OPTIMIZATION FOR MACHINE LEARNING

IASD/MASH - October 17, 2024

Today (Session 9/16): Subgradient methods

Homework: Currently in discussion

$\Rightarrow$ You will hear from us by November 4!

INTRO

→ So far we have seen (a lot of) gradient descent and its applications

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

GD iteration for minimizing a C^1 function f

stepsize > 0 (aka learning rate)

can be computed using automatic differentiation

→ The rest of the course: alternatives to GD

→ Today: what if the function is not C^1?
what if the function is not differentiable?

Examples:

- Hinge loss / ReLU activation: $h: t \mapsto \max(t, 0)$

$t < 0$: h is differentiable at t and $h'(0) = 0$

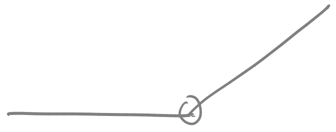
$t > 0$: ——— at t and $h'(t) = 1$

At $t = 0$, should we pick $h'(0) = 0$ or $h'(0) = 1$?

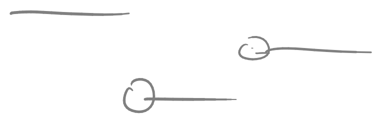
⇒ Actually, you can use any value between 0 and 1

- $x \mapsto \|x\|$ where $\|\cdot\|$ is any norm on $\mathbb{R}^d$ is not differentiable at $0_{\mathbb{R}^d}$ (that explains why we use $\|\cdot\|_2^2$ instead of $\|\cdot\|_2$ in linear regression $\quad \|x\|_2 = \sqrt{\sum_{i=1}^{d} x_i^2}$)

Terminology: We say that a function $f: \mathbb{R}^d \to \mathbb{R}$ is nonsmooth if there exists a point at which it is not differentiable.



ReLU
nonsmooth



nonsmooth
+ discontinuous

Nonsmooth functions are a broad class of functions

$\longrightarrow$ Focus: nonsmooth convex and continuous functions.

Key reference: Convex analysis (1970)
R.T. Rockafellar

1 Subgradients and subdifferentials

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

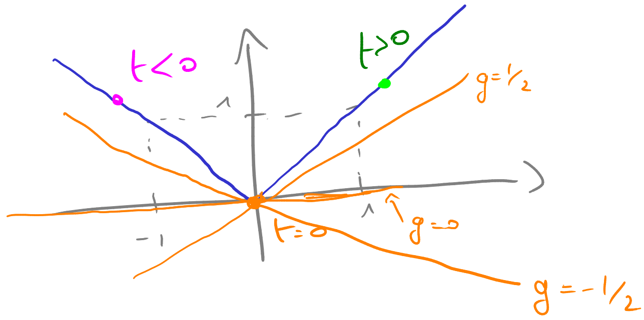
Definition: Let $x \in \mathbb{R}^d$. A vector $g \in \mathbb{R}^d$ is called a subgradient of f at x if

$$\forall y \in \mathbb{R}^d, \quad f(y) \geq f(x) + g^T(y-x)$$

The set of all subgradients of f at x is called the subdifferential of f at x, and denoted by $\partial f(x) \subseteq \mathbb{R}^d$

Ex) Consider $f : t \mapsto |t|$ in $\mathbb{R}$ $(d=1)$

f is not differentiable at 0.



$$f(t) = \begin{cases} t & \text{if } t \geq 0 \\ -t & \text{if } t \leq 0 \end{cases}$$

If $t > 0$:

$$\underset{\in \mathbb{R}}{g} \in \partial \underbrace{f(t)}_{\in \mathbb{R}} \iff \forall u \in \mathbb{R},\ f(u) \geq f(t) + g(u-t)$$

$$\iff \forall u \in \mathbb{R},\ |u| \geq |t| + g(u-t)$$

$$\iff \forall u \in \mathbb{R},\ |u| - t \geq g(u-t)$$

$$\iff \forall u \in \mathbb{R},\ \begin{cases} u - t \geq g(u-t) & \text{if } u \geq 0 \\ -u-t \geq g(u-t) & \text{if } u < 0 \end{cases}$$

$u = t \quad 0 \geq 0$

$-2t \geq g(-2t)$

$g \geq 1$

$u = 2t \quad t \geq gt \iff 1 \geq g$

$$\iff g = 1$$

$$\partial f(t) = \{1\} = \{f'(t)\}$$

Similarly, for any $t < 0$, $\partial f(t) = \{-1\} = \{f'(t)\}$

If $t = 0$,

$$g \in \partial f(0) \iff \forall u \in \mathbb{R},\ f(u) \geq \overbrace{f(0)}^{0} + g(u-0)$$

$$\iff \forall u \in \mathbb{R},\ |u| \geq gu$$

$$\iff \forall u \neq 0,\ \begin{cases} u \geq gu & \text{if } u > 0 \\ -u \geq gu & \text{if } u < 0 \end{cases}$$

$$\iff \begin{cases} 1 \geq g \\ -1 \leq g \end{cases}$$

Thus $\partial f(0) = [-1, 1]$

Properties of the subdifferential

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

(i) If f is differentiable at $x \in \mathbb{R}^d$, then $\partial f(x) = \{ \nabla f(x) \}$

(ii) By convexity, $\partial f(x)$ is always nonempty.

(iii) The converse of (i) is true: If $\partial f(x) = \{ g \}$, then f is differentiable at x. and $g = \nabla f(x)$

NB: For C^1 convex f, $f(y) \geq f(x) + \nabla f(x)^T (y - x) \quad \forall y \in \mathbb{R}^d$

↳ There exist calculus rules for the subdifferential

Subdifferential calculus

a) For any $f_1, f_2 : \mathbb{R}^d \to \mathbb{R}$ convex,

$$\forall x \in \mathbb{R}^d, \quad \partial(\underbrace{f_1 + f_2}_{\text{convex}})(x) = \underbrace{\partial f_1(x) + \partial f_2(x)}_{\{ g \in \mathbb{R}^d \mid g = g_1 + g_2,\ g_1 \in \partial f_1(x),\ g_2 \in \partial f_2(x) \}}$$

b) For any $f : \mathbb{R}^d \to \mathbb{R}$ convex, $\forall \alpha > 0$

$$\partial(\underbrace{\alpha f}_{\text{convex}})(x) = \underbrace{\alpha \partial f(x)}_{\{ \alpha g \mid g \in \partial f(x) \}}$$

c) For any $h : \mathbb{R}^m \to \mathbb{R}$ convex, $\forall A \in \mathbb{R}^{m \times d}$, $\forall b \in \mathbb{R}^m$,

let $f : \begin{array}{l} \mathbb{R}^d \to \mathbb{R} \\ x \mapsto h(Ax + b) \end{array}$

Then $\forall x \in \mathbb{R}^d$, $\partial f(x) = \underbrace{A^T \partial h(Ax+b)}_{\{A^T g \mid g \in \partial h(Ax+b)\}}$

NB: a) b) c) are generalizations of calculus rules for gradients/derivatives

d) Let $f : \begin{array}{l} \mathbb{R}^d \to \mathbb{R} \\ x \longmapsto \max\limits_{1 \le i \le m} f_i(x) \end{array}$ where $f_i : \mathbb{R}^d \to \mathbb{R}$ is convex

Then f is convex and

$$\partial f(x) = \operatorname{conv}\{ \partial f_i(x) : f_i(x) = f(x) \}$$

$$\operatorname{conv}(A) = \text{convex hull of } A$$
$$= \{ \alpha x + (1-\alpha) y \mid x \in A, y \in A, \alpha \in [0,1] \}$$

NB: Property d) has no equivalent for differentiable functions

Example: $f : x \longmapsto \max(a_1^T x + b_1, a_2^T x + b_2)$

$a_1 \in \mathbb{R}^d$
$a_2 \in \mathbb{R}^d$
$b_1 \in \mathbb{R}$, $b_2 \in \mathbb{R}$

NB: If $a_1 = a_2$, $f(x) = a_1^T x + \max(b_1, b_2)$

$a_1 \neq a_2$

$a_1 \neq a_2$ $\hookrightarrow$ f is not differentiable at any x for which

$$\underbrace{a_1^T x + b_1}_{f_1(x)} = \underbrace{a_2^T x + b_2}_{f_2(x)}$$

(If $a_1 = 1$, $a_2 = -1$, $b_1 = b_2 = 0$, $d = 1$, this is $|x|$!)

(If $a_1 = 1$, $b_1 = 0$, $a_2 = 0$, $b_2 = 0$, $d = 1$, this is $\mathrm{ReLU}(x) = \max(x, 0)$)

By the calculus rules,

$$\partial f(x) = \operatorname{conv}\{ \partial f_1(x), \partial f_2(x) \}$$
$$= \operatorname{conv}\{ a_1, a_2 \} = \{ \alpha a_1 + (1-\alpha) a_2 : \alpha \in [0,1] \}$$

Subgradients and automatic differentiation (AD)

↳ Much more tricky than computing gradients!

→ Subdifferentials are sets, but AD is designed to output one subgradient given an input

Q: which subgradient do you get?

→ The subgradient you get depends on the way $f(x)$ is encoded.

↳ Nice example (Bolte & Pauwels 2020)

"A mathematical model for automatic differentiation in machine learning"

$$\mathrm{ReLU}(t) = \max(t, 0)$$

$$\partial \mathrm{ReLU}(0) = [0, 1]$$

If $\mathrm{ReLU}(t)$ is encoded as $\max(-t, 0) + t$, then AD applied at 0 gives the subgradient $g = 1$

If $\mathrm{ReLU}(t)$ is encoded as $\frac{1}{2}\left(\max(t,0) + \max(-t,0) + t\right)$ then AD applied at 0 gives the subgradient $g = 0$

Remark For nonconvex functions:

→ The subdifferential can be empty ($t \mapsto -|t|$ at 0)

→ Another subdifferential is used: the Clarke subdifferential, that coincides with the subdifferential above when the function is convex

Exercise: Compute the subdifferential of

$$\|.\|_\infty : \mathbb{R}^d \to \mathbb{R}$$
$$x \mapsto \|x\|_\infty = \max_{1 \le i \le d} |x_i|$$

and

$$\|.\|_1 : \mathbb{R}^d \to \mathbb{R}$$
$$x \mapsto \|x\|_1 = \sum_{i=1}^{d} |x_i|$$

2 Subgradient methods

Setup: (P) $\underset{x \in \mathbb{R}^d}{\text{minimize}} \ f(x)$ $\qquad f : \mathbb{R}^d \to \mathbb{R}$ convex

Recall: when f C^1, we know that $x^* \in \operatorname{argmin} f(x)$

$\iff \nabla f(x^*) = 0_{\mathbb{R}^d}$

$\rightarrow$ Basis for GD:

If $\nabla f(x) \neq 0_{\mathbb{R}^d}$, f decreases in the direction of $-\nabla f(x)$

Q) How can we certify optimality or move towards a better point without gradients?

Theorem: Consider problem (P) and $x^* \in \mathbb{R}^d$.

$$\left[x^* \in \underset{x \in \mathbb{R}^d}{\operatorname{argmin}} \ f(x) \right] \iff \left[0_{\mathbb{R}^d} \in \partial f(x^*) \right]$$

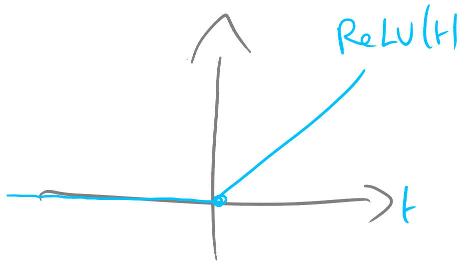
Optimality condition for convex nonsmooth problems

__Remark:__ If f C^1, this optimality condition reduces to

$$\nabla f(x^*) = 0_{\mathbb{R}^d}$$

$$0_{\mathbb{R}^d} \in \partial f(x^*)$$
$$\iff 0_{\mathbb{R}^d} \in \{\nabla f(x^*)\}$$
$$\iff 0_{\mathbb{R}^d} = \nabla f(x^*)$$

Ex) $\mathrm{ReLU}(t) = \max(t, 0)$



$\forall t < 0$, $\partial \mathrm{RELU}(t) = \{0\}$ contains 0

$\forall t > 0$, $\partial \mathrm{RELU}(t) = \{1\}$ does not contain 0

$\partial \mathrm{RELU}(0) = [0, 1]$ contains 0

$\underset{t \in \mathbb{R}}{\mathrm{argmin}}\ \mathrm{ReLU}(t) = \{t \leq 0\}$

$\hookrightarrow$ The optimality condition suggests that we can use the subdifferential in a way similar to the gradient in smooth optimization

Subgradient method (general form)

__Initialization__: $x_0 \in \mathbb{R}^d$

__Iteration k__: $x_{k+1} = x_k - \alpha_k g_k$, where $\alpha_k > 0$ and $g_k \in \partial f(x_k)$

__Key challenge__: Choosing the subgradient g_k!

$\rightarrow$ If f is differentiable at x_k, necessarily $g_k = \nabla f(x_k)$

$\Rightarrow$ GD iteration

$\rightarrow$ Otherwise, the method needs a way to choose g_k

- there are good theoretical choices
- but in practice g_k is fixed by code of f + AD tool

Ex) $f : \mathbb{R} \to \mathbb{R}$, $x \mapsto |x|$ $\quad x_0 = 0$

Any subgradient other than 0 ($g \in [-1,1]$, $g \neq 0$) is an ascent direction for f

$$\forall \alpha > 0, \quad f(x_0 - \alpha g) > f(x_0) = 0$$
$$\forall g \in \partial f(x_0), g \neq 0$$

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

$x \mapsto \|x\|_1 = \sum_{i=1}^{d} |x_i|$

$$\partial f\left(\underbrace{\begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}}_{e_1}\right) = \left\{ e_1 + \sum_{i=2}^{d} t_i e_i \;,\; t_i \in [-1,1] \right\}$$

$e_i = \begin{bmatrix} 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{bmatrix}$ ← i-th row

$f(e_1) = 1$

Any vector $g \in \partial f(e_1)$ with $g = e_1 + \sum_{i=2}^{d} t_i e_i$, $\sum_{i=2}^{d} |t_i| > 1$ is an ascent direction

$\alpha > 0$

$$f(e_1 - \alpha g) = |1-\alpha| + \sum_{i=2}^{d} \alpha |t_i|$$
$$> |1-\alpha| + \alpha$$

if $\alpha \in (0,1]$ $\quad f(e_1 - \alpha g) > 1 = f(e_1)$

$\alpha > 1$ $\quad f(e_1 - \alpha g) > \alpha - 1 + \alpha = 2\alpha - 1 > \alpha > 1 = f(e_1)$

Proposition: Let $f : \mathbb{R}^d \to \mathbb{R}$ convex and let $x \in \mathbb{R}^d$ such that $0_{\mathbb{R}^d} \notin \partial f(x)$.

Then, for any $g_{min} \in \underset{g \in \mathbb{R}^d}{\operatorname{argmin}} \left\{ \|g\|^2 \mid g \in \partial f(x) \right\}$ "minimum norm subgradient"

g_{min} defines a descent direction, in that

$$f(x - \alpha g_{min}) < f(x) \text{ for sufficiently small } \alpha.$$

↳ The proposition suggests to use $g_k \in \underset{g \in \mathbb{R}^d}{\operatorname{argmin}} \left\{ \|g\|^2 \mid g \in \partial f(x_k) \right\}$ in the subgradient method...

↳ ...but this involves solving an auxiliary optimization problem, which is not tractable in general, especially when there is no explicit formula for the subdifferential

↳ What can we prove when the subgradient method uses an arbitrary subgradient at every iteration?

→ f is not guaranteed to decrease at every iteration (unlike in GD, where we can guarantee that with a good stepsize)

→ The quantity of interest for convergence of subgradient methods cannot be $f(x_k) - \min_{x \in \mathbb{R}^d} f(x)$

Theorem Consider the subgradient method applied to (P) and suppose that x_k (kth iterate) is not a minimum ($\Leftrightarrow 0_{\mathbb{R}^d} \notin \partial f(x_k)$). Let $x^* \in \operatorname{argmin}_{x \in \mathbb{R}^d} f(x)$. Then for any $g_k \in \partial f(x_k)$, there exists $\alpha > 0$ such that

$$\underbrace{\|x_k - \alpha g_k - x^*\|^2}_{\text{distance between } x_k - \alpha g_k \text{ and } x^*} < \underbrace{\|x_k - x^*\|^2}_{\text{distance between } x_k \text{ and } x^*}$$

α depends on g_k, x_k and x^*

Proof For any $\alpha > 0$,

$$\|x_k - \alpha g_k - x^*\|^2 = \|(x_k - x^*) - \alpha g_k\|^2 = \|x_k - x^*\|^2 - 2\alpha g_k^T (x_k - x^*) + \alpha^2 \|g_k\|^2$$

Since $g_k \in \partial f(x_k)$, we have $f(y) \geq f(x_k) + g_k^T (y - x_k) \quad \forall y \in \mathbb{R}^d$

hence $f(x^*) \geq f(x_k) + g_k^T (x^* - x_k)$

$$\Leftrightarrow \quad -g_k^T (x_k - x^*) \leq f(x^*) - f(x_k)$$

Therefore,

$$\|x_k - \alpha g_k - x^*\|^2 \leq \|x_k - x^*\|^2 + 2\alpha \underbrace{\left(f(x^*) - f(x_k)\right)}_{< 0 \text{ because } x_k \text{ is not a minimum}} + \alpha^2 \|g_k\|^2$$

$$= \|x_k - x^*\|^2 - 2\alpha\left(f(x_k) - f(x^*)\right) + \alpha^2 \|g_k\|^2$$

For any $\alpha \in \left(0, \boxed{\frac{2\left(f(x_k) - f(x^*)\right)}{\|g_k\|^2}}\right)$, we have

$$-2\alpha\left(f(x_k) - f(x^*)\right) + \overbrace{\alpha^2\|g_k\|^2}^{\alpha\|g_k\|^2 \times \alpha} < -2\alpha\left(f(x_k) - f(x^*)\right) + \alpha\|g_k\|^2 \times \left(\frac{2\left(f(x_k) - f(x^*)\right)}{\|g_k\|^2}\right)$$

$$= -2\alpha\left(f(x_k) - f(x^*)\right) + 2\alpha\left(f(x_k) - f(x^*)\right)$$

$$= 0$$

Thus, for any $\alpha \in \left(0, \frac{2\left(f(x_k) - f(x^*)\right)}{\|g_k\|^2}\right)$,

$0 \notin \partial f(x_k)$

$$\|x_k - \alpha g_k - x^*\|^2 < \|x_k - x^*\|^2$$

In practice, one can use:

- predefined constant stepsizes during a fixed number of iterations
- predefined sequence of decreasing stepsizes

For both stepsize strategies, it is possible to obtain convergence rates for the subgradient method.

Theorem Suppose that $f : \mathbb{R}^d \to \mathbb{R}$ is convex, has a minimum and (for simplicity) that f has bounded subgradients, i.e.

$$\forall x \in \mathbb{R}^d, \ \forall g \in \partial f(x), \ \|g\| \le M < \infty$$

(true if f is M-Lipschitz continuous)

Let $x^* \in \operatorname{argmin}_{x \in \mathbb{R}^d} f(x)$

1) Fixed stepsize result

Run the subgradient method for $K \geq 1$ iterations with

$$\alpha_k = \frac{\|x_0 - x^*\|}{M\sqrt{K}}$$

theoretical value that works but you can also prove results for $\alpha_k = \theta/\sqrt{K}$

Then

$$f(\overline{x_K}) - f(x^*) \leq \frac{\|x_0 - x^*\| M}{\sqrt{K}}$$

where $\overline{x_K} = \frac{1}{K} \sum_{k=0}^{K-1} x_k$ (average iterate)

2) Decreasing stepsize

Run the subgradient method with $\alpha_k = \frac{\theta}{\sqrt{k+1}}$ $k \geq 0$ with $\theta > 0$

Then, after $K \geq 1$ iterations, we have

$$f(\overline{x_K}) - f(x^*) \leq O\left(\frac{\|x_0 - x^*\|^2}{\sqrt{K}} + \frac{M^2 \log K}{\sqrt{K}} \right)$$

where $\overline{x_K} = \frac{1}{\sum_{k=0}^{K-1} \alpha_k} \sum_{k=0}^{K-1} \alpha_k x_k$ (weighted average of the iterates)

Observations:

- The convergence rates apply to an average of the iterates, for which they show that this average behaves more smoothly than the actual iterates

$\Rightarrow$ Similar results for $\min_{0 \leq k \leq K-1} \left(f(x_k) - f(x^*) \right)$

"Best-iterate result"

(NB: Computing the best iterate requires to evaluate f, whereas the average can be computed directly from the iterates)

- The rates $O\left(\frac{1}{\sqrt{k}}\right)$ for fixed stepsize and
  $O\left(\frac{\log k}{\sqrt{k}}\right)$ for decreasing stepsize

  are worse than the rate of GD on a convex problem
  $O(1/k)$ $\Rightarrow$ nonsmooth convex minimization is harder than smooth convex minimization

- $O\left(\frac{1}{\sqrt{K}}\right)$ better than $O\left(\frac{\log K}{\sqrt{K}}\right)$

  but the first rate is obtained by fixing K and running the method with a stepsize that depends on K