Machine Learning and Optimization

Lecture 8: Subdifferentials and Nonsmooth Convex Optimization

CSCI 4140/6960 — Spring 2026

Course website: `cs.rpi.edu/~gittea/.../mlandopt.html`

Learning Objectives

By the end of this lecture, you should be able to:

1. Define extended-value convex functions, proper functions, and lower semicontinuous (lsc) functions, and explain why lsc is essential for optimization
2. Reformulate constrained convex problems as unconstrained problems using indicator functions
3. Define and compute subdifferentials for simple nonsmooth convex functions

Outline

Beyond Smooth Optimization

Extended-Value Functions and Lower Semicontinuity

Subdifferentials: Definition and Geometric Intuition

Computing Subdifferentials: Simple Examples

Summary and Looking Ahead

The Problem with Gradient Methods

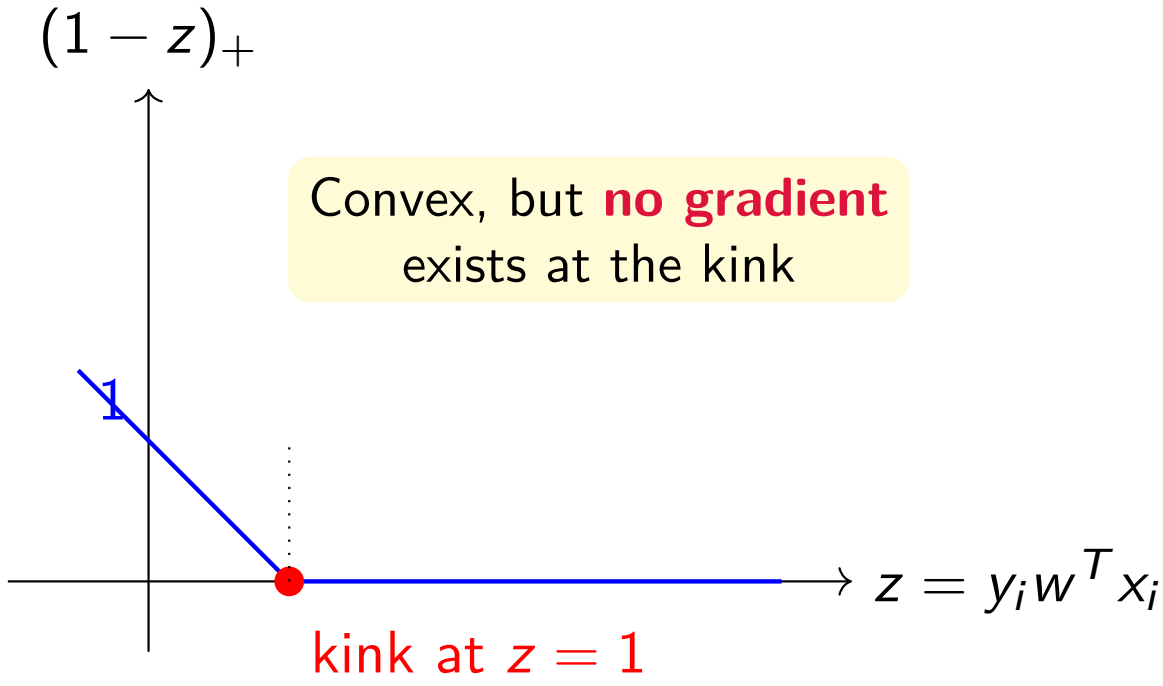
Classical gradient descent requires f to be differentiable.

Many important functions are **not differentiable** everywhere:

- $f(x) = |x|$ (absolute value)
- $f(x) = \|x\|_1$ (ℓ_1 regularization / LASSO)
- $f(w) = \frac{1}{n}\sum_{i=1}^{n}(1 - y_i w^T x_i)_+$ (SVM objective)

All of these are **convex** but have corners or kinks where no gradient exists.

Piecewise Linear Structure of the Hinge Loss



The SVM objective $f(w) = \frac{1}{n} \sum_{i=1}^{n} (1 - y_i w^T x_i)_+$ is a sum of such terms: piecewise linear with kinks at $y_i w^T x_i = 1$.

Motivating Questions and Preview

Question 1: How do we generalize the gradient to handle nondifferentiable convex functions?

Answer: The **subdifferential** $\partial f(x)$ replaces $\nabla f(x)$.

Necessary and sufficient optimality condition generalizes:

$$\textbf{Smooth:} \quad 0 = \nabla f(x^*) \quad \text{(single vector equation)}$$
$$\textbf{Nonsmooth:} \quad 0 \in \partial f(x^*) \quad \text{(set membership condition)}$$

Question 2: How do we handle **constrained** convex optimization problems?

$$\min_{x \in C} f(x) \quad \text{where } C \text{ and } f \text{ are convex}$$

Answer: use optimality conditions and algorithms developed for nondifferentiable convex functions!

What replaces differentiability?

When we drop differentiability, we lose several intuitions:

- **Existence:** Does a minimizer even exist? (Not always!)
- **Boundaries:** What happens at the "edge" of the function's domain?
- **Algorithms:** What replaces smoothness?

To build robust theory and algorithms, we need appropriate assumptions:

1. **Extended-value functions** (∞) to encode constraints as nondifferentiable objectives.
2. **Properness** to avoid triviality.
3. **Lower semicontinuity** to ensure objectives behave at the limit.

Extended-Value and Proper Functions

Definition

An **extended-value function** is a map $f : \mathbb{R}^d \to \mathbb{R} \cup \{\infty\}$.

The **effective domain** is:

$$\mathrm{dom}(f) = \{x : f(x) < \infty\}$$

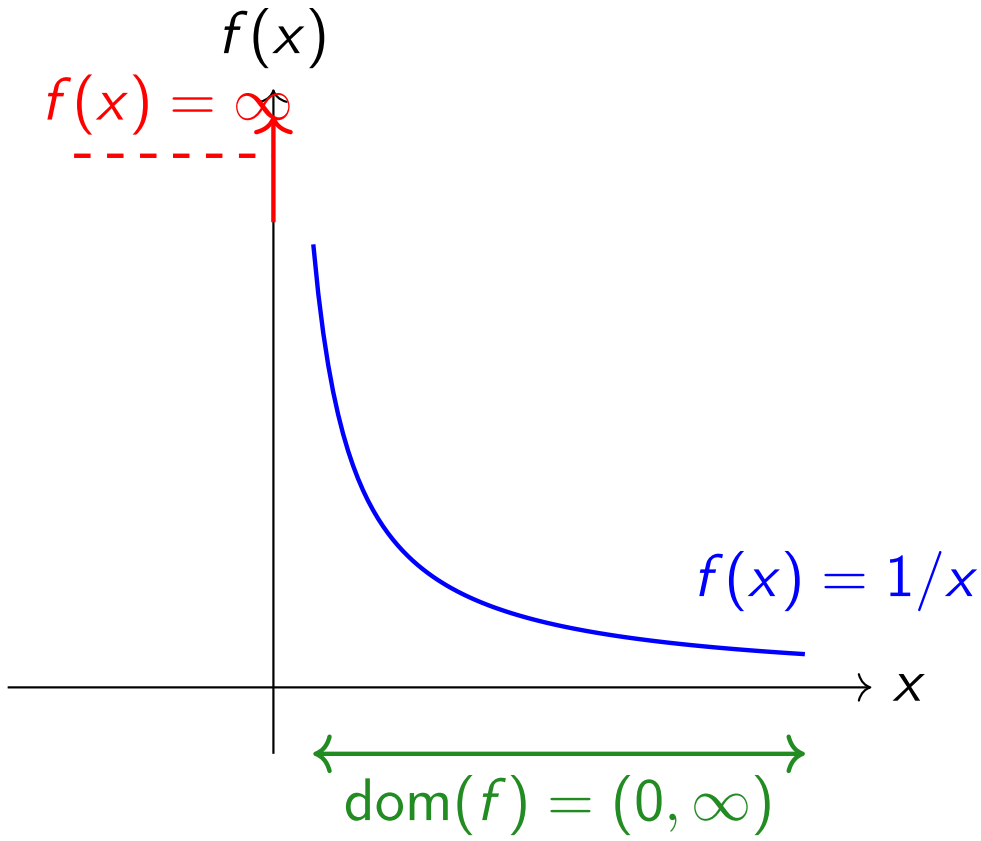
Definition

f is **proper** if:

1. $\mathrm{dom}(f) \neq \emptyset$ (finite somewhere)
2. $f(x) > -\infty$ for all x (never $-\infty$)

Why allow ∞? This lets us encode constraints implicitly, unifying constrained and unconstrained optimization.

Example: Extended-Value Function



$$f(x) = \begin{cases} 1/x & x > 0 \\ \infty & x \leq 0 \end{cases}$$ is a proper extended-value convex function.

Lower Semicontinuity (lsc)

Definition

$f : \mathbb{R}^d \to \mathbb{R} \cup \{\infty\}$ is **lower semicontinuous (lsc)** at x if:

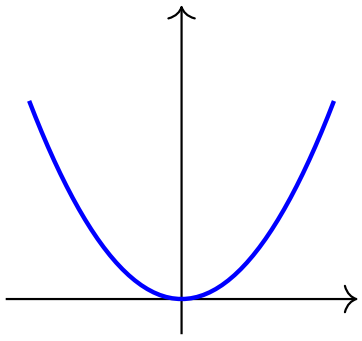
$$\liminf_{y \to x} f(y) \geq f(x)$$

Equivalently, for every sequence $x_n \to x$: $f(x) \leq \liminf_{n \to \infty} f(x_n)$.
Equivalently, the epigraph of f is closed (hence you will sometimes hear such functions referred to as "closed").

Intuition: The function value doesn't "jump up" at x. The graph has no upward discontinuities.

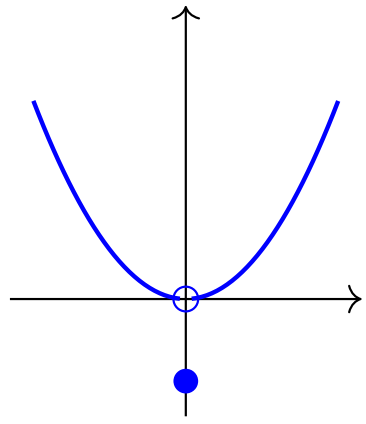
- Downward jumps are allowed (lsc)
- Upward jumps are **forbidden** (breaks lsc)

Visualizing Lower Semicontinuity



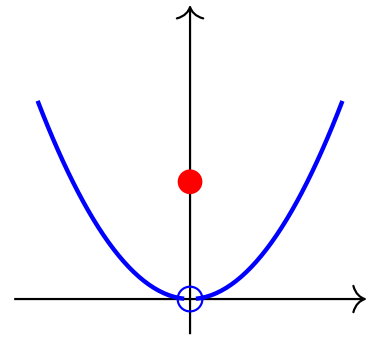
Continuous $\Rightarrow$ lsc

✓



Jumps down: lsc

✓



Jumps **up**: **not lsc**

×

Left: $f(x) = x^2$ (continuous). Middle: $f(0) = -1$, else x^2. Right: $f(0) = 1$, else x^2.

Why LSC Matters: Existence of Minimizers

Theorem (Existence of Minimizers)

If $f : \mathbb{R}^d \to \mathbb{R} \cup \{\infty\}$ is proper, lsc, and coercive ($f(x) \to \infty$ as $\|x\| \to \infty$), then f attains its minimum.

Why these assumptions are exactly what you need:

- Coercive $\Rightarrow$ sublevel sets $\{x : f(x) \leq c\}$ are bounded
- LSC $\Rightarrow$ sublevel sets are closed
- Closed + bounded in $\mathbb{R}^d \Rightarrow$ compact

Why compactness $\Rightarrow$ minimum is attained (proof sketch):
Let $L_c = \{x : f(x) \leq c\}$ and $m = \inf_{x \in L_c} f(x)$.

- Pick a minimizing sequence $x_n \in L_c$ with $f(x_n) \downarrow m$
- Compactness $\Rightarrow x_{n_k} \to x^\star \in L_c$
- LSC $\Rightarrow f(x^\star) \leq \liminf_k f(x_{n_k}) = m$
- Hence $f(x^\star) = m$

If *any* assumption fails, the infimum may exist but be unattainable.

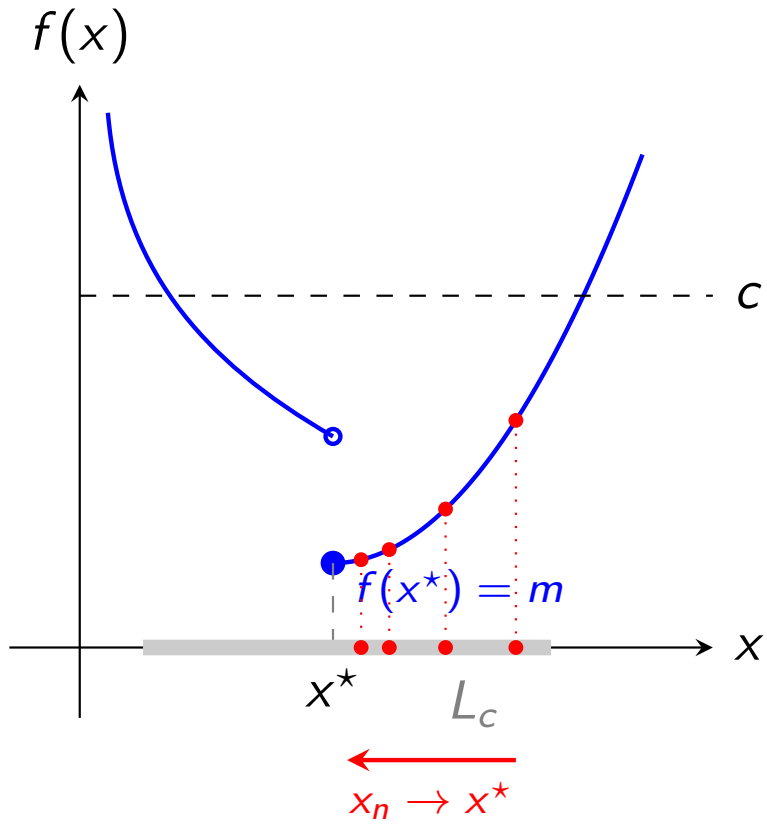
Proof Sketch: Compactness $\Rightarrow$ Minimum is Attained

Let $m = \inf_{x \in K} f(x)$.

1. **Sequence:**
   Pick $x_n \in L_c$ s.t. $f(x_n) \downarrow m$.
2. **Compactness:**
   L_c is compact $\Rightarrow$ a subsequence $x_{n_k} \to x^\star \in L_c$.
3. **LSC Property:**
   Value at limit is $\leq$ the limit of values:

   $$f(x^\star) \leq \liminf_{k \to \infty} f(x_{n_k}) = m$$

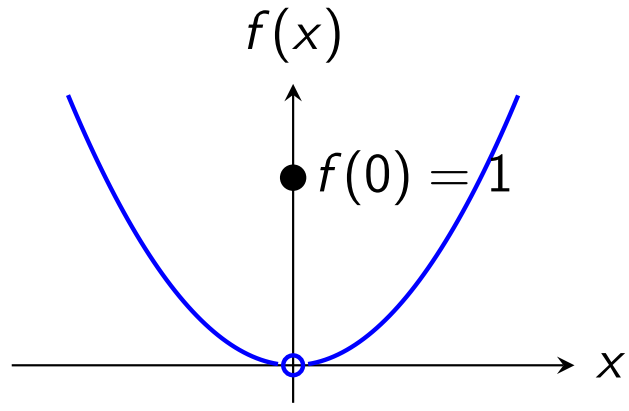
4. **Optimality:**
   Since m is the infimum, $f(x^\star) = m$.



Example: Non-LSC Function Without Minimizer

Consider:

$$f(x) = \begin{cases} x^2 & x \neq 0 \\ 1 & x = 0 \end{cases}$$



- $\inf_x f(x) = 0$ (as $x \to 0$)
- But $f(x) > 0$ for all x
- **No minimizer exists!**
- Not lsc at $x = 0$

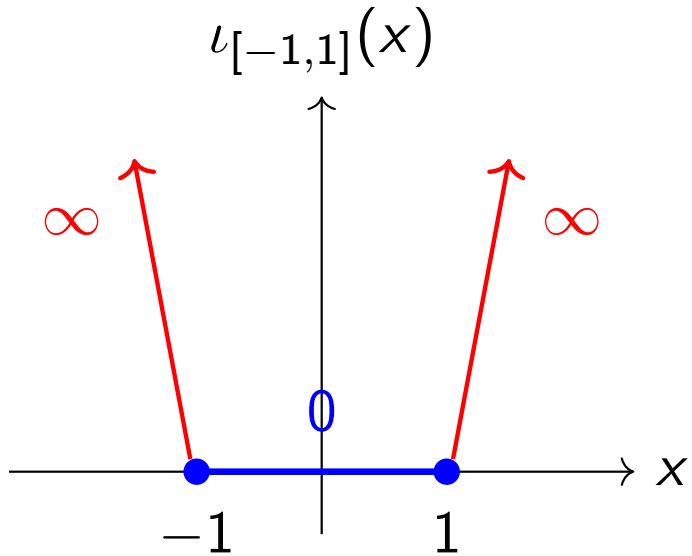
Numerical consequence: Any algorithm generating $x_n \to 0$ sees $f(x_n) \to 0$, but evaluating at the limit gives $f(0) = 1$. The infimum is not attained.

The Convex Indicator Function

For a convex set $C \subseteq \mathbb{R}^d$, define:

$$\iota_C(x) = \begin{cases} 0 & x \in C \\ \infty & x \notin C \end{cases}$$

Fact: ι_C is lsc if and only if C is closed.



Example: $\iota_{(0,1)}$ is NOT lsc — at $x = 0$, take $x_n = 1/n \to 0$ with $\iota_{(0,1)}(x_n) = 0$, but $\iota_{(0,1)}(0) = \infty$.

Constrained Optimization Using Indicator Functions

Original constrained problem:

$$\min_{x \in C} f(x)$$

Equivalent unconstrained problem:

$$\min_{x} \; f(x) + \iota_C(x)$$

where C is the feasible set.

Why is this useful?

1. **Unified framework:** Constrained and unconstrained have the same form
2. **Theoretical clarity:** Optimality conditions become cleaner
3. **Algorithmic simplicity:** Proximal/subgradient methods work with this form

Example: Reformulation

Original:

$$\min_x \ \frac{1}{2}\|Ax - b\|_2^2 + \lambda\|x\|_1 \quad \text{s.t.} \quad x \geq 0$$

Unconstrained reformulation:

$$\min_x \ \frac{1}{2}\|Ax - b\|_2^2 + \lambda\|x\|_1 + \iota_{\mathbb{R}^d_+}(x)$$

Now we have a single unconstrained (but nonsmooth!) minimization problem.

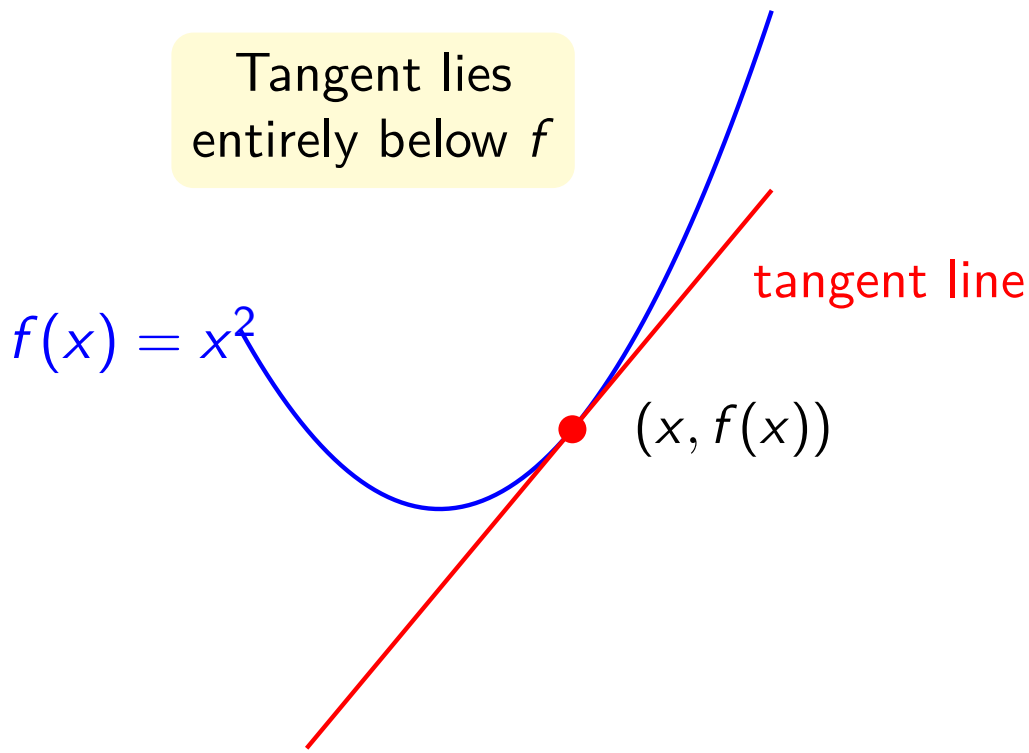
To analyze optimality, we need to differentiate $\iota_{\mathbb{R}^d_+}$. This is where subdifferentials come in.

Review: First-Order Characterization of Convex Functions

For **differentiable** convex f:

$$f(y) \geq f(x) + \langle \nabla f(x), y - x \rangle \quad \forall\, y$$

The gradient defines a **global underestimator** (supporting hyperplane).



The Subdifferential: General Definition

Intuition: $\partial f(x)$ is the set of **all possible slopes** of supporting hyperplanes at x.

- At smooth points: unique supporting hyperplane $\Rightarrow$ $\partial f(x) = \{\nabla f(x)\}$
- At corners/kinks: many supporting hyperplanes $\Rightarrow$ $\partial f(x)$ is a **set**

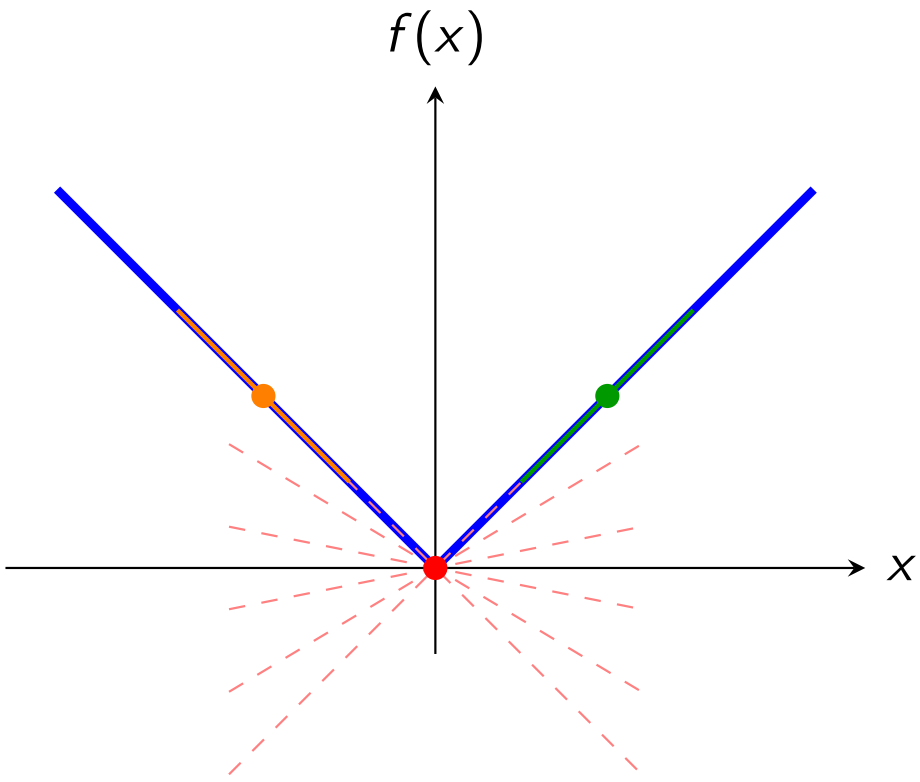
Definition

For convex $f : \mathbb{R}^d \to \mathbb{R} \cup \{\infty\}$, the **subdifferential** at $x \in \text{dom}(f)$ is:

$$\partial f(x) = \{v : f(y) \geq f(x) + \langle v, y - x \rangle \text{ for all } y \in \text{dom}(f)\}$$

Each $v \in \partial f(x)$ is called a **subgradient**.

Visualization: Subdifferential of $|x|$



Definitions:

- **At $x = 0$:**
  The "corner" allows a set of slopes:

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

- **At $x = 1$:**
  Differentiable region:

$$\partial f(1) = \{1\}$$

- **At $x = -1$:**
  Differentiable region:

$$\partial f(-1) = \{-1\}$$

Subdifferential of $|x|$: The Sign Function

$$\partial|x| = \begin{cases} \{-1\} & x < 0 \\ [-1, 1] & x = 0 \\ \{1\} & x > 0 \end{cases} = \text{sign}(x)$$

where $\text{sign}(0) = [-1, 1]$ is the **set-valued sign function**.

General pattern:

- At smooth points: subdifferential is a **singleton** (the gradient)
- At kinks: subdifferential is an **interval** between left and right derivatives

This set-valued generalization is what lets us do calculus on nonsmooth functions.

Key Facts About Subdifferentials

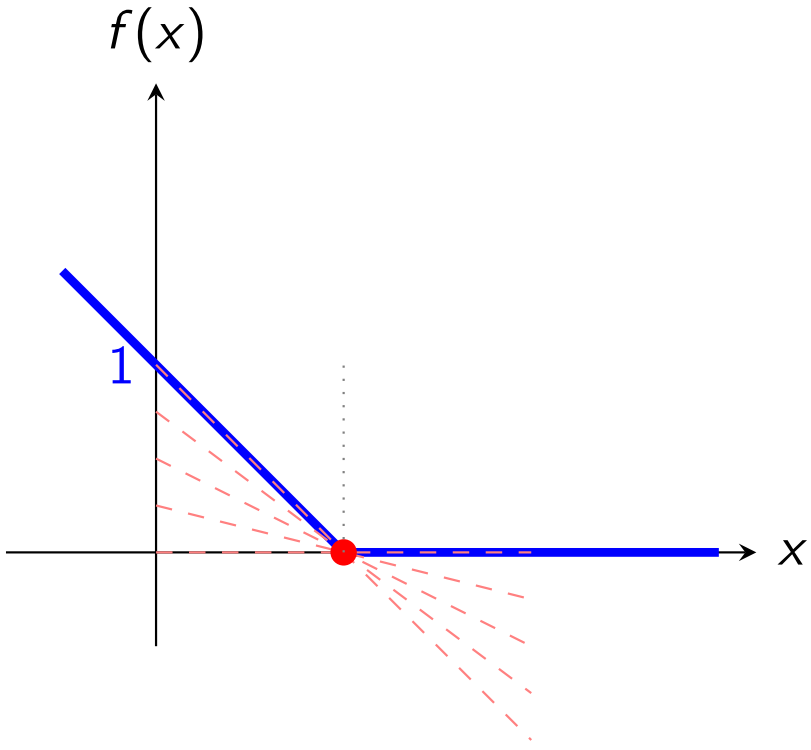
1. **Smoothness:** If f is differentiable at x, then $\partial f(x) = \{\nabla f(x)\}$

2. **Existence:** If f is proper, lsc, convex and $x \in \text{int}(\text{dom}(f))$, then $\partial f(x) \neq \emptyset$

3. **Convexity:** $\partial f(x)$ is always a closed convex set (possibly empty or unbounded)

4. **Boundary behavior:** At the boundary of $\text{dom}(f)$, the subdifferential may be empty or unbounded

Example: Subdifferential of $(1-x)_+$



Definition:

$$f(x) = \max\{0, 1-x\}$$

Subdifferential:

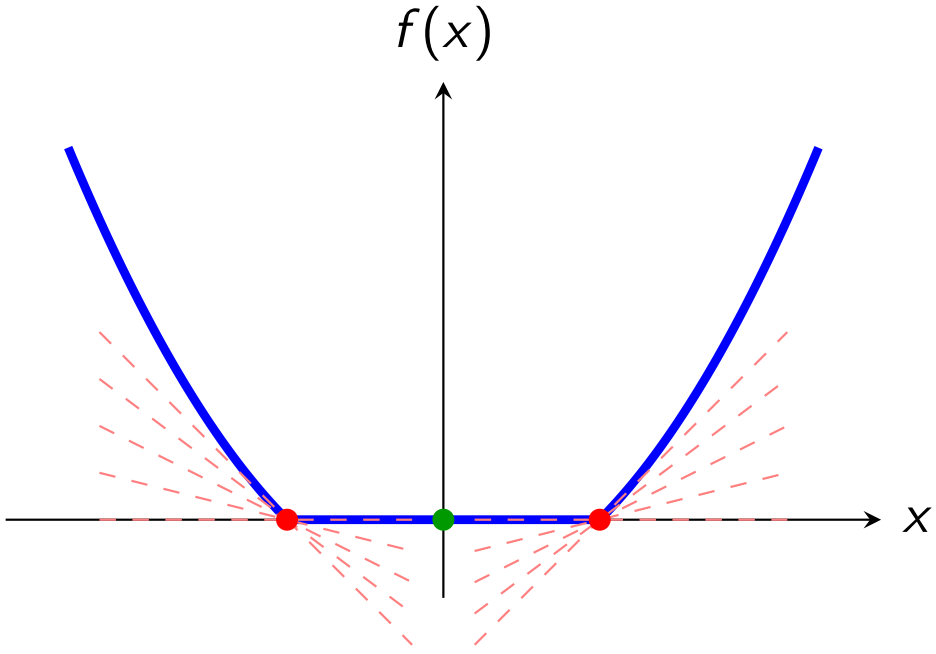
$$\partial f(x) = \begin{cases} \{-1\} & x < 1 \\ [-1, 0] & x = 1 \\ \{0\} & x > 1 \end{cases}$$

Intuition:
At the "kink" ($x = 1$), any slope $m \in [-1, 0]$ satisfies:

$$f(y) \geq f(1) + m(y-1)$$

for all y.

Example 2: $f(x) = \max\{0, \frac{1}{2}(x^2 - 1)\}$



Subdifferential:

$$\partial f(x) = \begin{cases} \{x\} & |x| > 1 \\ [0,1] & x = 1 \\ [-1,0] & x = -1 \\ \{0\} & |x| < 1 \end{cases}$$

Intuition:

The function is flat for $x \in (-1, 1)$, so the subgradient is uniquely 0.

At the boundaries, the subdifferential "fills the gap" between the slope of the flat part (0) and the parabola (x).

Example 3: Subdifferential of the ℓ_1 Norm

Separability

The ℓ_1 norm is separable across dimensions:

$$f(x) = \|x\|_1 = \sum_{i=1}^{d} |x_i|$$

Since changing x_i only affects the i-th term, we can compute the subdifferential **component-by-component**:

$$g \in \partial \|x\|_1 \iff g_i \in \partial |x_i| \quad \text{for each } i = 1, \ldots, d$$

Subdifferential of the ℓ_1 Norm, cont'd

Result

A vector g is a subgradient if its components satisfy:

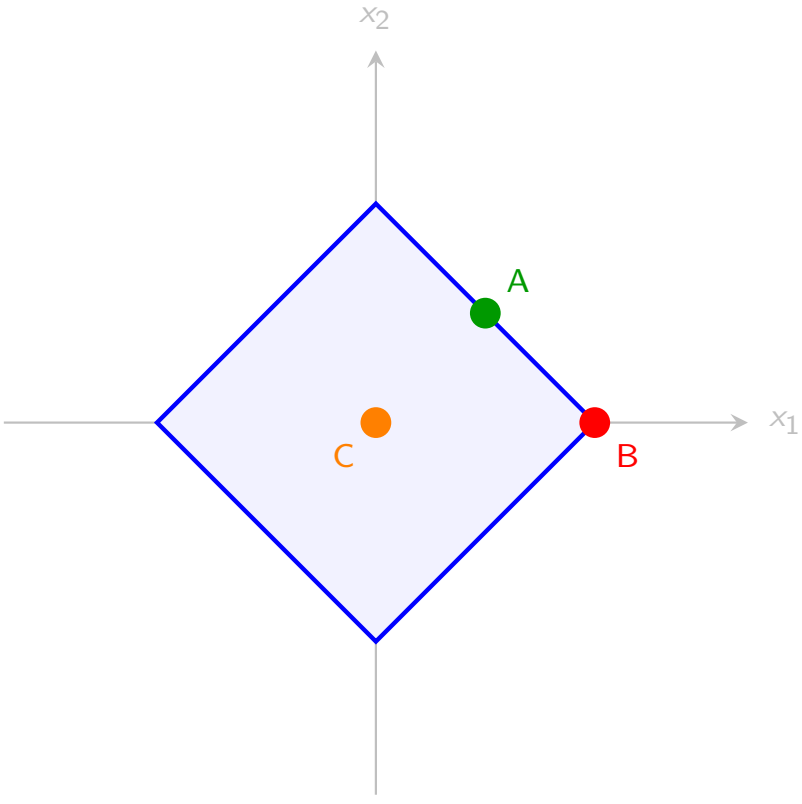
$$g_i \in \begin{cases} \{-1\} & x_i < 0 \\ [-1, 1] & x_i = 0 \\ \{1\} & x_i > 0 \end{cases}$$

Equivalently, the subdifferential is the **Cartesian product**:

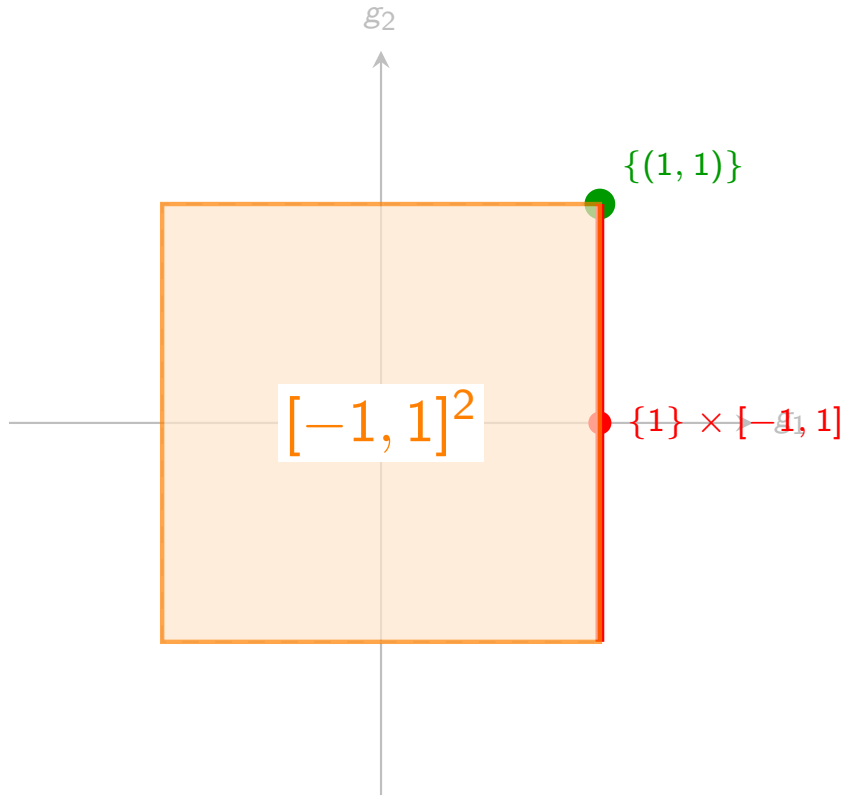
$$\partial \|x\|_1 = \partial |x_1| \times \cdots \times \partial |x_d|$$

ℓ_1 Subdifferential: Geometry in 2D

Primal Space (x)



Dual Space ($g \in \partial \|x\|_1$)



- **A**: Smooth edge ($x_1, x_2 > 0$)
- **B**: Corner $(1, 0)$
- **C**: Origin $(0, 0)$

Geometric Insight

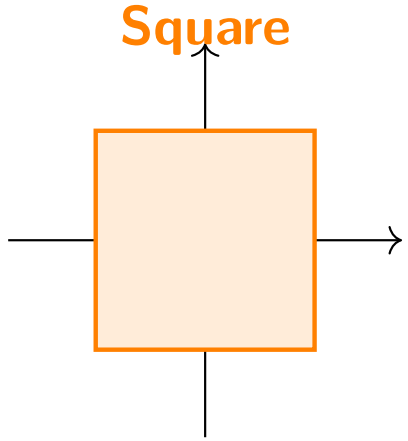
The subdifferential at the origin is the unit ball of the **dual norm** (ℓ_∞).

Example 4: Subdifferential of the ℓ_2 Norm

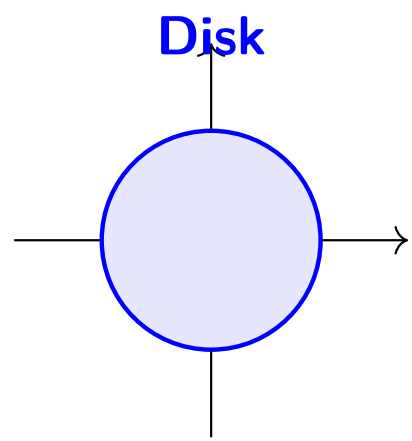
$$f(x) = \|x\|_2$$

- For $x \neq 0$: $\partial f(x) = \left\{ \dfrac{x}{\|x\|_2} \right\}$ (unit vector in direction of x)
- For $x = 0$: $\partial f(0) = \{v : \|v\|_2 \leq 1\}$ (closed unit ball)

Contrast with ℓ_1 at origin:



$\partial \|\cdot\|_1(0) = [-1, 1]^2$



$\partial \|\cdot\|_2(0) = B(0, 1)$

What We Learned

1. **Lower semicontinuity** is essential for existence of minimizers — non-lsc functions may have infima that are not attained
2. **Extended-value functions** and **convex indicators** ι_C unify constrained and unconstrained optimization into a single framework
3. **Subdifferentials** $\partial f(x)$ generalize gradients to nonsmooth convex functions as the set of all supporting hyperplane slopes

Gradients vs Subdifferentials

Property	**Gradient** $\nabla f(x)$	**Subdifferential** $\partial f(x)$
Exists when	f differentiable	$x \in \text{int}(\text{dom}(f))$, f proper, lsc, convex
Size	single vector	closed convex set
Optimality	$\nabla f(x^*) = 0$	$0 \in \partial f(x^*)$
Algorithm	gradient descent	subgradient descent
Smoothness req.	yes	no
Geometric meaning	tangent hyperplane slope	set of supporting hyperplane slopes

Computational Note

While subdifferentials are **sets**, algorithms typically work with a **single element** from $\partial f(x)$ at each iteration.

Common choices:

- At smooth points: the unique gradient $\nabla f(x)$
- At kinks: any convenient subgradient (e.g., a one-sided derivative)
- For ℓ_1 at zero coordinates: often $v_j = 0$ (minimum norm subgradient)

The choice of which element can affect algorithm behavior in practice, but **any choice is valid** for convergence guarantees.

Questions?

Next time: convex projections, subgradient calculus rules, optimality conditions