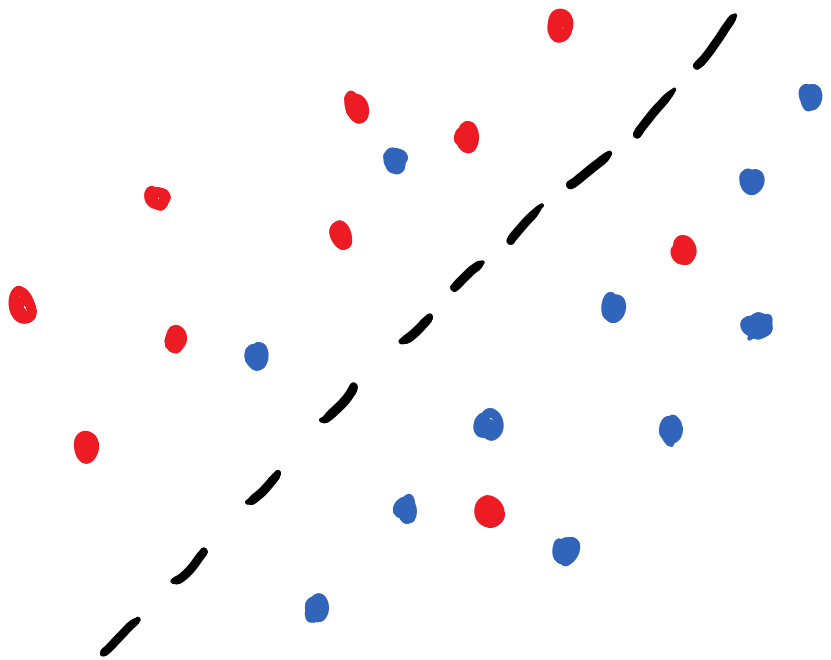
COMS 4771 Lecture 9

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?

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 $C > 0$:

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

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".

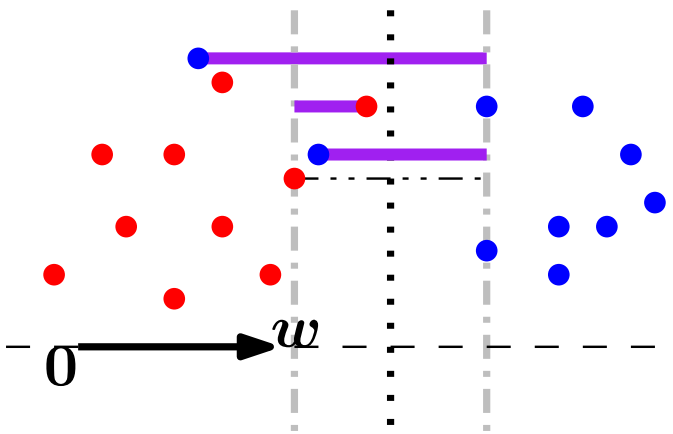
Other winners include: public key cryptography, Lempel-Ziv compression, Splay Trees, interior point method for linear programming, ...
and AdaBoost (discussed later in the course).

Slack interpretation

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

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

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

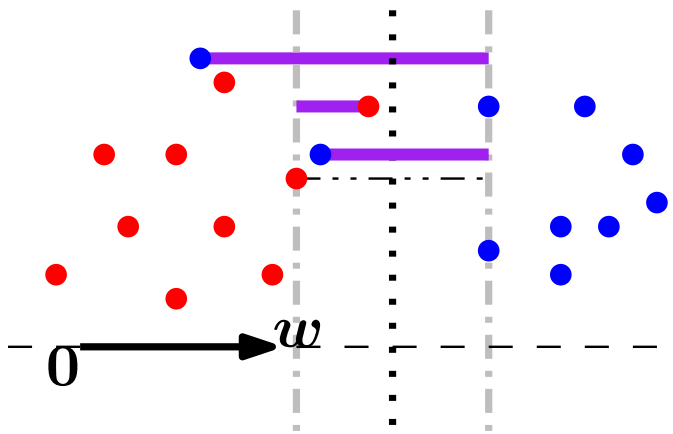


Slack interpretation

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

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

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



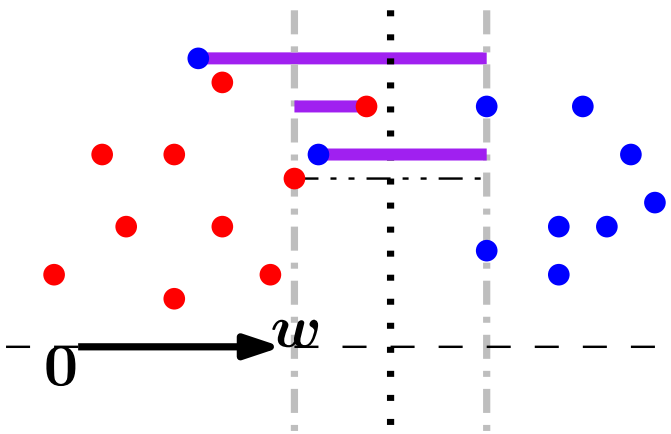
- For a given $(\boldsymbol{w}, \theta)$, $\xi_i/\|\boldsymbol{w}\|_2$ measures distance that $\boldsymbol{x}^{(i)}$ must be moved so that $y^{(i)}\langle \boldsymbol{w}, \boldsymbol{x}^{(i)} - \theta\rangle \geq 1$.

Slack interpretation

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

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

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



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

A different interpretation of slack

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

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

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

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

A different interpretation of slack

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

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

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

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

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 $\lambda = 1/(nC)$)

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

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

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

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

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

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

A different interpretation of slack

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

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

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

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

Equivalent unconstrained form:

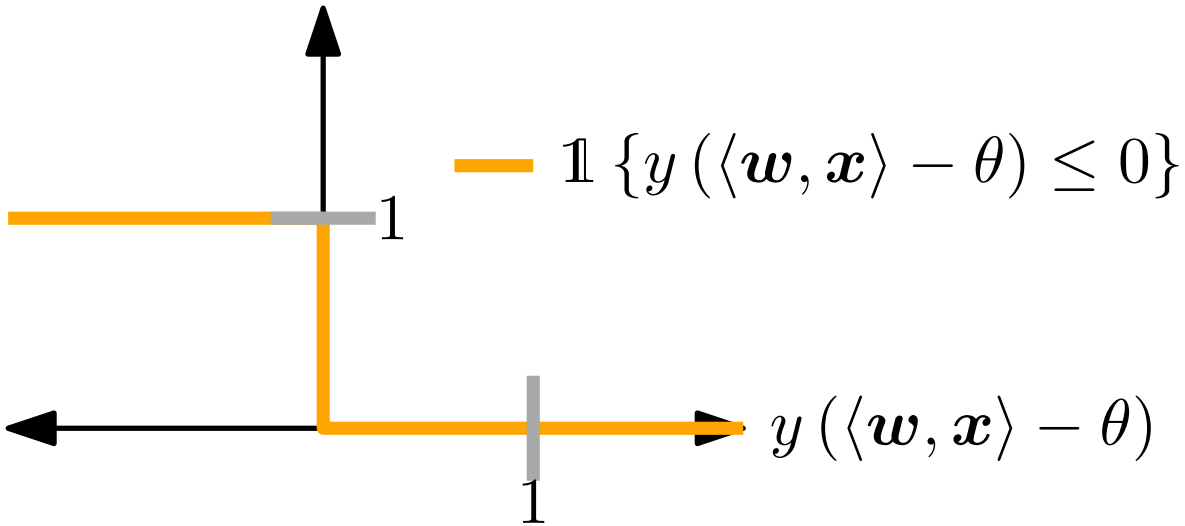
$$\min_{\boldsymbol{w}\in\mathbb{R}^d,\theta\in\mathbb{R}} \quad \frac{\lambda}{2}\|\boldsymbol{w}\|_2^2 + \frac{1}{n}\sum_{i=1}^n \ell_{\text{hl}}(\boldsymbol{w}, \theta; \boldsymbol{x}^{(i)}, y^{(i)})$$

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

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

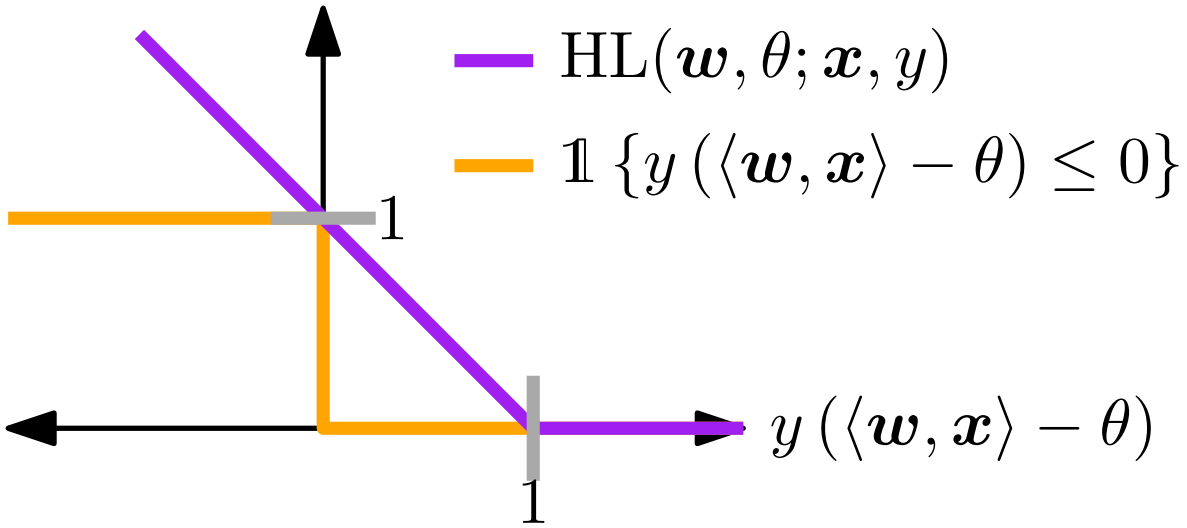
Zero-one loss vs. hinge loss



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

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

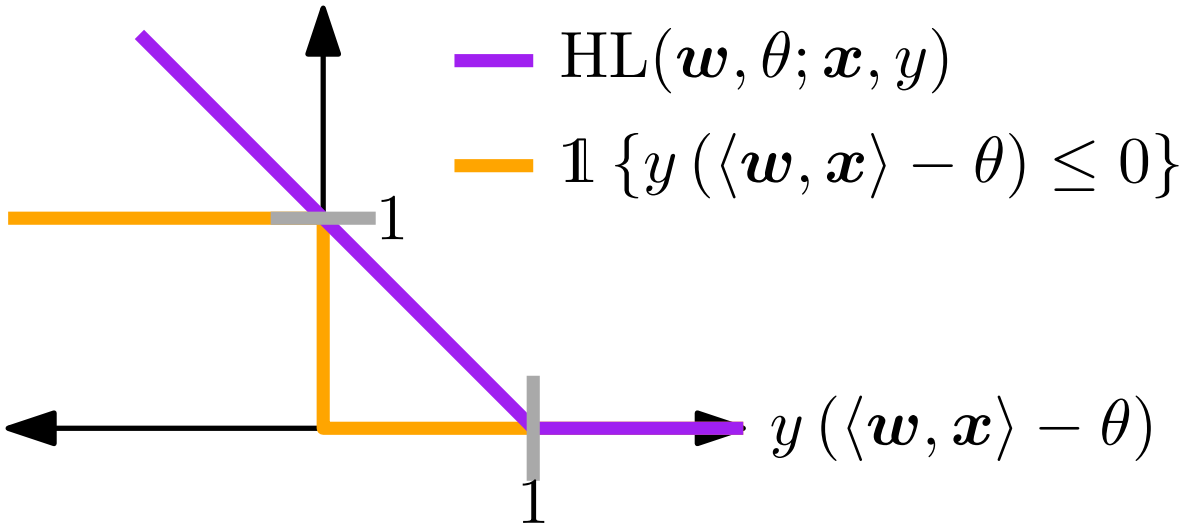
Zero-one loss vs. hinge loss



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

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

Zero-one loss vs. hinge loss

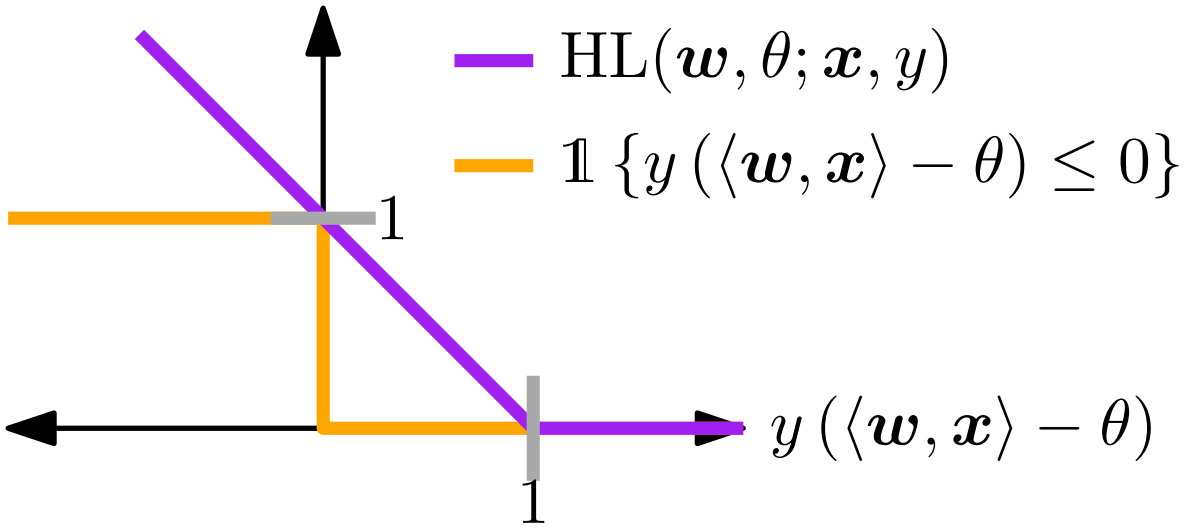


Hinge loss: an upper-bound on zero-one loss.

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

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.

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

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):

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

Soft-margin SVM:

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

General form

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

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

Soft-margin SVM:

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

Generic learning objective:

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

- **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:

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

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

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

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

Similarity to maximum likelihood

Generic learning objective:

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

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

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

(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.

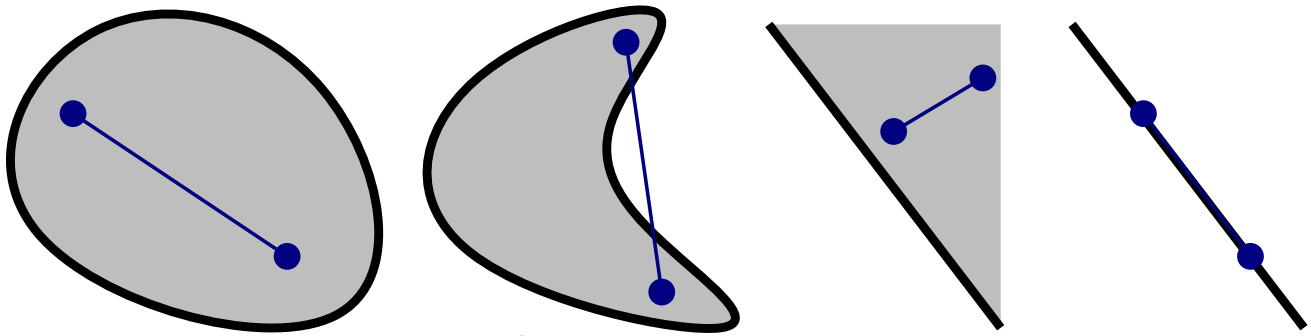
Learning via optimization

- 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
  - $R(\boldsymbol{w}) \propto \sum_{i=1}^n w_i \ln w_i$: "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

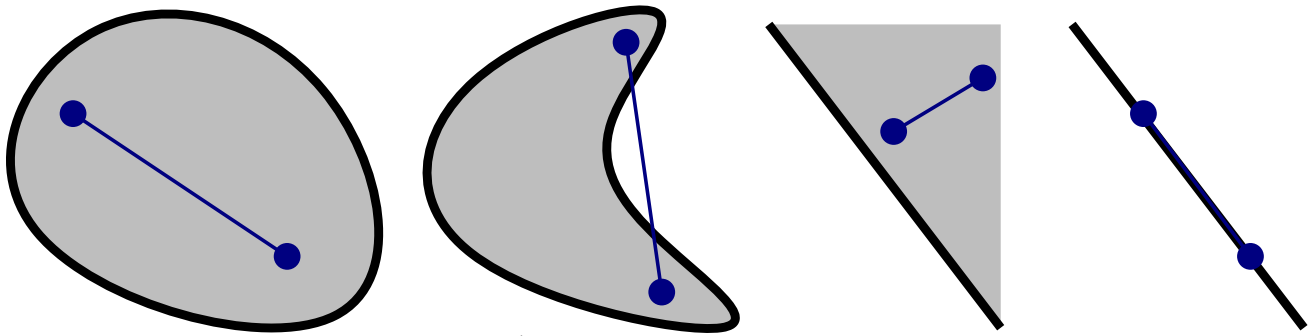
We say a set A is **convex** if, for every pair of points $\{\boldsymbol{x}, \boldsymbol{x}'\}$ in the set A, the line segment between the points $\boldsymbol{x}$ and $\boldsymbol{x}'$ is also contained in the set A.



convex not convex convex convex

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 not convex convex convex

Examples:

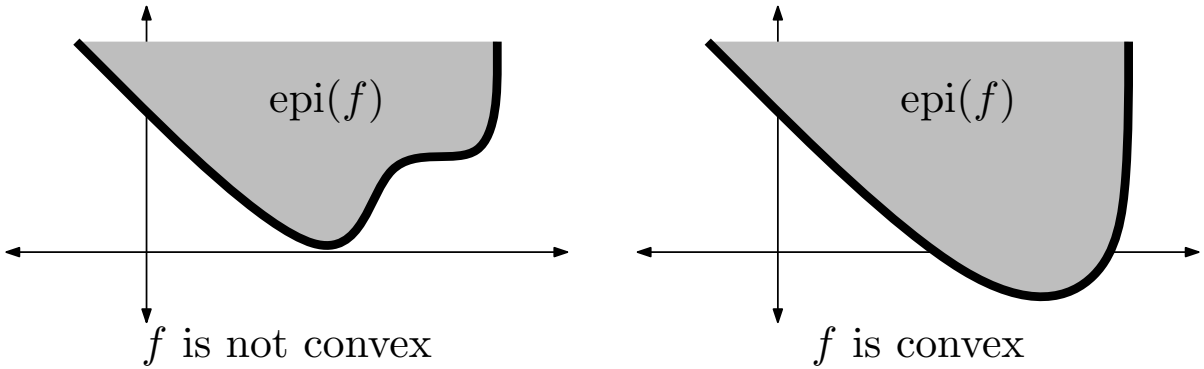
- All of $\mathbb{R}^d$.
- Empty set.
- Affine hyperplanes.
- Half-spaces: $\{\boldsymbol{a} \in \mathbb{R}^d : \langle \boldsymbol{a}, \boldsymbol{x} \rangle - b \leq 0\}$.
- Intersections of convex sets.
- Convex hulls of points.

Convex functions

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

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

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



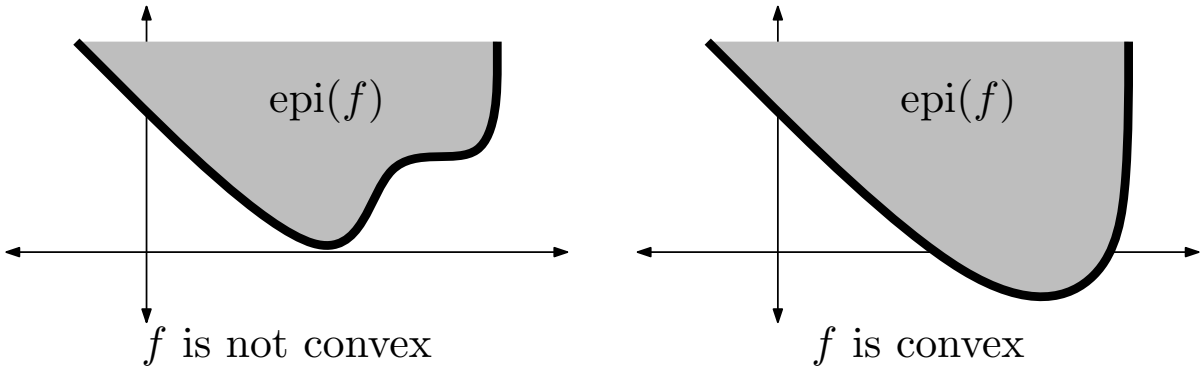
f is not convex $\qquad$ f is convex

Convex functions

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

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

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



f is not convex $\qquad$ f is convex

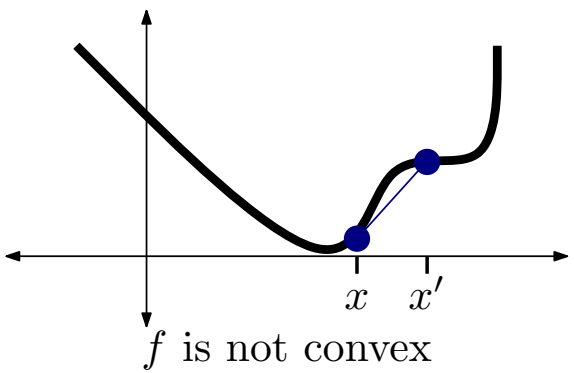
Examples:

- $f(x) = c^x$ 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 $\boldsymbol{A}$.

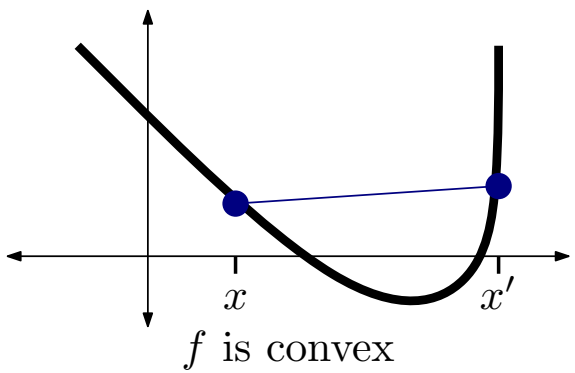
Equivalent definition of convex functions

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

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



f is not convex

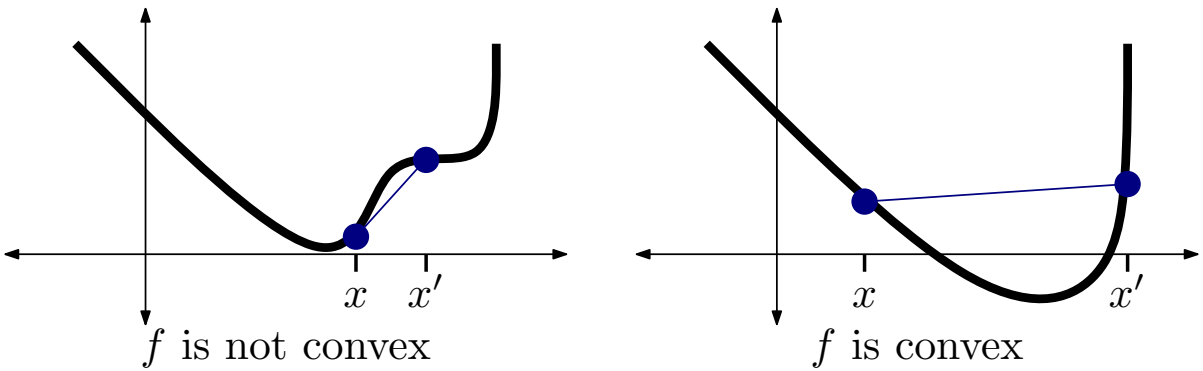


f is convex

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 $\alpha \in [0,1]$,

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



f is not convex $\qquad$ f is convex

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 $\sum_{i=1}^n \alpha_i = 1$,

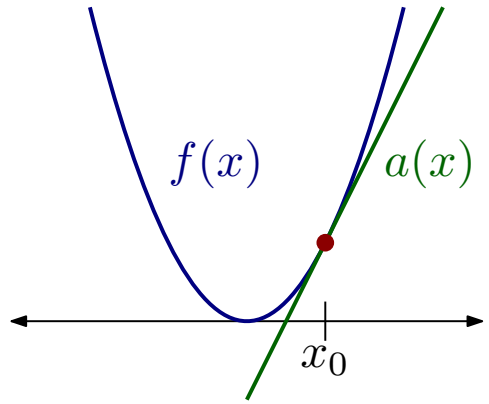
$$f\left(\sum_{i=1}^n \alpha_i \boldsymbol{x}_i\right) \leq \sum_{i=1}^n \alpha_i \cdot f(\boldsymbol{x}_i).$$

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



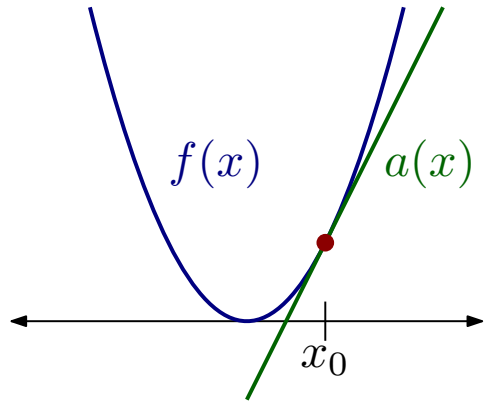
$$a(x) = f(x_0) + f'(x_0)(x - x_0)$$

Differentiable functions

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

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

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

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

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

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**.

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

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

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.

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*.

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*.

Fact: the feasible 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 a convex set.

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*.

Fact: the feasible 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 a convex set.

Fact: convex optimization problems can be solved efficiently (next lecture)

Example: Soft-Margin SVM

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

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

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

Example: Soft-Margin SVM

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

Bringing to standard form:

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