OPTIMIZATION FOR DATA SCIENCE

November 10, 2025

Today (Session 3/8, next session Dec. 8)

- Theory of gradient methods
- Numerical illustration(s)
- Exercise (time permitting)

THEORY OF GRADIENT METHODS

Key idea: What can we prove about (variants of) gradient descent?

(1) Gradient descent

Problem (P) $\min_{w \in \mathbb{R}^d} f(w)$

$f : \mathbb{R}^d \to \mathbb{R}$

C^1 (= at every point $w \in \mathbb{R}^d$, the gradient $\nabla f(w) \in \mathbb{R}^d$ exists and it represents the way f changes around w)

Recall: First-order necessary conditions

$\bar{w}$ is a local minimum of (P) $\Rightarrow$ $\nabla f(\bar{w}) = 0_{\mathbb{R}^d}$

local minimum: $\exists \varepsilon > 0$, $\forall w \in \mathbb{R}^d$, $\|w - \bar{w}\| \leq \varepsilon$, $f(\bar{w}) \leq f(w)$

When f is convex, we have a stronger property

$\bar{w}$ is a global minimum of (P) $\Leftrightarrow$ $\nabla f(\bar{w}) = 0_{\mathbb{R}^d}$

global minimum: $f(\bar{w}) \leq f(w)$ $\forall w \in \mathbb{R}^d$

$\bar{w} \in \operatorname{argmin}_{w \in \mathbb{R}^d} f(w)$

$\nabla f(\bar{w}) = 0_{\mathbb{R}^d}$ $\Leftrightarrow$ $[\nabla f(\bar{w})]_j = 0$ $\forall j = 1 \dots d$ $\Leftrightarrow$ $\|\nabla f(\bar{w})\| = 0$

↳ In general, finding points with zero gradient is not possible directly $\Rightarrow$ use iterative algorithms instead

iterative algorithms: $\{w_0, w_1, w_2, \dots\}$

↳ Idea behind gradient descent

$\nabla f(w) \neq 0_{\mathbb{R}^d} \Rightarrow w$ not a local minimum

- If $\nabla f(w) \neq 0_{\mathbb{R}^d}$, then the function decreases locally in the direction $-\nabla f(w)$ → direction in which the function decreases the most locally around w
- Mathematically, there exists $\alpha > 0$ such that

$$f(w - \alpha \nabla f(w)) < f(w)$$

$f(w + \alpha d)$ is approximately $f(w) + \alpha \nabla f(w)^T d$ around w

Gradient descent (pseudo-code)

Initialization: Choose $w_0 \in \mathbb{R}^d$

Iteration k: Pick $\alpha_k > 0$.

Set $w_{k+1} = w_k - \alpha_k \nabla f(w_k)$ ← Gradient descent iteration

Implementation

- Stop the method after exhausting a budget (of iterations, of CPU/GPU time, calls to f or ∇f, ...)

what we did in the notebook in session 2

- Consider a convergence criterion, e.g. $\|\nabla f(w_k)\| \leq 10^{-10}$

$\|w_{k+1} - w_k\| \leq 10^{-15}$

$|f(w_{k+1}) - f(w_k)| \leq 10^{-10} |f(w_k)|$

Q: Suppose we run gradient descent for K iterations. What can we guarantee about the algorithm? How do we pick $\{\alpha_k\}_{k=0,\dots,K-1}$ (stepsizes)?

② Theory for constant stepsizes

Assumption: f is $C^{1,1}_L$ = "C^1 with L-Lipschitz continuous gradient"

- f is C^1
- ∇f is L-Lipschitz continuous:

$$\forall (v,w) \in (\mathbb{R}^d)^2, \quad \|\nabla f(v) - \nabla f(w)\| \leq L\|v-w\|$$

"The gradients cannot differ too much if the points v and w are close"

NB: The class of $C^{1,1}_L$ functions is the simplest one for which we can prove convergence rates for gradient descent, but not the only one.

Examples:

- Linear regression $f(w) = \frac{1}{2n}\|Xw - y\|^2$, $X \in \mathbb{R}^{n\times d}$, $y \in \mathbb{R}^n$

$$f \in C^{1,1}_L \text{ with } L = \frac{\|X^TX\|}{n}$$

Can compute this explicitly (done in the notebook)

- Logistic regression $f(w) = \frac{1}{n}\sum_{i=1}^n \log(1 + e^{-y_i x_i^T w})$, $x_i \in \mathbb{R}^d$, $y_i \in \{-1,1\}$

$$f \in C^{1,1}_L \text{ with } L = \frac{\|X^TX\|}{4n} \qquad X = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}$$

- Quadratic function $f(w) = \frac{1}{2}w^TCw$, $C \geq 0$

$f \in C^{1,1}_L$ with L largest eigenvalue of C

Proposition ("Descent lemma")

Suppose f is $C^{1,1}_L$. Let $w \in \mathbb{R}^d$ be a point such that $\nabla f(w) \neq 0_{\mathbb{R}^d}$.

Then, for any $\alpha \in \left(0, \frac{2}{L}\right)$, we have

$$f\left(w - \alpha \nabla f(w)\right) \leq f(w) - \left(\alpha - \frac{L\alpha^2}{2}\right) \|\nabla f(w)\|^2$$

($\alpha - \frac{L\alpha^2}{2} > 0$ when $\alpha \in (0, \frac{2}{L})$; $\|\nabla f(w)\|^2 > 0$ by assumption)

$$< f(w)$$

and using $\alpha = \frac{1}{L}$ gives the best decrease guarantee:

$$f\left(w - \frac{1}{L}\nabla f(w)\right) \leq f(w) - \frac{1}{2L}\|\nabla f(w)\|^2$$

↳ Consequences on implementation

- Suggests that choosing $\alpha_k = \alpha \in (0, \frac{2}{L})$ leads to convergence
- Suggests that using $\alpha_k = \frac{1}{L}$ $\forall k$ should give the best results

→ On our quadratic example, we saw that $\alpha < \frac{2}{L} = 2$ was a good choice, but slow for small values of α.

We also saw that $\alpha = \frac{1}{L} = 1$ was a good choice, but not necessarily the best.

Takeaway: $\alpha_k = \frac{1}{L}$ $\forall k$ is a good stepsize choice, but not necessarily the best

NB: To use $\frac{1}{L}$ as a stepsize, need L!

Other stepsize choices (besides constant stepsizes)

- Decreasing stepsizes: $\alpha_k > 0$, $\alpha_k \searrow 0$ $(\alpha_{k+1} \leq \alpha_k)$

  (ex) $\alpha_k = \frac{\alpha_0}{k+1}$, $\alpha_k = \frac{\alpha_0}{\sqrt{k+1}}$, ...

  ⊕ $\alpha_k < \frac{2}{L}$ for sufficiently large k $\Rightarrow$ Guarantees convergence

  ⊖ If α_k decreases too quickly, convergence will be slow

- Adaptive stepsizes: Compute α_k according to $f(w_k)$, $\nabla f(w_k)$ and possibly more information

Classical example: Backtracking line search

$(w_k, \nabla f(w_k) \neq 0_{\mathbb{R}^d})$

Compute $\alpha_k > 0$ as the largest value in $\{1, \frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \dots\}$ such that

$$f(w_k - \alpha_k \nabla f(w_k)) < f(w_k) - c\alpha_k \|\nabla f(w_k)\|^2$$

$c \in (0, \frac{1}{2}]$

Try $\alpha_k = 1$. If it doesn't work, try $\alpha_k = 1/2$. ...

↑ Approximation of the descent lemma (that does not require to know L)

NB: $c = 1/2$, $\alpha_k = \frac{1}{L}$ works

⊕ L not needed, backtracking always returns a value (worst case $\alpha_k < \frac{2}{L}$)

can have large stepsizes (e.g. 1)

⊖ Need to evaluate f (not needed by other implementations of gradient descent)

possibly multiple times per iteration

(3) Convergence rates for gradient descent

Setup: minimize $f(w)$ w.r.t. $w \in \mathbb{R}^d$

$f \in C_L^{1,1}$

$f(w) \geq f_{low}$ $\forall w$, where $f_{low} \in \mathbb{R}$

"if bounded below"

↳ We run K iterations of gradient descent ($K \geq 1$) starting from $w_0 \in \mathbb{R}^d$ and using $\alpha_0 = \alpha_1 = \dots = \alpha_{K-1} = \frac{1}{L}$.

a) f nonconvex

We can show that

$$\min_{0 \leq k \leq K-1} \|\nabla f(w_k)\| \leq \frac{2L(f(w_0) - f_{low})}{\sqrt{K}} = O\left(\frac{1}{\sqrt{K}}\right)$$

"of order $\frac{1}{\sqrt{K}}$"

minimum gradient norm over $w_0, \dots, w_{K-1}$

→ The inequality means that after K iterations, there exists an iterate with gradient norm smaller than $O\left(\frac{1}{\sqrt{K}}\right)$

→ As $K \to +\infty$, $O\left(\frac{1}{\sqrt{K}}\right) \to 0$ ⇒ the method "converges" to a point with zero gradient

→ If the iteration budget increases from K to $100K$, the guarantee goes from $O\left(\frac{1}{\sqrt{K}}\right)$ to $O\left(\frac{1}{10\sqrt{K}}\right)$

⇒ with 100 times more iterations, get 10 times more accurate.

↑ small gradient norm

We say that the convergence rate of gradient descent in the nonconvex case is $\frac{1}{\sqrt{K}}$.

b) Convex case

In addition to $f \in C^{1,1}_L$, we assume that f is convex and that it has a minimum $w^* \in \operatorname{argmin}_{w \in \mathbb{R}^d} f(w)$

We define $f^* = f(w^*) = \min_{w \in \mathbb{R}^d} f(w)$

We can show that

$$f(w_K) - f^* \leq \frac{L \|w_0 - w^*\|^2}{2K} = O\left(\frac{1}{K}\right)$$

$f(w_K) - f^*$: optimality gap in terms of function value

Convergence rate of gradient descent in the convex case is $\frac{1}{K}$.

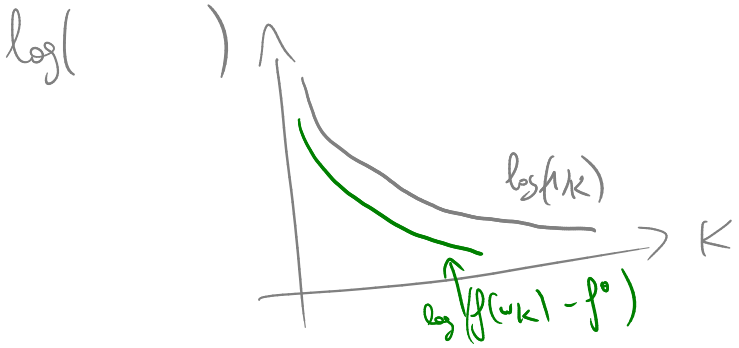
- $\frac{1}{K}$ decreases faster than $\frac{1}{\sqrt{K}}$ $\Rightarrow$ we say that gradient descent has a faster convergence rate on convex problems than on nonconvex problems

- $f(w_K) - f^*$ is a more interesting quantity for optimization than $\min_{0 \leq k \leq K-1} \|\nabla f(w_k)\|$ $\Rightarrow$ we say that gradient descent has stronger guarantees on convex problems than on nonconvex problems

NB: You can actually show $\min_{0 \leq k \leq K-1} \|\nabla f(w_k)\| \leq O\left(\frac{1}{K}\right)$ in the convex case

- In the convex case, we are guaranteed that after K iterations

the function value at the last iterate is within $O(\frac{1}{K})$ of the optimal value ($f^* \leq f(w_K) \leq f^* + O(\frac{1}{K})$)



(e.g. logistic regression plot in notebook)

c) Strongly convex f

We assume $f \in C^{1,1}_L$, μ-strongly convex $(0 < \mu \leq L)$

We let w^* be the minimum of f (it exists and it is unique)

and $f^* = f(w^*) = \min_{w \in \mathbb{R}^d} f(w)$

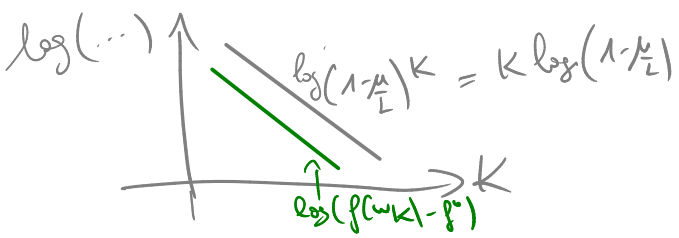
Then,

$$f(w_K) - f^* \leq \left(1 - \frac{\mu}{L}\right)^K \left(f(w_0) - f^*\right) = O\left(\left(1-\frac{\mu}{L}\right)^K\right)$$

$\in (0,1)$

We say that gradient descent converges at a rate $\underbrace{\left(1-\frac{\mu}{L}\right)^K}_{t^K \quad t \in (0,1)}$ in the μ-strongly convex case

- $\left(1-\frac{\mu}{L}\right)^K$ decreases (much) faster than $\frac{1}{K}$ $\Rightarrow$ the convergence rate of gradient is faster in the strongly convex case



(see linear regression plot in notebook)

Takeaways

- Convergence rates: For the same algorithm,

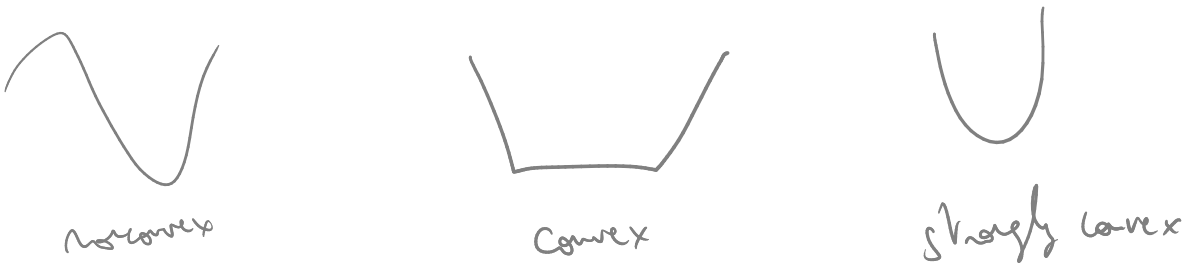
$O\left(\frac{1}{\sqrt{k}}\right)$ nonconvex f — $\min_{0 \le i \le k-1} \|\nabla f(w_i)\|$

$O\left(\frac{1}{k}\right)$ convex f

$O\left(\left(1-\frac{\mu}{L}\right)^k\right)$ μ-strongly convex f

$f(w_k) - f^*$

- Illustrates that strongly convex problems are easier to solve than convex problems, that are easier to solve than nonconvex problems.



(4) Advanced gradient descent

a) In the convex case / strongly convex case

Q) Are the rates for gradient descent the best possible?

A) No. Late 1970s: Within the class of algorithms that use 1 gradient per iteration (and nothing more), the best rates are $\frac{1}{k^2}$ for convex f

$\left(1-\sqrt{\frac{\mu}{L}}\right)^k$ for strongly convex f

A method with optimal convergence rates was found in 1983 by Nesterov: Nesterov's method / Accelerated gradient

For strongly convex quadratics, a method was proposed in 1964 by Polyak: Polyak's method / Heavy ball method.

Gradient descent:

$$w_{k+1} = w_k - \alpha_k \nabla f(w_k), \quad \alpha_k > 0$$

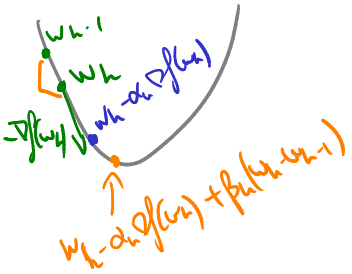
$w_k - w_{k-1} = -\alpha_{k-1} \nabla f(w_{k-1})$

Heavy ball method:

$$w_{k+1} = w_k - \alpha_k \nabla f(w_k) + \beta_k (w_k - w_{k-1}), \quad \alpha_k > 0, \ \beta_k > 0$$

Momentum step

$w_k - w_{k-1}$: previous step at iteration $k-1$



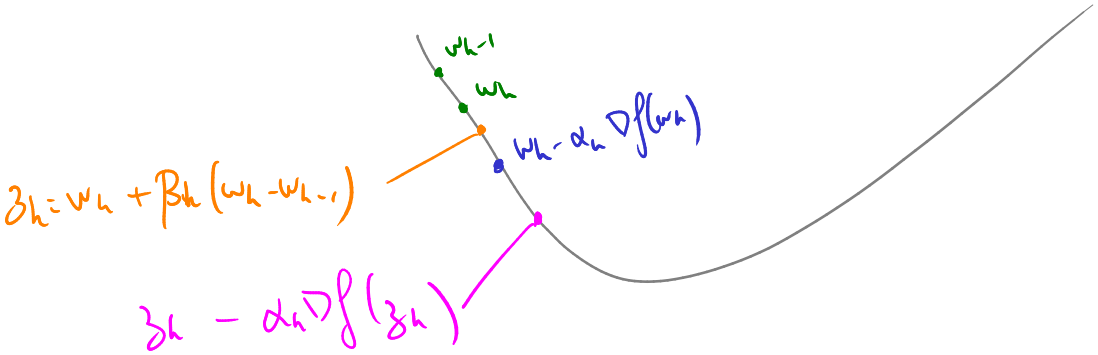
→ Achieves the best rate for strongly convex quadratic functions! *

→ But it can diverge for other convex functions

* $\alpha_k = \frac{1}{L}$, $\beta_k = \frac{\sqrt{L} - \sqrt{\mu}}{\sqrt{L} + \sqrt{\mu}}$

Accelerated gradient

$$w_{k+1} = w_k - \alpha_k \nabla f\left(w_k + \beta_k (w_k - w_{k-1})\right) + \beta_k (w_k - w_{k-1})$$



Convergence rates:

$$f(w_K) - f^* \leq \begin{cases} O\left(\frac{1}{K^2}\right) & \text{in the convex case} \\ O\left(\left(1 - \sqrt{\frac{\mu}{L}}\right)^K\right) & \text{in the strongly convex case} \end{cases}$$

b) In the nonconvex case

Gradient descent has rate $\frac{1}{\sqrt{K}}$: Best for $C^{1,1}_L$ functions !

Q) What does gradient descent converges to?

A point $\bar{v}$ such that $\nabla f(\bar{v}) = 0$ can be

- a local minimum
- a local maximum
- a saddle point

Informal Theorem (2015!)

Gradient descent almost always avoids saddle points and maxima