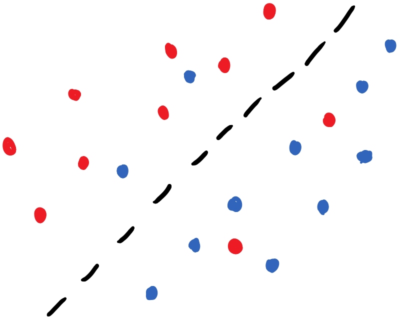
COMS 4771 Lecture 9

1. Soft-margin SVMs and surrogate losses
2. Convex optimization

Soft-margin SVMs and surrogate losses

Non-separable cases

Non-separable cases:
No linear classifier has zero training error on S.



But if non-separability is only due to a handful of points,
can we still find a good linear classifier?

no need to achieve
0 training error rate

Soft-margin SVMs (Cortes and Vapnik, 1995)

When $S = ((\boldsymbol{x}^{(1)}, y^{(1)}), \ldots, (\boldsymbol{x}^{(n)}, y^{(n)}))$ is not linearly separable, the (primal) SVM optimization problem

$$\begin{aligned} \min_{\boldsymbol{w} \in \mathbb{R}^d, \theta \in \mathbb{R}} \quad & \frac{1}{2}\|\boldsymbol{w}\|_2^2 \\ \text{s.t.} \quad & y^{(i)}\Big(\langle \boldsymbol{w}, \boldsymbol{x}^{(i)} \rangle - \theta\Big) \geq 1 \qquad \text{for } i = 1, 2, \ldots, n \end{aligned}$$

has no solution.

Soft-margin SVMs (Cortes and Vapnik, 1995)

When $S = ((\boldsymbol{x}^{(1)}, y^{(1)}), \ldots, (\boldsymbol{x}^{(n)}, y^{(n)}))$ is not linearly separable, the (primal) SVM optimization problem

$$\begin{aligned} \min_{\boldsymbol{w} \in \mathbb{R}^d, \theta \in \mathbb{R}} \quad & \frac{1}{2}\|\boldsymbol{w}\|_2^2 \\ \text{s.t.} \quad & y^{(i)}\Big(\langle \boldsymbol{w}, \boldsymbol{x}^{(i)} \rangle - \theta\Big) \geq 1 \qquad \text{for } i = 1, 2, \ldots, n \end{aligned}$$

has no solution.

Introduce **slack variables** $\xi_1, \xi_2, \ldots, \xi_n \geq 0$, and a trade-off parameter $C > 0$:

$$\begin{aligned} \min_{\boldsymbol{w} \in \mathbb{R}^d, \theta \in \mathbb{R}, \boldsymbol{\xi} \in \mathbb{R}^n} \quad & \frac{1}{2}\|\boldsymbol{w}\|_2^2 + C \sum_{i=1}^{n} \xi_i \\ \text{s.t.} \quad & y^{(i)}\Big(\langle \boldsymbol{w}, \boldsymbol{x}^{(i)} \rangle - \theta\Big) \geq 1 - \xi_i \qquad \text{for } i = 1, 2, \ldots, n \\ & \xi_i \geq 0 \qquad \text{for } i = 1, 2, \ldots, n \end{aligned}$$

Text

which is **always feasible**—"soft margin" SVM.

Soft-margin SVMs (Cortes and Vapnik, 1995)

Winner of 2008 ACM Paris Kanellakis Award:

For "their revolutionary development of a highly effective algorithm known as Support Vector Machines (SVM), a set of related supervised learning methods used for data classification and regression", which is "one of the most frequently used algorithms in machine learning, and is used in medical diagnosis, weather forecasting, and intrusion detection among many other practical applications".

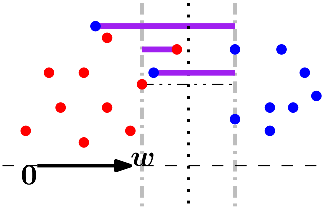
Other winners include: public key cryptography, Lempel-Ziv compression, Splay Trees, interior point method for linear programming, ...
and AdaBoost (discussed later in the course).

Slack interpretation

$$\min_{\boldsymbol{w}\in\mathbb{R}^d,\theta\in\mathbb{R},\boldsymbol{\xi}\in\mathbb{R}^n} \quad \frac{1}{2}\|\boldsymbol{w}\|_2^2 + C\sum_{i=1}^{n}\xi_i$$

$$\text{s.t.} \quad y^{(i)}\left(\langle \boldsymbol{w}, \boldsymbol{x}^{(i)}\rangle - \theta\right) \geq 1 - \xi_i \qquad \text{for } i = 1, 2, \ldots, n$$

$$\xi_i \geq 0 \qquad \text{for } i = 1, 2, \ldots, n$$

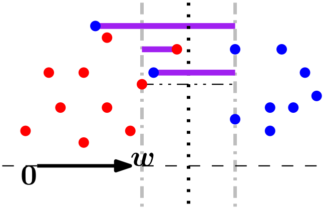


Slack interpretation

$$\min_{\boldsymbol{w}\in\mathbb{R}^d,\theta\in\mathbb{R},\boldsymbol{\xi}\in\mathbb{R}^n} \quad \frac{1}{2}\|\boldsymbol{w}\|_2^2 + C\sum_{i=1}^n \xi_i$$

$$\text{s.t.} \quad y^{(i)}\Big(\langle \boldsymbol{w}, \boldsymbol{x}^{(i)}\rangle - \theta\Big) \geq 1 - \xi_i \qquad \text{for } i = 1, 2, \ldots, n$$

$$\xi_i \geq 0 \qquad \text{for } i = 1, 2, \ldots, n$$



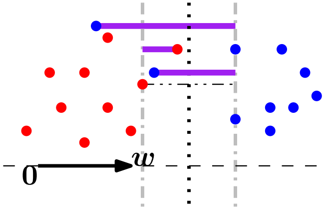
- For a given $(\boldsymbol{w}, \theta)$, $\xi_i / \|\boldsymbol{w}\|_2$ measures distance that $\boldsymbol{x}^{(i)}$ must be moved so that $y^{(i)}\langle \boldsymbol{w}, \boldsymbol{x}^{(i)} - \theta\rangle \geq 1$.

Slack interpretation

$$\min_{\boldsymbol{w}\in\mathbb{R}^d,\theta\in\mathbb{R},\boldsymbol{\xi}\in\mathbb{R}^n} \quad \frac{1}{2}\|\boldsymbol{w}\|_2^2 + C\sum_{i=1}^n \xi_i$$

$$\text{s.t.} \quad y^{(i)}\left(\langle \boldsymbol{w}, \boldsymbol{x}^{(i)}\rangle - \theta\right) \geq 1 - \xi_i \qquad \text{for } i = 1, 2, \ldots, n$$

$$\xi_i \geq 0 \qquad \text{for } i = 1, 2, \ldots, n$$



- For a given $(\boldsymbol{w}, \theta)$, $\xi_i/\|\boldsymbol{w}\|_2$ measures distance that $\boldsymbol{x}^{(i)}$ must be moved so that $y^{(i)}\langle \boldsymbol{w}, \boldsymbol{x}^{(i)} - \theta\rangle \geq 1$.
- C controls trade-off between slack penalties and size of margin (which is $1/\|\boldsymbol{w}\|_2$).

Constraints with non-negative slack variables: (using $\lambda = 1/(nC)$)

$$\min_{\boldsymbol{w}\in\mathbb{R}^d, \theta\in\mathbb{R}, \boldsymbol{\xi}\in\mathbb{R}^n} \quad \frac{\lambda}{2}\|\boldsymbol{w}\|_2^2 + \frac{1}{n}\sum_{i=1}^{n}\xi_i$$

$$\text{s.t.} \quad y^{(i)}\left(\langle \boldsymbol{w}, \boldsymbol{x}^{(i)}\rangle - \theta\right) \geq 1 - \xi_i \qquad \text{for } i = 1, 2, \ldots, n$$

$$\xi_i \geq 0 \qquad \text{for } i = 1, 2, \ldots, n$$

A different interpretation of slack

Constraints with non-negative slack variables: (using $\lambda = 1/(nC)$)

$$\begin{aligned} \min_{\boldsymbol{w}\in\mathbb{R}^d,\theta\in\mathbb{R},\boldsymbol{\xi}\in\mathbb{R}^n} \quad & \frac{\lambda}{2}\|\boldsymbol{w}\|_2^2 + \frac{1}{n}\sum_{i=1}^{n}\xi_i \\ \text{s.t.} \quad & y^{(i)}\Big(\langle \boldsymbol{w}, \boldsymbol{x}^{(i)}\rangle - \theta\Big) \geq 1 - \xi_i \qquad \text{for } i = 1, 2, \ldots, n \\ & \xi_i \geq 0 \qquad \text{for } i = 1, 2, \ldots, n \end{aligned}$$

Equivalent unconstrained form:

$$\min_{\boldsymbol{w}\in\mathbb{R}^d,\theta\in\mathbb{R}} \quad \frac{\lambda}{2}\|\boldsymbol{w}\|_2^2 + \frac{1}{n}\sum_{i=1}^{n}\Big[1 - y^{(i)}\Big(\langle \boldsymbol{w}, \boldsymbol{x}^{(i)}\rangle - \theta\Big)\Big]_+$$

Notation: $[a]_+ := \max\{0, a\}$.

A different interpretation of slack

Constraints with non-negative slack variables: (using $\lambda = 1/(nC)$)

$$\begin{aligned} \min_{\boldsymbol{w}\in\mathbb{R}^d,\theta\in\mathbb{R},\boldsymbol{\xi}\in\mathbb{R}^n} \quad & \frac{\lambda}{2}\|\boldsymbol{w}\|_2^2 + \frac{1}{n}\sum_{i=1}^{n}\xi_i \\ \text{s.t.} \quad & y^{(i)}\left(\langle \boldsymbol{w}, \boldsymbol{x}^{(i)}\rangle - \theta\right) \geq 1 - \xi_i \qquad \text{for } i = 1, 2, \ldots, n \\ & \xi_i \geq 0 \qquad \text{for } i = 1, 2, \ldots, n \end{aligned}$$

Equivalent unconstrained form:

$$\min_{\boldsymbol{w}\in\mathbb{R}^d,\theta\in\mathbb{R}} \quad \frac{\lambda}{2}\|\boldsymbol{w}\|_2^2 + \frac{1}{n}\sum_{i=1}^{n}\left[1 - y^{(i)}\left(\langle \boldsymbol{w}, \boldsymbol{x}^{(i)}\rangle - \theta\right)\right]_+$$

Notation: $[a]_+ := \max\{0, a\}$.

The **hinge loss** of a linear classifier $f_{\boldsymbol{w},\theta}$ on an example $(\boldsymbol{x}, y)$ is defined to be

$$\mathrm{HL}(\boldsymbol{w}, \theta; \boldsymbol{x}, y) := \Big[1 - y(\langle \boldsymbol{w}, \boldsymbol{x}\rangle - \theta)\Big]_+.$$

A different interpretation of slack

Constraints with non-negative slack variables: (using $\lambda = 1/(nC)$)

$$\begin{aligned} \min_{\boldsymbol{w}\in\mathbb{R}^d,\theta\in\mathbb{R},\boldsymbol{\xi}\in\mathbb{R}^n} \quad & \frac{\lambda}{2}\|\boldsymbol{w}\|_2^2 + \frac{1}{n}\sum_{i=1}^{n}\xi_i \\ \text{s.t.} \quad & y^{(i)}\left(\langle \boldsymbol{w}, \boldsymbol{x}^{(i)}\rangle - \theta\right) \geq 1 - \xi_i \qquad \text{for } i = 1, 2, \ldots, n \\ & \xi_i \geq 0 \qquad \text{for } i = 1, 2, \ldots, n \end{aligned}$$

Equivalent unconstrained form:

$$\min_{\boldsymbol{w}\in\mathbb{R}^d,\theta\in\mathbb{R}} \quad \frac{\lambda}{2}\|\boldsymbol{w}\|_2^2 + \frac{1}{n}\sum_{i=1}^{n}\mathrm{HL}(\boldsymbol{w},\theta;\boldsymbol{x}^{(i)},y^{(i)})$$

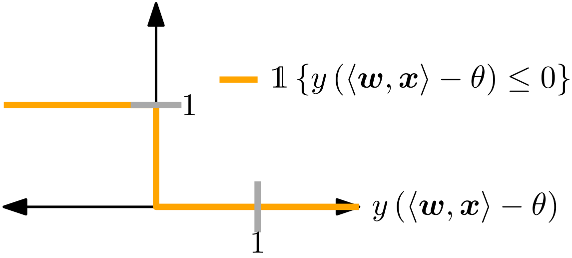
Notation: $[a]_+ := \max\{0, a\}$.

trade off : larger margin ? lower hinge

The **hinge loss** of a linear classifier $f_{\boldsymbol{w},\theta}$ on an example $(\boldsymbol{x}, y)$ is defined to be

$$\mathrm{HL}(\boldsymbol{w},\theta;\boldsymbol{x},y) := \Big[1 - y(\langle \boldsymbol{w}, \boldsymbol{x}\rangle - \theta)\Big]_+.$$

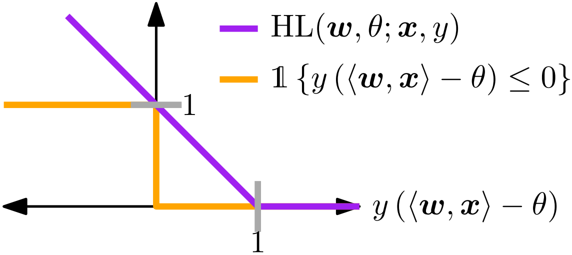
Zero-one loss vs. hinge loss



Zero-one loss: count if $f_{\boldsymbol{w},\theta}(\boldsymbol{x}) \neq y$.

$$\mathbb{1}\{y(\langle \boldsymbol{w}, \boldsymbol{x}\rangle - \theta) \le 0\} \;\le\; \Big[1 - y(\langle \boldsymbol{w}, \boldsymbol{x}\rangle - \theta)\Big]_+ \;=\; \mathrm{HL}(\boldsymbol{w}, \theta; \boldsymbol{x}, y).$$

Zero-one loss vs. hinge loss



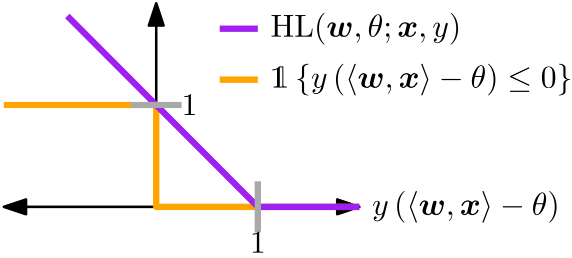
Hinge loss: an upper-bound on zero-one loss.

$$\boxed{\mathbb{1}\{y(\langle \boldsymbol{w}, \boldsymbol{x}\rangle - \theta) \le 0\}} \le \left[1 - \boxed{y(\langle \boldsymbol{w}, \boldsymbol{x}\rangle - \theta)}\right]_+ = \mathrm{HL}(\boldsymbol{w}, \theta; \boldsymbol{x}, y).$$

wrong predication!

When this smaller than one, a penalty would be incured!

Zero-one loss vs. hinge loss

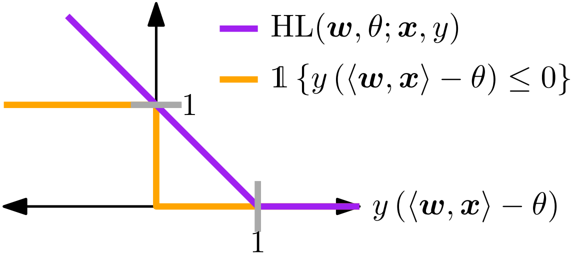


Hinge loss: an upper-bound on zero-one loss.

$$\mathbb{1}\{y(\langle \boldsymbol{w}, \boldsymbol{x} \rangle - \theta) \le 0\} \;\le\; \Big[1 - y(\langle \boldsymbol{w}, \boldsymbol{x} \rangle - \theta)\Big]_+ \;=\; \mathrm{HL}(\boldsymbol{w}, \theta; \boldsymbol{x}, y).$$

Soft-margin SVM minimizes an upper-bound on the training error, plus a term that favors large margins.

Zero-one loss vs. hinge loss



Hinge loss: an upper-bound on zero-one loss.

$$\mathbb{1}\{y(\langle \boldsymbol{w}, \boldsymbol{x} \rangle - \theta) \le 0\} \;\le\; \Big[1 - y(\langle \boldsymbol{w}, \boldsymbol{x} \rangle - \theta)\Big]_+ \;=\; \mathrm{HL}(\boldsymbol{w}, \theta; \boldsymbol{x}, y).$$

Soft-margin SVM minimizes an upper-bound on the training error, plus a term that favors large margins.

This is **computationally tractable** (unlike minimizing training error) because the hinge loss is a **convex function** of $(\boldsymbol{w}, \theta)$, and so is $\frac{\lambda}{2}\|\boldsymbol{w}\|_2^2$.

General form

Empirical risk minimization (i.e., minimize training error):

$$\min_{\boldsymbol{w} \in \mathbb{R}^d, \theta \in \mathbb{R}} \qquad \frac{1}{n} \sum_{i=1}^{n} \mathbb{1}\left\{ y^{(i)} \Big(\langle \boldsymbol{w}, \boldsymbol{x}^{(i)} \rangle - \theta \Big) \leq 0 \right\}$$

Soft-margin SVM:

$$\min_{\boldsymbol{w} \in \mathbb{R}^d, \theta \in \mathbb{R}} \qquad \frac{\lambda}{2} \|\boldsymbol{w}\|_2^2 + \frac{1}{n} \sum_{i=1}^{n} \mathrm{HL}(\boldsymbol{w}, \theta; \boldsymbol{x}^{(i)}, y^{(i)})$$

General form

Empirical risk minimization (i.e., minimize training error):

$$\min_{\boldsymbol{w} \in \mathbb{R}^d, \theta \in \mathbb{R}} \qquad \frac{1}{n} \sum_{i=1}^{n} \mathbb{1}\left\{ y^{(i)} \left(\langle \boldsymbol{w}, \boldsymbol{x}^{(i)} \rangle - \theta \right) \leq 0 \right\}$$

Soft-margin SVM:

$$\min_{\boldsymbol{w} \in \mathbb{R}^d, \theta \in \mathbb{R}} \qquad \frac{\lambda}{2} \|\boldsymbol{w}\|_2^2 + \frac{1}{n} \sum_{i=1}^{n} \mathrm{HL}(\boldsymbol{w}, \theta; \boldsymbol{x}^{(i)}, y^{(i)})$$

Generic learning objective:

$$\min_{\boldsymbol{w} \in \mathbb{R}^d} \qquad R(\boldsymbol{w}) + \frac{1}{n} \sum_{i=1}^{n} \ell(\boldsymbol{w}; \boldsymbol{x}^{(i)}, y^{(i)})$$

- **Regularization**: encodes "learning bias" (e.g., preference for large margins), sometimes promotes stability.
- **Data fitting/empirical loss**: how poorly does the classifier "fit" the data.

Similarity to maximum likelihood

Generic learning objective:

$$\min_{\boldsymbol{w} \in \mathbb{R}^d} \qquad R(\boldsymbol{w}) + \frac{1}{n} \sum_{i=1}^{n} \ell(\boldsymbol{w}; \boldsymbol{x}^{(i)}, y^{(i)})$$

Text

Maximum likelihood estimation for a parameteric family $\{p(\cdot; \boldsymbol{\theta}) : \boldsymbol{\theta} \in \mathcal{T}\}$ (assuming data $\boldsymbol{z}^{(1)}, \ldots, \boldsymbol{z}^{(n)}$ are iid):

$$\min_{\boldsymbol{\theta} \in \mathbb{R}^d} \qquad -\frac{1}{n} \sum_{i=1}^{n} \log p(\boldsymbol{z}^{(i)}; \boldsymbol{\theta})$$

(i.e., minimize $1/n$ times negative log-likelihood of parameter $\boldsymbol{\theta}$).

Similarity to maximum likelihood

Generic learning objective:

$$\min_{\boldsymbol{w} \in \mathbb{R}^d} \qquad R(\boldsymbol{w}) + \frac{1}{n} \sum_{i=1}^{n} \ell(\boldsymbol{w}; \boldsymbol{x}^{(i)}, y^{(i)})$$

Maximum likelihood estimation for a parameteric family $\{p(\cdot; \boldsymbol{\theta}) : \boldsymbol{\theta} \in \mathcal{T}\}$ (assuming data $\boldsymbol{z}^{(1)}, \ldots, \boldsymbol{z}^{(n)}$ are iid):

$$\min_{\boldsymbol{\theta} \in \mathbb{R}^d} \qquad -\frac{1}{n} \sum_{i=1}^{n} \log p(\boldsymbol{z}^{(i)}; \boldsymbol{\theta})$$

(i.e., minimize $1/n$ times negative log-likelihood of parameter $\boldsymbol{\theta}$).

Sometimes generic learning objective is called "generalized (and penalized, because of $R(\boldsymbol{w})$) maximum likelihood estimation".

Learning via optimization

- Many different choices for regularization and loss.
  - $R(\boldsymbol{w}) \propto \|\boldsymbol{w}\|_2^2$: encourage large margins
  - $R(\boldsymbol{w}) \propto \|\boldsymbol{w}\|_1$: encourage $\boldsymbol{w}$ to be sparse
  - $R(\boldsymbol{w}) \propto \sum_{i=1}^{n} w_i \ln w_i$: "maximum entropy" interpretation
  - $\ell(\boldsymbol{w}, \theta; \boldsymbol{x}, y) = [\langle \boldsymbol{w}, \boldsymbol{x} \rangle - \theta - y]_+^2$
  - $\ell(\boldsymbol{w}, \theta; \boldsymbol{x}, y) = \ln(1 + \exp(-y(\langle \boldsymbol{w}, \boldsymbol{x} \rangle - \theta)))$ ("logistic regression")
  - ...
  - Also used beyond classification (e.g., regression, parameter estimation, clustering)

Learning via optimization

- Many different choices for regularization and loss.
  - $R(\boldsymbol{w}) \propto \|\boldsymbol{w}\|_2^2$: encourage large margins
  - $R(\boldsymbol{w}) \propto \|\boldsymbol{w}\|_1$: encourage $\boldsymbol{w}$ to be sparse
  - $R(\boldsymbol{w}) \propto \sum_{i=1}^{n} w_i \ln w_i$: "maximum entropy" interpretation
  - $\ell(\boldsymbol{w}, \theta; \boldsymbol{x}, y) = [\langle \boldsymbol{w}, \boldsymbol{x} \rangle - \theta - y]_+^2$
  - $\ell(\boldsymbol{w}, \theta; \boldsymbol{x}, y) = \ln(1 + \exp(-y(\langle \boldsymbol{w}, \boldsymbol{x} \rangle - \theta)))$ ("logistic regression")
  - ...
  - Also used beyond classification (e.g., regression, parameter estimation, clustering)
- Often want ℓ be an upper-bound on zero-one loss—i.e., a **surrogate loss**.

Learning via optimization

- Many different choices for regularization and loss.
  - $R(\boldsymbol{w}) \propto \|\boldsymbol{w}\|_2^2$: encourage large margins
  - $R(\boldsymbol{w}) \propto \|\boldsymbol{w}\|_1$: encourage $\boldsymbol{w}$ to be sparse
  - $R(\boldsymbol{w}) \propto \sum_{i=1}^{n} w_i \ln w_i$: "maximum entropy" interpretation
  - $\ell(\boldsymbol{w}, \theta; \boldsymbol{x}, y) = [\langle \boldsymbol{w}, \boldsymbol{x} \rangle - \theta - y]_+^2$
  - $\ell(\boldsymbol{w}, \theta; \boldsymbol{x}, y) = \ln(1 + \exp(-y(\langle \boldsymbol{w}, \boldsymbol{x} \rangle - \theta)))$ ("logistic regression")
  - ...
  - Also used beyond classification (e.g., regression, parameter estimation, clustering)
- Often want ℓ be an upper-bound on zero-one loss—i.e., a **surrogate loss**.
- Trade-off parameter λ: usually determine using hold out error or cross validation error.

Learning via optimization

- Many different choices for regularization and loss.
  - $R(\boldsymbol{w}) \propto \|\boldsymbol{w}\|_2^2$: encourage large margins
  - $R(\boldsymbol{w}) \propto \|\boldsymbol{w}\|_1$: encourage $\boldsymbol{w}$ to be sparse
  - $R(\boldsymbol{w}) \propto \sum_{i=1}^{n} w_i \ln w_i$: "maximum entropy" interpretation
  - $\ell(\boldsymbol{w}, \theta; \boldsymbol{x}, y) = [\langle \boldsymbol{w}, \boldsymbol{x} \rangle - \theta - y]_+^2$
  - $\ell(\boldsymbol{w}, \theta; \boldsymbol{x}, y) = \ln(1 + \exp(-y(\langle \boldsymbol{w}, \boldsymbol{x} \rangle - \theta)))$ ("logistic regression")
  - ...
  - Also used beyond classification (e.g., regression, parameter estimation, clustering)
- Often want ℓ be an upper-bound on zero-one loss—i.e., a **surrogate loss**.
- Trade-off parameter λ: usually determine using hold out error or cross validation error.
- Computationally easier when overall objective function is **convex**: possible to efficiently find global minimizer in polynomial time.

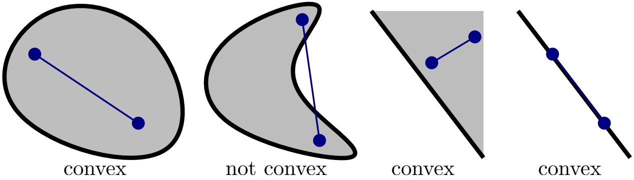
Learning via optimization

- Many different choices for regularization and loss.
  - $R(\boldsymbol{w}) \propto \|\boldsymbol{w}\|_2^2$: encourage large margins
  - $R(\boldsymbol{w}) \propto \|\boldsymbol{w}\|_1$: encourage $\boldsymbol{w}$ to be sparse absolute value
  - $R(\boldsymbol{w}) \propto \sum_{i=1}^{n} w_i \ln w_i$: "maximum entropy" interpretation
  - $\ell(\boldsymbol{w}, \theta; \boldsymbol{x}, y) = [\langle \boldsymbol{w}, \boldsymbol{x} \rangle - \theta - y]_+^2$
  - $\ell(\boldsymbol{w}, \theta; \boldsymbol{x}, y) = \ln(1 + \exp(-y(\langle \boldsymbol{w}, \boldsymbol{x} \rangle - \theta)))$ ("logistic regression")
  - ...
  - Also used beyond classification (e.g., regression, parameter estimation, clustering)
- Often want ℓ be an upper-bound on zero-one loss—i.e., a **surrogate loss**.
- Trade-off parameter λ: usually determine using hold out error or cross validation error.
- Computationally easier when overall objective function is **convex**: possible to efficiently find global minimizer in polynomial time.
- **Next**: techniques for analyzing and solving these optimization problems.

Introduction to convexity

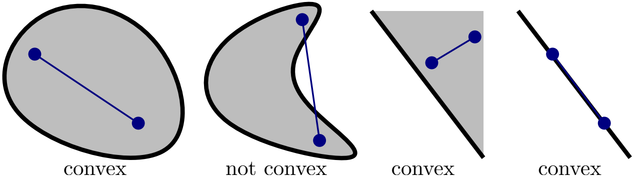
Convex sets

We say a set A is **convex** if, for every pair of points $\{\boldsymbol{x}, \boldsymbol{x}'\}$ in the set A, the line segment between the points $\boldsymbol{x}$ and $\boldsymbol{x}'$ is also contained in the set A.



Convex sets

We say a set A is **convex** if, for every pair of points $\{\boldsymbol{x}, \boldsymbol{x}'\}$ in the set A, the line segment between the points $\boldsymbol{x}$ and $\boldsymbol{x}'$ is also contained in the set A.



Examples:

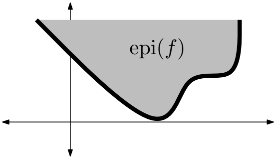
- All of $\mathbb{R}^d$.
- Empty set.
- Affine hyperplanes.
- Half-spaces: $\{\boldsymbol{a} \in \mathbb{R}^d : \langle \boldsymbol{a}, \boldsymbol{x} \rangle - b \leq 0\}$.
- Intersections of convex sets.
- Convex hulls of points.

Convex functions

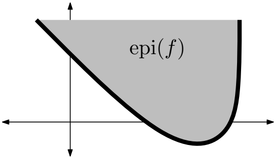
For any function $f\colon \mathbb{R}^d \to \mathbb{R}$, the **epigraph** of f, denoted $\operatorname{epi}(f)$, is the set

$$\operatorname{epi}(f) := \{(\boldsymbol{x}, b) \in \mathbb{R}^{d+1} : f(\boldsymbol{x}) \leq b\}.$$

We say a function f is **convex** if $\operatorname{epi}(f)$ is a convex set in $\mathbb{R}^{d+1}$.



f is not convex



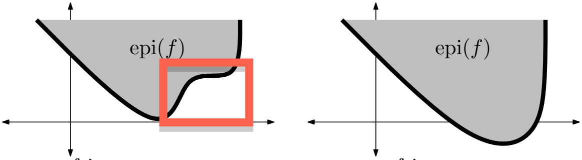
f is convex

Convex functions

For any function $f \colon \mathbb{R}^d \to \mathbb{R}$, the **epigraph** of f, denoted $\operatorname{epi}(f)$, is the set

$$\operatorname{epi}(f) := \{(\boldsymbol{x}, b) \in \mathbb{R}^{d+1} : f(\boldsymbol{x}) \leq b\}.$$

We say a function f is **convex** if $\operatorname{epi}(f)$ is a convex set in $\mathbb{R}^{d+1}$.



f is not convex

f is convex

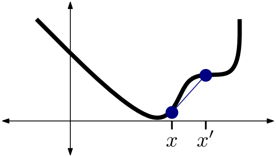
Examples:

- $f(x) = c^x$ for any $c > 0$ (on $\mathbb{R}$)
- $f(x) = |x|^c$ for any $c \geq 1$ (on $\mathbb{R}$)
- $f(\boldsymbol{x}) = c$ for any constant $c \in \mathbb{R}$.
- $f(\boldsymbol{x}) = \langle \boldsymbol{a}, \boldsymbol{x} \rangle$ for any $\boldsymbol{a} \in \mathbb{R}^d$.
- $f(\boldsymbol{x}) = \|\boldsymbol{x}\|_p$ for any $1 \leq p \leq \infty$.
- $f(\boldsymbol{x}) = \langle \boldsymbol{x}, \boldsymbol{A}\boldsymbol{x} \rangle$ for symmetric positive semidefinite $\boldsymbol{A}$.

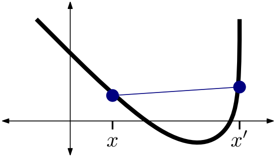
Equivalent definition of convex functions

A function $f\colon \mathbb{R}^d \to \mathbb{R}$ is convex if and only if, for any $\boldsymbol{x}, \boldsymbol{x}' \in \mathbb{R}^d$ and $\alpha \in [0,1]$,

$$f\big((1-\alpha)\boldsymbol{x} + \alpha\boldsymbol{x}'\big) \ \le \ (1-\alpha)\cdot f(\boldsymbol{x}) + \alpha \cdot f(\boldsymbol{x}').$$



f is not convex

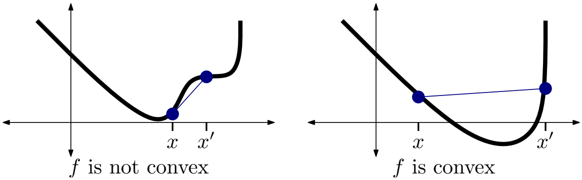


f is convex

Equivalent definition of convex functions

A function $f\colon \mathbb{R}^d \to \mathbb{R}$ is convex if and only if, for any $\boldsymbol{x}, \boldsymbol{x}' \in \mathbb{R}^d$ and $\alpha \in [0,1]$,

$$f\big((1-\alpha)\boldsymbol{x} + \alpha\boldsymbol{x}'\big) \;\le\; (1-\alpha)\cdot f(\boldsymbol{x}) + \alpha \cdot f(\boldsymbol{x}').$$



Covex points

Jensen's inequality

If $f\colon \mathbb{R}^d \to \mathbb{R}$ is convex, then for any $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n \in \mathbb{R}^d$ and $\alpha_1, \alpha_2, \ldots, \alpha_n \in [0,1]$ such that $\sum_{i=1}^n \alpha_i = 1$,

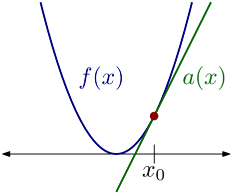
$$f\left(\sum_{i=1}^n \alpha_i \boldsymbol{x}_i\right) \le \sum_{i=1}^n \alpha_i \cdot f(\boldsymbol{x}_i).$$

Differentiable functions

If $f\colon \mathbb{R}^d \to \mathbb{R}$ is differentiable, then f is convex if and only if

$$f(\boldsymbol{x}) \geq f(\boldsymbol{x}_0) + \langle \nabla f(\boldsymbol{x}_0), \boldsymbol{x} - \boldsymbol{x}_0 \rangle$$

for all $\boldsymbol{x}, \boldsymbol{x}_0 \in \mathbb{R}^d$.



$$a(x) = f(x_0) + f'(x_0)(x - x_0)$$

Text

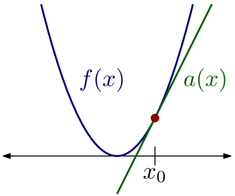
Convexity of differentiable functions

Differentiable functions

If $f\colon \mathbb{R}^d \to \mathbb{R}$ is differentiable, then f is convex if and only if

$$f(\boldsymbol{x}) \geq f(\boldsymbol{x}_0) + \langle \nabla f(\boldsymbol{x}_0), \boldsymbol{x} - \boldsymbol{x}_0 \rangle$$

for all $\boldsymbol{x}, \boldsymbol{x}_0 \in \mathbb{R}^d$.



$$a(x) = f(x_0) + f'(x_0)(x - x_0)$$

Twice-differentiable functions

If $f\colon \mathbb{R}^d \to \mathbb{R}$ is twice-differentiable, then f is convex if and only if

$$\nabla^2 f(\boldsymbol{x}) \succeq \boldsymbol{0}$$

for all $\boldsymbol{x} \in \mathbb{R}^d$ (i.e., the Hessian, or matrix of second-derivatives, is positive semidefinite for all $\boldsymbol{x}$).

Building new convex functions from old ones

- $f(\boldsymbol{x}) = c \cdot h(\boldsymbol{x}) + g(\boldsymbol{x})$ for convex functions h, g and scalar $c \geq 0$.

  Example: $f(\boldsymbol{x}) = -2\langle \boldsymbol{a}, \boldsymbol{x} \rangle + 1$.

- $f(\boldsymbol{x}) = \max\{h(\boldsymbol{x}),\ g(\boldsymbol{x})\}$ for convex functions h, g.

  Example: $f(\boldsymbol{x}) = \max\{0,\ 1 - \langle \boldsymbol{a}, \boldsymbol{x} \rangle\}$.

- $f(\boldsymbol{x}) = g(\boldsymbol{A}\boldsymbol{x} + \boldsymbol{b})$ for any $\boldsymbol{A} \in \mathbb{R}^{m \times d}$ and $\boldsymbol{b} \in \mathbb{R}^m$ and convex function $g \colon \mathbb{R}^m \to \mathbb{R}$.

  Examples: $f(\boldsymbol{x}) = \exp(\langle \boldsymbol{a}, \boldsymbol{x} \rangle)$, $f(\boldsymbol{x}) = \langle \boldsymbol{a}, \boldsymbol{x} \rangle^2$.

- $f(\boldsymbol{x}) = g(h_1(\boldsymbol{x}), \dots, h_m(\boldsymbol{x}))$ for any convex $h_1, \dots, h_m$ and convex $g \colon \mathbb{R}^m \to \mathbb{R}$ such that g is non-decreasing in each argument.

  Example: $f(\boldsymbol{x}) = \exp(\langle \boldsymbol{a}, \boldsymbol{x} \rangle^2)$ (but not $\exp(-\langle \boldsymbol{a}, \boldsymbol{x} \rangle^2)$).

- Many other composition rules.

How to test for convexity

- First principles (via definitions).
- First- or second-derivative tests (assuming derivatives exist).
- Valid transformation of existing convex functions.

Convex optimization problems

Optimization problems

A typical optimization problem is written as

$$\begin{aligned} \min_{\boldsymbol{x} \in \mathbb{R}^d} \quad & f_0(\boldsymbol{x}) \\ \text{s.t.} \quad & f_i(\boldsymbol{x}) \leq 0 \quad i = 1, 2, \ldots, n \end{aligned}$$

where $f_0 \colon \mathbb{R}^d \to \mathbb{R}$ is the **objective function** and $f_1, f_2, \ldots, f_n \colon \mathbb{R}^d \to \mathbb{R}$ are the **constraint functions**.

Optimization problems

A typical optimization problem is written as

$$\begin{aligned} \min_{\boldsymbol{x} \in \mathbb{R}^d} \quad & f_0(\boldsymbol{x}) \\ \text{s.t.} \quad & f_i(\boldsymbol{x}) \leq 0 \quad i = 1, 2, \ldots, n \end{aligned}$$

where $f_0 \colon \mathbb{R}^d \to \mathbb{R}$ is the **objective function** and $f_1, f_2, \ldots, f_n \colon \mathbb{R}^d \to \mathbb{R}$ are the **constraint functions**.

The inequalities $f_i(\boldsymbol{x}) \leq 0$ are **constraints**.

Optimization problems

A typical optimization problem is written as

$$\begin{aligned} \min_{\boldsymbol{x}\in\mathbb{R}^d} \quad & f_0(\boldsymbol{x}) \\ \text{s.t.} \quad & f_i(\boldsymbol{x}) \leq 0 \quad i = 1, 2, \ldots, n \end{aligned}$$

where $f_0 \colon \mathbb{R}^d \to \mathbb{R}$ is the **objective function** and $f_1, f_2, \ldots, f_n \colon \mathbb{R}^d \to \mathbb{R}$ are the **constraint functions**.

The inequalities $f_i(\boldsymbol{x}) \leq 0$ are **constraints**.

The set $A := \left\{\boldsymbol{x} \in \mathbb{R}^d : f_i(\boldsymbol{x}) \leq 0 \text{ for all } i = 1, 2, \ldots, n\right\}$ is the **feasible set**.

Optimization problems

A typical optimization problem is written as

$$\begin{aligned} \min_{\boldsymbol{x} \in \mathbb{R}^d} \quad & f_0(\boldsymbol{x}) \\ \text{s.t.} \quad & f_i(\boldsymbol{x}) \leq 0 \quad i = 1, 2, \ldots, n \end{aligned}$$

where $f_0 \colon \mathbb{R}^d \to \mathbb{R}$ is the **objective function** and $f_1, f_2, \ldots, f_n \colon \mathbb{R}^d \to \mathbb{R}$ are the **constraint functions**.

The inequalities $f_i(\boldsymbol{x}) \leq 0$ are **constraints**.

The set $A := \{\boldsymbol{x} \in \mathbb{R}^d : f_i(\boldsymbol{x}) \leq 0 \text{ for all } i = 1, 2, \ldots, n\}$ is the **feasible set**.

The goal: Find $\boldsymbol{x} \in A$ so that $f_0(\boldsymbol{x})$ is as small as possible.

Optimization problems

A typical optimization problem is written as

$$\begin{aligned} \min_{\boldsymbol{x}\in\mathbb{R}^d} \quad & f_0(\boldsymbol{x}) \\ \text{s.t.} \quad & f_i(\boldsymbol{x}) \le 0 \quad i = 1, 2, \ldots, n \end{aligned}$$

where $f_0 \colon \mathbb{R}^d \to \mathbb{R}$ is the **objective function** and $f_1, f_2, \ldots, f_n \colon \mathbb{R}^d \to \mathbb{R}$ are the **constraint functions**.

The inequalities $f_i(\boldsymbol{x}) \le 0$ are **constraints**.

The set $A := \left\{\boldsymbol{x} \in \mathbb{R}^d : f_i(\boldsymbol{x}) \le 0 \text{ for all } i = 1, 2, \ldots, n\right\}$ is the **feasible set**.

The goal: Find $\boldsymbol{x} \in A$ so that $f_0(\boldsymbol{x})$ is as small as possible.

The **(optimal) value** of the optimization problem is the smallest such value of $f_0(\boldsymbol{x})$ achieved by a feasible point $\boldsymbol{x} \in A$.

Optimization problems

A typical optimization problem is written as

$$\begin{aligned} \min_{\boldsymbol{x}\in\mathbb{R}^d} \quad & f_0(\boldsymbol{x}) \\ \text{s.t.} \quad & f_i(\boldsymbol{x}) \le 0 \quad i = 1, 2, \ldots, n \end{aligned}$$

where $f_0 \colon \mathbb{R}^d \to \mathbb{R}$ is the **objective function** and $f_1, f_2, \ldots, f_n \colon \mathbb{R}^d \to \mathbb{R}$ are the **constraint functions**.

The inequalities $f_i(\boldsymbol{x}) \le 0$ are **constraints**.

The set $A := \{\boldsymbol{x} \in \mathbb{R}^d : f_i(\boldsymbol{x}) \le 0 \text{ for all } i = 1, 2, \ldots, n\}$ is the **feasible set**.

The goal: Find $\boldsymbol{x} \in A$ so that $f_0(\boldsymbol{x})$ is as small as possible.

The **(optimal) value** of the optimization problem is the smallest such value of $f_0(\boldsymbol{x})$ achieved by a feasible point $\boldsymbol{x} \in A$.

A point $\boldsymbol{x} \in A$ achieving the optimal value is a **(global) minimizer** of the problem.

Convex optimization problems

Standard form of a **convex optimization problem**:

$$\begin{aligned} \min_{\boldsymbol{x} \in \mathbb{R}^d} \quad & f_0(\boldsymbol{x}) \\ \text{s.t.} \quad & f_i(\boldsymbol{x}) \leq 0 \quad i = 1, 2, \ldots, n \end{aligned}$$

where $f_0, f_1, \ldots, f_n : \mathbb{R}^d \to \mathbb{R}$ are *convex functions*.

Standard form of a **convex optimization problem**:

$$\begin{aligned} \min_{\boldsymbol{x} \in \mathbb{R}^d} \quad & f_0(\boldsymbol{x}) \\ \text{s.t.} \quad & f_i(\boldsymbol{x}) \leq 0 \quad i = 1, 2, \dots, n \end{aligned}$$

where $f_0, f_1, \dots, f_n \colon \mathbb{R}^d \to \mathbb{R}$ are *convex functions*.

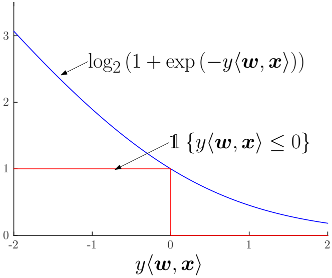
Fact: the feasible set $A := \{\boldsymbol{x} \in \mathbb{R}^d : f_i(\boldsymbol{x}) \leq 0 \text{ for all } i = 1, 2, \dots, n\}$ is a convex set.

Example: non-negative logistic regression

Training data $S := ((\boldsymbol{x}^{(1)}, y^{(1)}), \ldots, (\boldsymbol{x}^{(n)}, y^{(n)}))$ with $(\boldsymbol{x}^{(i)}, y^{(i)}) \in \mathbb{R}^d \times \{\pm 1\}$.

$$\min_{\boldsymbol{w} \in \mathbb{R}^d} \quad f(\boldsymbol{w}) := \frac{1}{n} \sum_{i=1}^{n} \ln\Big(1 + \exp\Big(-y^{(i)} \langle \boldsymbol{w}, \boldsymbol{x}^{(i)} \rangle\Big)\Big)$$

$$\text{s.t.} \quad w_i \geq 0 \quad \text{for } i = 1, 2, \ldots, d.$$



Like SVM, but using a different surrogate loss, no regularization term, and only want non-negative weights.

Example: non-negative logistic regression

Objective function:

$$f_0(\boldsymbol{w}) := \frac{1}{n} \sum_{i=1}^{n} \ln\Big(1 + \exp\Big(-y^{(i)} \langle \boldsymbol{w}, \boldsymbol{x}^{(i)} \rangle\Big)\Big).$$

Example: non-negative logistic regression

Objective function:

$$f_0(\boldsymbol{w}) := \frac{1}{n} \sum_{i=1}^{n} \ln\Big(1 + \exp\Big(-y^{(i)} \langle \boldsymbol{w}, \boldsymbol{x}^{(i)} \rangle\Big)\Big).$$

Second-derivative test of convexity: Is $\nabla^2 f_0(\boldsymbol{w})$ positive semidefinite?

$$\nabla^2 f_0(\boldsymbol{w}) = \frac{1}{n} \sum_{i=1}^{n} \frac{1}{1 + e^{\langle \boldsymbol{w}, \boldsymbol{x}^{(i)} \rangle}} \cdot \frac{e^{\langle \boldsymbol{w}, \boldsymbol{x}^{(i)} \rangle}}{1 + e^{\langle \boldsymbol{w}, \boldsymbol{x}^{(i)} \rangle}} \boldsymbol{x}^{(i)} (\boldsymbol{x}^{(i)})^\top.$$

Objective function:

$$f_0(\boldsymbol{w}) := \frac{1}{n}\sum_{i=1}^{n} \ln\Big(1 + \exp\Big(-y^{(i)}\langle \boldsymbol{w}, \boldsymbol{x}^{(i)}\rangle\Big)\Big).$$

Second-derivative test of convexity: Is $\nabla^2 f_0(\boldsymbol{w})$ positive semidefinite?

$$\nabla^2 f_0(\boldsymbol{w}) = \frac{1}{n}\sum_{i=1}^{n} \frac{1}{1+e^{\langle \boldsymbol{w}, \boldsymbol{x}^{(i)}\rangle}} \cdot \frac{e^{\langle \boldsymbol{w}, \boldsymbol{x}^{(i)}\rangle}}{1+e^{\langle \boldsymbol{w}, \boldsymbol{x}^{(i)}\rangle}} \boldsymbol{x}^{(i)}(\boldsymbol{x}^{(i)})^\top.$$

Yes: for any $\boldsymbol{v} \in \mathbb{R}^d$,

$$\langle \nabla^2 f_0(\boldsymbol{w})\boldsymbol{v}, \boldsymbol{v}\rangle = \frac{1}{n}\sum_{i=1}^{n} \frac{1}{1+e^{\langle \boldsymbol{w}, \boldsymbol{x}^{(i)}\rangle}} \cdot \frac{e^{\langle \boldsymbol{w}, \boldsymbol{x}^{(i)}\rangle}}{1+e^{\langle \boldsymbol{w}, \boldsymbol{x}^{(i)}\rangle}} \langle \boldsymbol{v}, \boldsymbol{x}^{(i)}\rangle^2 \geq 0.$$

Example: non-negative logistic regression

Objective function:

$$f_0(\boldsymbol{w}) := \frac{1}{n}\sum_{i=1}^{n} \ln\Big(1 + \exp\Big(-y^{(i)}\langle \boldsymbol{w}, \boldsymbol{x}^{(i)}\rangle\Big)\Big).$$

Second-derivative test of convexity: Is $\nabla^2 f_0(\boldsymbol{w})$ positive semidefinite?

$$\nabla^2 f_0(\boldsymbol{w}) = \frac{1}{n}\sum_{i=1}^{n} \frac{1}{1 + e^{\langle \boldsymbol{w}, \boldsymbol{x}^{(i)}\rangle}} \cdot \frac{e^{\langle \boldsymbol{w}, \boldsymbol{x}^{(i)}\rangle}}{1 + e^{\langle \boldsymbol{w}, \boldsymbol{x}^{(i)}\rangle}} \boldsymbol{x}^{(i)} (\boldsymbol{x}^{(i)})^\top.$$

Yes: for any $\boldsymbol{v} \in \mathbb{R}^d$,

$$\boxed{\langle \nabla^2 f_0(\boldsymbol{w})\boldsymbol{v}, \boldsymbol{v}\rangle = \frac{1}{n}\sum_{i=1}^{n} \frac{1}{1 + e^{\langle \boldsymbol{w}, \boldsymbol{x}^{(i)}\rangle}} \cdot \frac{e^{\langle \boldsymbol{w}, \boldsymbol{x}^{(i)}\rangle}}{1 + e^{\langle \boldsymbol{w}, \boldsymbol{x}^{(i)}\rangle}} \langle \boldsymbol{v}, \boldsymbol{x}^{(i)}\rangle^2 \geq 0.}$$

Conclusion: objective function $f_0(\boldsymbol{w})$ is convex.

Example: non-negative logistic regression

Constraints:

$$w_i \geq 0 \quad \text{for } i = 1, 2, \ldots, d.$$

Example: non-negative logistic regression

Constraints:

$$w_i \geq 0 \quad \text{for } i = 1, 2, \ldots, d.$$

In standard form: $f_i(\boldsymbol{w}) \leq 0$, where

$$f_i(\boldsymbol{w}) := -w_i \quad \text{for } i = 1, 2, \ldots, d.$$

Example: non-negative logistic regression

Constraints:

$$w_i \geq 0 \quad \text{for } i = 1, 2, \ldots, d.$$

In standard form: $f_i(\boldsymbol{w}) \leq 0$, where

$$f_i(\boldsymbol{w}) := -w_i \quad \text{for } i = 1, 2, \ldots, d.$$

Each f_i is a linear function of $\boldsymbol{w}$, and therefore is convex.

Example: non-negative logistic regression

Constraints:

$$w_i \geq 0 \quad \text{for } i = 1, 2, \ldots, d.$$

In standard form: $f_i(\boldsymbol{w}) \leq 0$, where

$$f_i(\boldsymbol{w}) := -w_i \quad \text{for } i = 1, 2, \ldots, d.$$

Each f_i is a linear function of $\boldsymbol{w}$, and therefore is convex.

Conclusion: constraint functions $f_i(\boldsymbol{w})$ are convex.

Example: non-negative logistic regression

Constraints:

$$w_i \geq 0 \quad \text{for } i = 1, 2, \ldots, d.$$

In standard form: $f_i(\boldsymbol{w}) \leq 0$, where

$$f_i(\boldsymbol{w}) := -w_i \quad \text{for } i = 1, 2, \ldots, d.$$

Each f_i is a linear function of $\boldsymbol{w}$, and therefore is convex.

Conclusion: constraint functions $f_i(\boldsymbol{w})$ are convex.

Overall problem is a convex optimization problem. $\square$

Local minimizers

Consider an optimization problem (not necessarily convex):

$$\begin{aligned} \min_{\boldsymbol{x} \in \mathbb{R}^d} \quad & f_0(\boldsymbol{x}) \\ \text{s.t.} \quad & \boldsymbol{x} \in A. \end{aligned}$$

Local minimizers

Consider an optimization problem (not necessarily convex):

$$\begin{aligned} \min_{\boldsymbol{x} \in \mathbb{R}^d} \quad & f_0(\boldsymbol{x}) \\ \text{s.t.} \quad & \boldsymbol{x} \in A. \end{aligned}$$

We say $\tilde{\boldsymbol{x}} \in A$ is a **local minimizer** if there is an open ball

$$U := \left\{ \boldsymbol{x} \in \mathbb{R}^d : \|\boldsymbol{x} - \tilde{\boldsymbol{x}}\|_2 < r \right\}$$

of positive radius $r > 0$ such that $\tilde{\boldsymbol{x}}$ is a global minimizer for

$$\begin{aligned} \min_{\boldsymbol{x} \in \mathbb{R}^d} \quad & f_0(\boldsymbol{x}) \\ \text{s.t.} \quad & \boldsymbol{x} \in A \cap U. \end{aligned}$$

Local minimizers

Consider an optimization problem (not necessarily convex):

$$\begin{aligned} \min_{\boldsymbol{x}\in\mathbb{R}^d} \quad & f_0(\boldsymbol{x}) \\ \text{s.t.} \quad & \boldsymbol{x} \in A. \end{aligned}$$

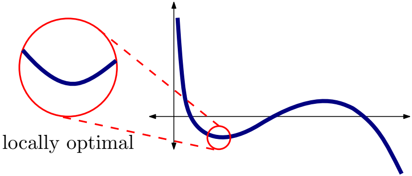
We say $\tilde{\boldsymbol{x}} \in A$ is a **local minimizer** if there is an open ball

$$U := \left\{ \boldsymbol{x} \in \mathbb{R}^d : \|\boldsymbol{x} - \tilde{\boldsymbol{x}}\|_2 < r \right\}$$

of positive radius $r > 0$ such that $\tilde{\boldsymbol{x}}$ is a global minimizer for

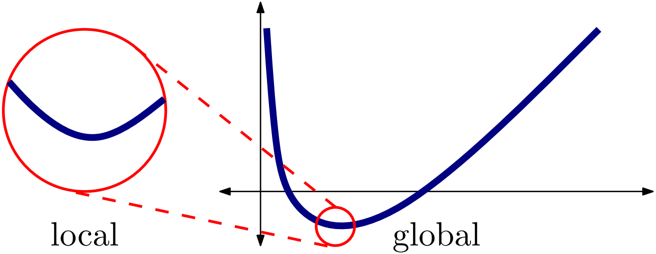
$$\begin{aligned} \min_{\boldsymbol{x}\in\mathbb{R}^d} \quad & f_0(\boldsymbol{x}) \\ \text{s.t.} \quad & \boldsymbol{x} \in A \cap U. \end{aligned}$$

Nothing looks better than $\tilde{\boldsymbol{x}}$ in the immediate vicinity of $\tilde{\boldsymbol{x}}$.



LOCAL-TO-GLOBAL PHENOMENON

If the optimization problem is convex, and $\tilde{\boldsymbol{x}} \in A$ is a local minimizer,
then **it is also a global minimizer**.



Solving convex optimization problems

Unconstrained convex optimization problems

$$\min_{\boldsymbol{x} \in \mathbb{R}^d} \qquad f(\boldsymbol{x})$$

(f is the convex objective function).

Unconstrained convex optimization problems

$$\min_{\boldsymbol{x} \in \mathbb{R}^d} \quad f(\boldsymbol{x})$$

(f is the convex objective function).

Optimality condition for differentiable convex objectives

$$\boldsymbol{x} \text{ is a global minimizer if and only if } \boxed{\nabla f(\boldsymbol{x}) = \mathbf{0}}.$$

guarantee convex

Local optimization for convex objectives

Local optimization

Main idea: locally change $\boldsymbol{x} \to \boldsymbol{x} + \boldsymbol{\delta}$ to improve its objective value $f(\boldsymbol{x}) \to f(\boldsymbol{x} + \boldsymbol{\delta})$.

Local optimization

Main idea: locally change $\boldsymbol{x} \to \boldsymbol{x} + \boldsymbol{\delta}$ to improve its objective value $f(\boldsymbol{x}) \to f(\boldsymbol{x} + \boldsymbol{\delta})$. **What should δ be?**

Local optimization

Main idea: locally change $\boldsymbol{x} \to \boldsymbol{x} + \boldsymbol{\delta}$ to improve its objective value $f(\boldsymbol{x}) \to f(\boldsymbol{x} + \boldsymbol{\delta})$. **What should δ be?**

By convexity of f: $\quad f(\boldsymbol{x} + \boldsymbol{\delta}) \geq f(\boldsymbol{x}) + \langle \nabla f(\boldsymbol{x}), \boldsymbol{\delta} \rangle.$

Local optimization

Main idea: locally change $\boldsymbol{x} \to \boldsymbol{x} + \boldsymbol{\delta}$ to improve its objective value $f(\boldsymbol{x}) \to f(\boldsymbol{x} + \boldsymbol{\delta})$. **What should δ be?**

By convexity of f: $\quad f(\boldsymbol{x} + \boldsymbol{\delta}) \geq f(\boldsymbol{x}) + \langle \nabla f(\boldsymbol{x}), \boldsymbol{\delta} \rangle.$

If $\langle \nabla f(\boldsymbol{x}), \boldsymbol{\delta} \rangle \geq 0$, then

$$f(\boldsymbol{x} + \boldsymbol{\delta}) \geq f(\boldsymbol{x}).$$

Local optimization

Main idea: locally change $\boldsymbol{x} \to \boldsymbol{x} + \boldsymbol{\delta}$ to improve its objective value $f(\boldsymbol{x}) \to f(\boldsymbol{x} + \boldsymbol{\delta})$. **What should δ be?**

By convexity of f: $\quad f(\boldsymbol{x} + \boldsymbol{\delta}) \geq f(\boldsymbol{x}) + \langle \nabla f(\boldsymbol{x}), \boldsymbol{\delta} \rangle.$

If $\langle \nabla f(\boldsymbol{x}), \boldsymbol{\delta} \rangle \geq 0$, then

$$f(\boldsymbol{x} + \boldsymbol{\delta}) \geq f(\boldsymbol{x}). \quad \text{Oops!}$$

Local optimization

Main idea: locally change $\boldsymbol{x} \to \boldsymbol{x} + \boldsymbol{\delta}$ to improve its objective value $f(\boldsymbol{x}) \to f(\boldsymbol{x} + \boldsymbol{\delta})$. **What should δ be?**

By convexity of f: $\quad f(\boldsymbol{x} + \boldsymbol{\delta}) \geq f(\boldsymbol{x}) + \langle \nabla f(\boldsymbol{x}), \boldsymbol{\delta} \rangle$.

If $\langle \nabla f(\boldsymbol{x}), \boldsymbol{\delta} \rangle \geq 0$, then

$$f(\boldsymbol{x} + \boldsymbol{\delta}) \geq f(\boldsymbol{x}). \quad \text{Oops!}$$

Moral: to be useful, the change $\boldsymbol{\delta}$ must satisfy

$$\langle \nabla f(\boldsymbol{x}), \boldsymbol{\delta} \rangle < 0.$$

For example, $\boldsymbol{\delta} := -\eta \nabla f(\boldsymbol{x})$ for some $\eta > 0$:

$$\langle \nabla f(\boldsymbol{x}), -\eta \nabla f(\boldsymbol{x}) \rangle \;=\; -\eta \|\nabla f(\boldsymbol{x})\|_2^2 \;<\; 0$$

as long as $\nabla f(\boldsymbol{x}) \neq \mathbf{0}$.

Gradient descent for differentiable objectives

- Start with some initial $\boldsymbol{x}^{(1)} \in \mathbb{R}^d$.
- For $t = 1, 2, \ldots$ until some stopping condition is satisfied.
  - Compute gradient of f at $\boldsymbol{x}^{(t)}$:

$$\boldsymbol{\lambda}^{(t)} := \nabla f(\boldsymbol{x}^{(t)}).$$

  - Update:

$$\boldsymbol{x}^{(t+1)} := \boldsymbol{x}^{(t)} - \eta_t \boldsymbol{\lambda}^{(t)}.$$

Gradient descent for differentiable objectives

- Start with some initial $\boldsymbol{x}^{(1)} \in \mathbb{R}^d$.
- For $t = 1, 2, \ldots$ until some stopping condition is satisfied.
  - Compute gradient of f at $\boldsymbol{x}^{(t)}$:

$$\boldsymbol{\lambda}^{(t)} := \nabla f(\boldsymbol{x}^{(t)}).$$

  - Update:

$$\boldsymbol{x}^{(t+1)} := \boldsymbol{x}^{(t)} - \eta_t \boldsymbol{\lambda}^{(t)}.$$

Here, $\eta_1, \eta_2, \ldots > 0$ are the **step sizes**.

Gradient descent for differentiable objectives

- Start with some initial $\boldsymbol{x}^{(1)} \in \mathbb{R}^d$.
- For $t = 1, 2, \ldots$ until some stopping condition is satisfied.
  - Compute gradient of f at $\boldsymbol{x}^{(t)}$:
  $$\boldsymbol{\lambda}^{(t)} := \nabla f(\boldsymbol{x}^{(t)}).$$
  - Update:
  $$\boldsymbol{x}^{(t+1)} := \boldsymbol{x}^{(t)} - \eta_t \boldsymbol{\lambda}^{(t)}.$$

Here, $\eta_1, \eta_2, \ldots > 0$ are the **step sizes**. Common choices include:

1. Set $\eta_t := c$ for some constant $c > 0$.
2. Set $\eta_t := c/\sqrt{t}$ for some constant $c > 0$.
3. Set η_t using a line search procedure.

Backtracking line search

Goal: given $\boldsymbol{x} \in \mathbb{R}^d$ and $\boldsymbol{\lambda} = \nabla f(\boldsymbol{x}) \in \mathbb{R}^d$, find $\eta > 0$ so that $f(\boldsymbol{x} - \eta\boldsymbol{\lambda}) < f(\boldsymbol{x})$ by a reasonable amount.

- Start with $\eta := 1$.
- While $f(\boldsymbol{x} - \eta\boldsymbol{\lambda}) > f(\boldsymbol{x}) - \frac{1}{2}\eta\|\boldsymbol{\lambda}\|_2^2$: Set $\eta := \frac{1}{2}\eta$.

Backtracking line search

Goal: given $\boldsymbol{x} \in \mathbb{R}^d$ and $\boldsymbol{\lambda} = \nabla f(\boldsymbol{x}) \in \mathbb{R}^d$, find $\eta > 0$ so that $f(\boldsymbol{x} - \eta\boldsymbol{\lambda}) < f(\boldsymbol{x})$ by a reasonable amount.

- Start with $\eta := 1$.
- While $f(\boldsymbol{x} - \eta\boldsymbol{\lambda}) > f(\boldsymbol{x}) - \frac{1}{2}\eta\|\boldsymbol{\lambda}\|_2^2$: Set $\eta := \frac{1}{2}\eta$.

Main idea: $f(\boldsymbol{x} - \eta\boldsymbol{\lambda}) \approx f(\boldsymbol{x}) - \eta\|\boldsymbol{\lambda}\|_2^2$, so optimistically hope to decrease value by about $\eta\|\boldsymbol{\lambda}\|_2^2$ when η is small enough.

Backtracking line search

Goal: given $\boldsymbol{x} \in \mathbb{R}^d$ and $\boldsymbol{\lambda} = \nabla f(\boldsymbol{x}) \in \mathbb{R}^d$, find $\eta > 0$ so that $f(\boldsymbol{x} - \eta\boldsymbol{\lambda}) < f(\boldsymbol{x})$ by a reasonable amount.

- Start with $\eta := 1$.
- While $f(\boldsymbol{x} - \eta\boldsymbol{\lambda}) > f(\boldsymbol{x}) - \frac{1}{2}\eta\|\boldsymbol{\lambda}\|_2^2$:   Set $\eta := \frac{1}{2}\eta$.

Main idea: $f(\boldsymbol{x} - \eta\boldsymbol{\lambda}) \approx f(\boldsymbol{x}) - \eta\|\boldsymbol{\lambda}\|_2^2$, so optimistically hope to decrease value by about $\eta\|\boldsymbol{\lambda}\|_2^2$ when η is small enough.

Settle for decreasing by $\frac{1}{2}\eta\|\boldsymbol{\lambda}\|_2^2$: upon termination,

$$f(\boldsymbol{x} - \eta\boldsymbol{\lambda}) \;\le\; f(\boldsymbol{x}) - \frac{1}{2}\eta\|\boldsymbol{\lambda}\|_2^2.$$

Backtracking line search

Goal: given $\boldsymbol{x} \in \mathbb{R}^d$ and $\boldsymbol{\lambda} = \nabla f(\boldsymbol{x}) \in \mathbb{R}^d$, find $\eta > 0$ so that $f(\boldsymbol{x} - \eta\boldsymbol{\lambda}) < f(\boldsymbol{x})$ by a reasonable amount.

- Start with $\eta := 1$.
- While $f(\boldsymbol{x} - \eta\boldsymbol{\lambda}) > f(\boldsymbol{x}) - \frac{1}{2}\eta\|\boldsymbol{\lambda}\|_2^2$: Set $\eta := \frac{1}{2}\eta$.

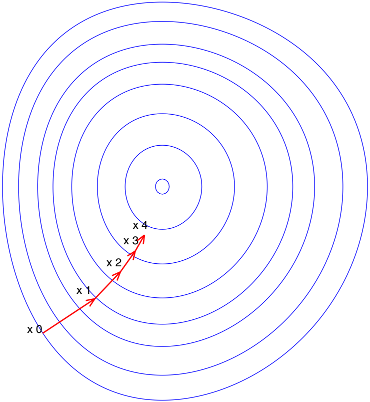
Main idea: $f(\boldsymbol{x} - \eta\boldsymbol{\lambda}) \approx f(\boldsymbol{x}) - \eta\|\boldsymbol{\lambda}\|_2^2$, so optimistically hope to decrease value by about $\eta\|\boldsymbol{\lambda}\|_2^2$ when η is small enough.

Settle for decreasing by $\frac{1}{2}\eta\|\boldsymbol{\lambda}\|_2^2$: upon termination,

$$f(\boldsymbol{x} - \eta\boldsymbol{\lambda}) \;\leq\; f(\boldsymbol{x}) - \frac{1}{2}\eta\|\boldsymbol{\lambda}\|_2^2.$$

Many other line search methods are possible.

Illustration of gradient descent

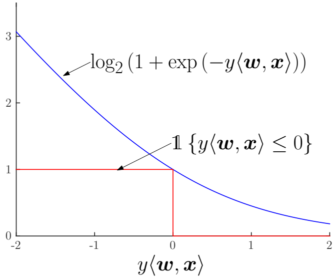


If f is convex (and satisfies some other smoothness and curvature conditions), then $f(\boldsymbol{x}^{(t)})$ converges to the optimal value at a geometric rate.

Example: (unconstrained) logistic regression

Training data $S := ((\boldsymbol{x}^{(1)}, y^{(1)}), \ldots, (\boldsymbol{x}^{(n)}, y^{(n)}))$ with $(\boldsymbol{x}^{(i)}, y^{(i)}) \in \mathbb{R}^d \times \{\pm 1\}$.

$$\min_{\boldsymbol{w} \in \mathbb{R}^d} \qquad f(\boldsymbol{w}) := \frac{1}{n} \sum_{i=1}^{n} \ln\Big(1 + \exp\Big(-y^{(i)} \langle \boldsymbol{w}, \boldsymbol{x}^{(i)} \rangle\Big)\Big)$$

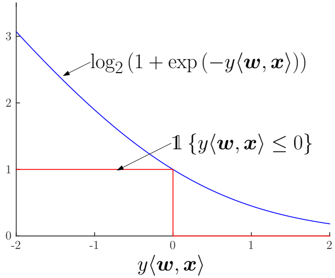


We've already established that objective $f(\boldsymbol{w})$ is convex.

Example: (unconstrained) logistic regression

Training data $S := ((\boldsymbol{x}^{(1)}, y^{(1)}), \ldots, (\boldsymbol{x}^{(n)}, y^{(n)}))$ with $(\boldsymbol{x}^{(i)}, y^{(i)}) \in \mathbb{R}^d \times \{\pm 1\}$.

$$\min_{\boldsymbol{w} \in \mathbb{R}^d} \qquad f(\boldsymbol{w}) := \frac{1}{n} \sum_{i=1}^{n} \ln\Big(1 + \exp\Big(-y^{(i)} \langle \boldsymbol{w}, \boldsymbol{x}^{(i)} \rangle\Big)\Big)$$



We've already established that objective $f(\boldsymbol{w})$ is convex.

Question: How do we compute its gradient at a given point $\boldsymbol{w} \in \mathbb{R}^d$?

Example: (unconstrained) logistic regression

Gradient of f at $\boldsymbol{w}$:

$$\nabla f(\boldsymbol{w}) = -\frac{1}{n}\sum_{i=1}^{n}\frac{1}{1+e^{y^{(i)}\langle \boldsymbol{w},\boldsymbol{x}^{(i)}\rangle}}y^{(i)}\boldsymbol{x}^{(i)}$$

Example: (unconstrained) logistic regression

Gradient of f at $\boldsymbol{w}$:

$$\nabla f(\boldsymbol{w}) = -\frac{1}{n} \sum_{i=1}^{n} \frac{1}{1 + e^{y^{(i)} \langle \boldsymbol{w}, \boldsymbol{x}^{(i)} \rangle}} y^{(i)} \boldsymbol{x}^{(i)}$$

Gradient descent algorithm:

- Start with some initial $\boldsymbol{w}^{(1)} \in \mathbb{R}^d$.
- For $t = 1, 2, \ldots$ until some stopping condition is satisfied.

$$\begin{aligned}
\boldsymbol{w}^{(t+1)} &:= \boldsymbol{w}^{(t)} - \eta_t \nabla f(\boldsymbol{w}^{(t)}) \\
&= \boldsymbol{w}^{(t)} + \eta_t \frac{1}{n} \sum_{i=1}^{n} \frac{1}{1 + e^{y^{(i)} \langle \boldsymbol{w}, \boldsymbol{x}^{(i)} \rangle}} y^{(i)} \boldsymbol{x}^{(i)}.
\end{aligned}$$

Stopping condition

- In many applications of (convex) optimization, care about solving problems to very high precision.

  Example: stop when gradient is close enough to zero ($\|\nabla f(\boldsymbol{x})\|_2 \le \epsilon$ for some small parameter $\epsilon > 0$).

Stopping condition

- In many applications of (convex) optimization, care about solving problems to very high precision.

  Example: stop when gradient is close enough to zero ($\|\nabla f(\boldsymbol{x})\|_2 \leq \epsilon$ for some small parameter $\epsilon > 0$).

- For machine learning applications: optimization problem based on training data often just a means-to-an-end.

Stopping condition

- In many applications of (convex) optimization, care about solving problems to very high precision.

  Example: stop when gradient is close enough to zero ($\|\nabla f(\boldsymbol{x})\|_2 \leq \epsilon$ for some small parameter $\epsilon > 0$).

- For machine learning applications: optimization problem based on training data often just a means-to-an-end.

  We really just care about true error.

Stopping condition

- In many applications of (convex) optimization, care about solving problems to very high precision.

  Example: stop when gradient is close enough to zero ($\|\nabla f(\boldsymbol{x})\|_2 \leq \epsilon$ for some small parameter $\epsilon > 0$).

- For machine learning applications: optimization problem based on training data often just a means-to-an-end.

  We really just care about true error.

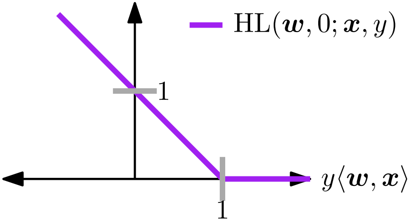
- Running gradient descent to convergence not strictly necessary: **may be beneficial to stop early (e.g., based on hold-out error).**

Non-differentiable convex objectives

Some convex functions f are not differentiable everywhere;
gradient descent not even well-specified for these problems.

Example: hinge loss

$$f(\boldsymbol{w}) = \mathrm{HL}(\boldsymbol{w}, 0; \boldsymbol{x}, y) = \Big[1 - y\langle \boldsymbol{w}, \boldsymbol{x}\rangle\Big]_+ .$$



Not differentiable at $\boldsymbol{w} \in \mathbb{R}^d$ where $y\langle \boldsymbol{w}, \boldsymbol{x}\rangle = 1$.

Subgradients

Although not every function f is differentiable everywhere,
every **convex function** f has **subgradients** *everywhere*†.

We say $\boldsymbol{\lambda} \in \mathbb{R}^d$ is a **subgradient** of a function $f\colon \mathbb{R}^d \to \mathbb{R}$ at $\boldsymbol{x}_0 \in \mathbb{R}^d$ if

$$f(\boldsymbol{x}) \geq f(\boldsymbol{x}_0) + \langle \boldsymbol{\lambda}, \boldsymbol{x} - \boldsymbol{x}_0 \rangle \quad \forall \boldsymbol{x} \in \mathbb{R}^d.$$

†some technical conditions apply.

Subgradients

Although not every function f is differentiable everywhere,
every **convex function** f has **subgradients** *everywhere*[†].

We say $\boldsymbol{\lambda} \in \mathbb{R}^d$ is a **subgradient** of a function $f\colon \mathbb{R}^d \to \mathbb{R}$ at $\boldsymbol{x}_0 \in \mathbb{R}^d$ if

$$f(\boldsymbol{x}) \geq f(\boldsymbol{x}_0) + \langle \boldsymbol{\lambda}, \boldsymbol{x} - \boldsymbol{x}_0 \rangle \quad \forall \boldsymbol{x} \in \mathbb{R}^d.$$

In other words, a subgradient of a convex function f at a point $\boldsymbol{x}_0$ specifies an **affine lower bound** on the function ...

[†]some technical conditions apply.

Subgradients

Although not every function f is differentiable everywhere,
every **convex function** f has **subgradients** *everywhere*†.

We say $\boldsymbol{\lambda} \in \mathbb{R}^d$ is a **subgradient** of a function $f \colon \mathbb{R}^d \to \mathbb{R}$ at $\boldsymbol{x}_0 \in \mathbb{R}^d$ if

$$f(\boldsymbol{x}) \geq f(\boldsymbol{x}_0) + \langle \boldsymbol{\lambda}, \boldsymbol{x} - \boldsymbol{x}_0 \rangle \quad \forall \boldsymbol{x} \in \mathbb{R}^d.$$

In other words, a subgradient of a convex function f at a point $\boldsymbol{x}_0$ specifies an **affine lower bound** on the function ...

... **just like the gradient in the case of a differentiable function**:

$$f(\boldsymbol{x}) \geq f(\boldsymbol{x}_0) + \langle \nabla f(\boldsymbol{x}_0), \boldsymbol{x} - \boldsymbol{x}_0 \rangle \quad \forall \boldsymbol{x} \in \mathbb{R}^d.$$

†some technical conditions apply.

Subgradients

Although not every function f is differentiable everywhere,
every **convex function** f has **subgradients** *everywhere*†.

We say $\boldsymbol{\lambda} \in \mathbb{R}^d$ is a **subgradient** of a function $f\colon \mathbb{R}^d \to \mathbb{R}$ at $\boldsymbol{x}_0 \in \mathbb{R}^d$ if

$$f(\boldsymbol{x}) \geq f(\boldsymbol{x}_0) + \langle \boldsymbol{\lambda}, \boldsymbol{x} - \boldsymbol{x}_0 \rangle \quad \forall \boldsymbol{x} \in \mathbb{R}^d.$$

In other words, a subgradient of a convex function f at a point $\boldsymbol{x}_0$ specifies an **affine lower bound** on the function ...

... **just like the gradient in the case of a differentiable function**:

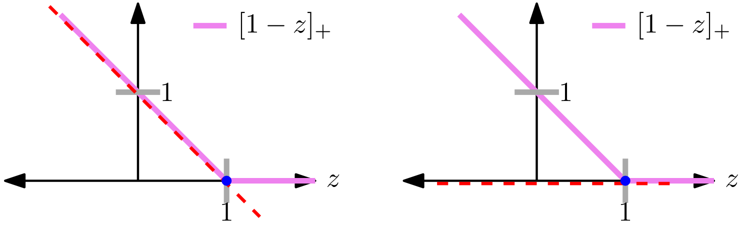
$$f(\boldsymbol{x}) \geq f(\boldsymbol{x}_0) + \langle \nabla f(\boldsymbol{x}_0), \boldsymbol{x} - \boldsymbol{x}_0 \rangle \quad \forall \boldsymbol{x} \in \mathbb{R}^d.$$

There might be many subgradients at a given point x_0—i.e., many affine lower bounds: call the entire set the **subdifferential of f at x_0**, $\partial f(\boldsymbol{x}_0)$.

†some technical conditions apply.

Example: subgradient of hinge loss

Consider one-dimensional function $f(z) := [1 - z]_+ = \max\{0, 1 - z\}$.

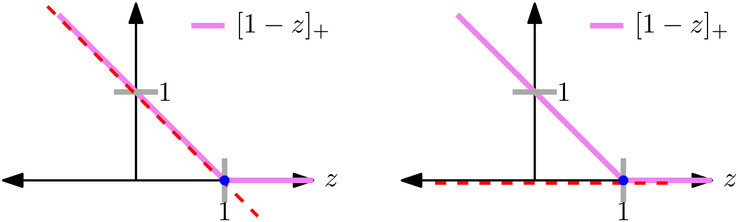


Two subgradients of f at $z = 1$: -1 and 0.

$$f(z) \;\geq\; f(1) + (-1) \cdot (z - 1) \;=\; 1 - z\;;$$
$$f(z) \;\geq\; f(1) + (0) \cdot (z - 1) \;=\; 0.$$

Example: subgradient of hinge loss

Consider one-dimensional function $f(z) := [1-z]_+ = \max\{0, 1-z\}$.



Two subgradients of f at $z = 1$: $\boxed{-1 \text{ and } 0.}$

$$f(z) \;\geq\; f(1) + (-1)\cdot(z-1) \;=\; 1 - z\;;$$
$$f(z) \;\geq\; f(1) + (0)\cdot(z-1) \;=\; 0.$$

Actually, **infinitely-many subgradients**: $\boxed{\text{all } \lambda \in [-1, 0] \text{ satisfy}}$

$$f(z) \;\geq\; f(1) + \lambda\cdot(z-1).$$

Subgradient calculus

Suppose $g, g_1, g_2, \ldots, g_n$ are convex functions.
Below, sufficient conditions under which f is convex, and corresponding subdifferential:

- **Addition**: If $f(\boldsymbol{x}) = g_1(\boldsymbol{x}) + g_2(\boldsymbol{x})$, then $\partial f(\boldsymbol{x}) = \partial g_1(\boldsymbol{x}) + \partial g_2(\boldsymbol{x})$.
- **Positive scaling**: If $f(\boldsymbol{x}) = \alpha \cdot g(\boldsymbol{x})$ for some $\alpha > 0$, then $\partial f(\boldsymbol{x}) = \alpha \cdot \partial g(\boldsymbol{x})$.
- **Affine composition**: If $f(\boldsymbol{x}) = g(\boldsymbol{A}\boldsymbol{x} + \boldsymbol{b})$, then $\partial f(\boldsymbol{x}) = \boldsymbol{A}^\top \partial g(\boldsymbol{A}\boldsymbol{x} + \boldsymbol{b})$.
- **Finite pointwise maximum**: If $f(\boldsymbol{x}) = \max_{i \in [n]} g_i(\boldsymbol{x})$, then

$$\partial f(\boldsymbol{x}) = \operatorname{conv}\left(\bigcup_{i \in [n] : g_i(\boldsymbol{x}) = f(\boldsymbol{x})} \partial g_i(\boldsymbol{x}) \right).$$

- **More general composition**: If $h \colon \mathbb{R}^n \to \mathbb{R}$ is convex and non-decreasing in each argument, and $f(\boldsymbol{x}) := h(g_1(\boldsymbol{x}), g_2(\boldsymbol{x}), \ldots, g_n(\boldsymbol{x}))$, then

$$\partial f(\boldsymbol{x}) = \bigcup_{(\lambda_1, \lambda_2, \ldots, \lambda_n) \in \partial h(g_1(\boldsymbol{x}), g_2(\boldsymbol{x}), \ldots, g_n(\boldsymbol{x}))} \partial\left(\sum_{i=1}^n \lambda_i g_i(\boldsymbol{x}) \right).$$

Subgradient descent for general convex objectives

- Start with some initial $\boldsymbol{x}^{(1)} \in \mathbb{R}^d$.
- For $t = 1, 2, \ldots$ until some stopping condition is satisfied.
  - Compute *any* subgradient $\boldsymbol{\lambda}^{(t)}$ of f at $\boldsymbol{x}^{(t)}$.
  - Update:

$$\boldsymbol{x}^{(t+1)} := \boldsymbol{x}^{(t)} - \eta_t \boldsymbol{\lambda}^{(t)}.$$