

# Optimization and Computational Linear Algebra for Data Science

## Lecture 12: Gradient descent

Léo MIOLANE · leo.miolane@gmail.com

November 22, 2019

**Warning:** *This material is not meant to be lecture notes. It only gathers the main concepts and results from the lecture, without any additional explanation, motivation, examples, figures...*

In these notes,  $f$  denotes a twice differentiable **convex** function from  $\mathbb{R}^n$  to  $\mathbb{R}$ .

## 1 Gradient descent

Given an initial point  $x^{(0)} \in \mathbb{R}^n$ , the gradient descent algorithm follows the updates:

$$x^{(t+1)} = x^{(t)} - \alpha_t \nabla f(x^{(t)}), \quad (1)$$

where the step-size  $\alpha_t$  remains to be determined. The step (1) is a very natural strategy to minimize  $f$ , since  $-\nabla f(x)$  is the direction of steepest descent at  $x$ . Since  $f(x+h) = f(x) + \langle \nabla f(x), h \rangle + o(\|h\|)$  we have

$$\begin{aligned} f(x^{(t+1)}) &= f(x^{(t)}) - \alpha_t \|\nabla f(x^{(t)})\|^2 + o(\alpha_t) \\ &< f(x^{(t)}) \end{aligned}$$

for  $\alpha_t$  small enough (provided that  $\nabla f(x^{(t)}) \neq 0$ ). Hence if the step-sizes  $\alpha_t$  are chosen very small, the sequence  $(f(x^{(t)}))_{k \geq 0}$  is decreasing! However, if  $\alpha_t$  are too small, the algorithm may never converge.

### 1.1 Convergence analysis

**Notation:** Given a symmetric matrix  $M$  we will denote by  $\lambda_{\min}(M)$  and  $\lambda_{\max}(M)$  the smallest and largest eigenvalues of  $M$ .

#### Definition 1.1

For  $L, \gamma > 0$ , we say that a twice-differentiable convex function  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  is

- $L$ -smooth if for all  $x \in \mathbb{R}^n$ ,  $\lambda_{\max}(H_f(x)) \leq L$ .
- $\gamma$ -strongly convex if for all  $x \in \mathbb{R}^n$ ,  $\lambda_{\min}(H_f(x)) \geq \gamma$ .

#### Theorem 1.1

Assume that  $f$  is  $L$ -smooth and that  $f$  admits a (global) minimizer  $x^* \in \mathbb{R}^n$ . Then the gradient descent iterates (1) with constant step-size  $\alpha_k = 1/L$  verify

$$f(x^{(t)}) - f(x^*) \leq \frac{2L\|x^{(0)} - x^*\|^2}{t+4}.$$

**Why did we used step sizes of  $1/L$  ?** If  $f$  is  $L$ -smooth one can prove (see Homework 9) that for all  $x, h \in \mathbb{R}^n$ :

$$f(x+h) \leq f(x) + \langle \nabla f(x), h \rangle + \frac{L}{2} \|h\|^2. \quad (2)$$

Then, one can check (exercise!) that when  $x$  is fixed, the minimum of the right-hand side is minimum for  $h = -\frac{1}{L} \nabla f(x)$ .

### Theorem 1.2

Assume that  $f$  is  $L$ -smooth and  $\gamma$ -strongly convex. Then  $f$  admits a unique minimizer global  $x^*$  and the gradient descent iterates (1) with constant step-size  $\alpha_k = 1/L$  verify

$$f(x^{(t)}) - f(x^*) \leq \left(1 - \frac{\gamma}{L}\right)^t (f(x^{(0)}) - f(x^*)).$$

**Remark 1.1.** The  $\gamma$ -strong convexity of  $f$  implies that for all  $x \in \mathbb{R}^n$ ,

$$\frac{\gamma}{2} \|x - x^*\|^2 \leq f(x) - f(x^*).$$

Combining this with Theorem 1.2 gives a bound of the distance to the minimizer  $x^*$ :

$$\|x^{(t)} - x^*\|^2 \leq \frac{2}{\gamma} \left(1 - \frac{\gamma}{L}\right)^t (f(x^{(0)}) - f(x^*)).$$

**Proof.** Let  $t \geq 0$ . Applying (2) for  $x = x^{(t)}$  and  $h = x^{(t)} - L^{-1} \nabla f(x^{(t)})$ , we get

$$f(x^{(t+1)}) \leq f(x^{(t)}) - \frac{1}{L} \|\nabla f(x^{(t)})\|^2 + \frac{1}{2L} \|\nabla f(x^{(t)})\|^2 = f(x^{(t)}) - \frac{1}{2L} \|\nabla f(x^{(t)})\|^2.$$

Now, since  $f$  is  $\gamma$ -strongly convex, we have for all  $x \in \mathbb{R}^n$

$$f(x) - f(x^*) \leq 2\gamma \|\nabla f(x)\|^2.$$

We get that  $f(x^{(t+1)}) \leq f(x^{(t)}) - \frac{\gamma}{L} (f(x^{(t)}) - f(x^*))$ , hence

$$f(x^{(t+1)}) - f(x^*) \leq \left(1 - \frac{\gamma}{L}\right) (f(x^{(t)}) - f(x^*)),$$

from which the theorem follows. □

## 1.2 Choosing the step size in practice

In practice, one may not have access to  $L$  and need hence to choose the step size  $\alpha_t$ . A popular method is the so-called “backtracking line search” a goes as follows. Fix a parameter  $\beta \in (0, 1)$ . Start with  $\alpha = 1$  and while

$$f(x^{(t)} - \alpha \nabla f(x^{(t)})) > f(x^{(t)}) - \frac{\alpha}{2} \|\nabla f(x^{(t)})\|^2,$$

update  $\alpha = \beta \alpha$ . Then choose  $\alpha_t = \alpha$ .

## 2 Newton's method

We assume here that  $f$  is  $\gamma$ -strongly convex and  $L$ -smooth. Newton's method performs updates according to

$$x^{(t+1)} = x^{(t)} - H_f(x^{(t)})^{-1} \nabla f(x^{(t)}). \quad (3)$$

The (important!) difference with gradient descent is that the step-size  $\alpha_k$  is now replaced by the inverse<sup>1</sup> of the Hessian of  $f$ . The idea begin Newton's method is to minimize the second order approximation of  $f$  at  $x^{(t)}$  :

$$f(x^{(t)} + h) \simeq f(x^{(t)}) + \langle \nabla f(x^{(t)}), h \rangle + \frac{1}{2} h^\top H_f(x^{(t)}) h \quad (4)$$

with respect to  $h$  and then choose  $x^{(t+1)} = x^{(t)} + h$ . It is an easy exercise to see that the minimizer of the right-hand side of (4) is  $h = -H_f(x^{(t)})^{-1} \nabla f(x^{(t)})$ , leading to the recursion (5).

It can be shown (see for instance [1]) that for  $t$  large enough

$$\|x^{(t)} - x^*\|^2 \leq C e^{-\rho 2^t}, \quad (5)$$

where  $C, \rho$  are constants depending on  $f$  and  $x^{(0)}$ . We say that Newton's method converges *quadratically* to the minimizer  $x^*$ . Newton's method is much faster than gradient descent, whose speed (given by Theorem 1.2) is of order  $C' e^{-\rho' t}$ .

The main drawback of Newton's method is its computational complexity. Each step of the method require to compute the inverse of the  $n \times n$  Hessian matrix of  $f$  at  $x^{(t)}$ , which require  $O(n^3)$  operations. This makes Newton's method unpractical for large scale applications.

## Further reading



## References

- [1] Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, <https://web.stanford.edu/~boyd/cvxbook/>, 2004.

---

<sup>1</sup>The Hessian of  $f$  is indeed invertible at all  $x$  since its smallest eigenvalue is always greater than  $\gamma > 0$ .