\nabla f(x_k)}{\|\nabla f(x_k)\|}$$
>
> Но в таком варианте, $h(k)$ точно должно зависеть от итерации, и стремиться к 0 по k, так как отношение градиента к его норме будет в пределах константы.
>
> *В литературе, обычно, антиградиент не нормируется*

На прошлом занятии мы обсудили неточные методы поиска шага при градиентном спуске. Рассмотрим два других подхода: наискорейший градиентный спуск и планирование шага.

Наискорейший градиентный спуск (*steepest GD*)

$$x_{k+1} = x_k - h^* \nabla f(x_k)$$

$$h^* = \underset{h}{\operatorname{argmin}} f(x_k - h\nabla x_k)$$

Замечание: $\underset{h}{\operatorname{argmin}} f$ — такое значение h, при котором функция f достигает минимального значения. Например,

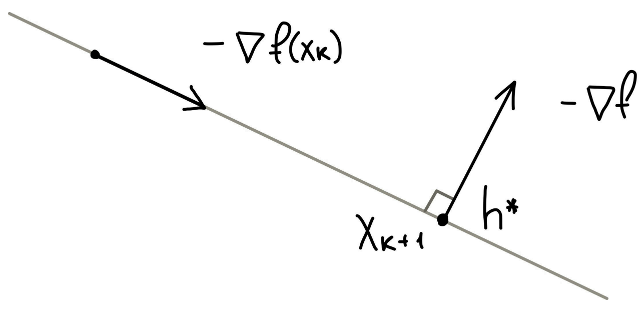
$$\min(x^2+1) = 1$$

$$\operatorname{argmin}(x^2+1) = 0$$

То есть в рамках большой задачи на минимизацию нам нужно решить маленькую задачу на минимизацию.

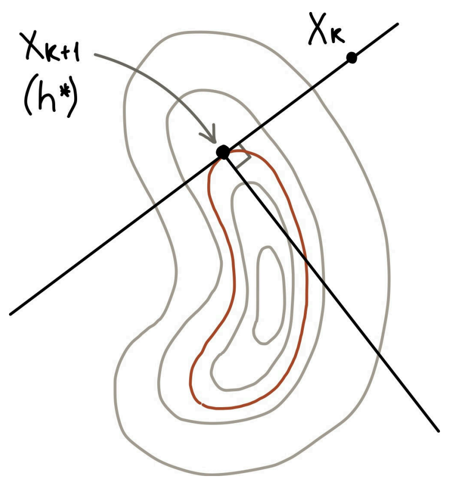
Проиллюстрируем:

Идём от x_k по направлению антиградиента. Когда мы сделаем шаг h^*, мы попадём в точку, где производная по направлению равна 0, то есть проекция градиента по этому направлению равна 0, а сам градиент в точке будет перпендикулярен направлению.



После каждого шага, направление градиентного спуска будет меняться на 90°

Если мы пользуемся обычным шагом, то мы на какое-то расстояние идём, потом снова ищем градиент, снова идём куда-то еще.



Если мы используем наискорейший градиентный спуск, мы остановимся в точке, где линия уровня касается луча направления, и поворачиваемся на 90°. Найти эту точку можно, например, методом дихотомии или золотого сечения.

Планирование шага (*scheduling*)

Замечание: k — номер итерации шага

Методы планирования шага:

1. Постоянный:

$$h(k) = h_0$$

2. Кусочно-постоянный:

$$h(k) = h_i$$

$$k_i \le k < k_{i+1}$$

$$\text{Пример: } h(i) = \frac{h_{i-1}}{2}$$

Идея: держим шаг постоянным до тех пор, пока не определим, что мы застряли в одном месте, и не уменьшим шаг.

3. Функциональный:

Exponential decay (экспоненциальное затухание):

$$h(k) = h_0 \cdot e^{-\lambda k}$$

Polynomial decay (полиномиальное затухание):

$$h(k) = h_0(\beta k + 1)^{-\alpha}$$

$$h_0 = \frac{1}{\sqrt{k+1}}$$

Здесь $\alpha, \beta, \lambda > 0$ — гиперпараметры. Рекомендуемые значения — $\alpha := \frac{1}{2}, \beta := 1$, но можно пробовать другие. Подбор гиперпараметров является вычислительно тяжёлой задачей, так как одно измерение — это один запуск алгоритма оптимизации.

Критерии остановки

- Идеал:

$$\|x_k - x^*\| < \varepsilon,\ |f(x_k) - f^*| < \varepsilon$$

$$f^* = \min f(x)$$
$$x^* = \operatorname{argmin} f(x)$$

Хоть мы и не знаем значения этих норм, иногда мы их можем оценивать сверху

Абсолютные варианты:

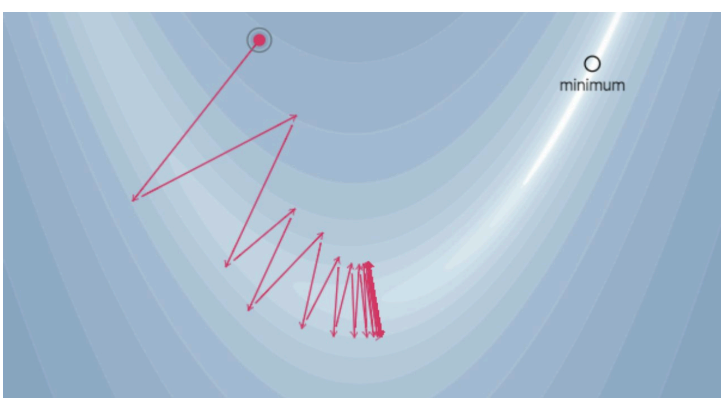
- $\|x_{k+1} - x_k\| < \varepsilon$
- $|f(x_{k+1}) - f(x_k)| < \varepsilon$
- $\underbrace{\|\nabla f(x_k)\|^2}_{\approx f(x_k) - f^*} < \varepsilon$

Относительные варианты (поправка на масштаб):

- $\|x_{k+1} - x_k\| < \varepsilon(\|x_{k+1}\| + 1)$
- $|f(x_{k+1}) - f(x_k)| < \varepsilon(|f(x_{k+1}) + 1|)$
- $\|\nabla f(x_k)\|^2 < \varepsilon \|\nabla f(x_0)\|^2$

- Контроль числа итераций метода

<u>Обусловленность</u>



Пример плохо обсуловленной функции

Если функция имеет линии уровня, формирующие овраги, как на картинке, пытаясь применить к ней метод градиентного спуска, мы начинаем ходить зигзагами из стороны в сторону, очень медленно приближаясь к минимуму функции или не приближаясь к нему вовсе.

Пример: квадратичная форма с минимумом в точке $(0, 0)$

$f(x) = 0.01x_1^2 + 10x_2^2$

$f(x) = x^T Ax + Bx + C$, где A — симметричная положительно определённая матрица, $Bx + C$ — параллельный перенос

$A = \begin{pmatrix} 0.01 & 0 \\ 0 & 10 \end{pmatrix}$

Если $f(x)$ — произвольная, тогда мы будем рассматривать <u>гессиан</u> (*Hessian*) функции.

$\underbrace{\text{condition number}}_{\text{число обусловленности}} = \frac{\lambda_{\max}}{\lambda_{\min}}; \quad \lambda_{\max}, \lambda_{\min}$ — собственные числа матрицы.

Если большое, то задача плохо обусловленная (*ill-conditioned*)

Можно доказать, что скорость сходимости нашего метода зависит от числа обусловленности: чем число ближе к 1, тем скорость больше.