Convex optimization

Soft-margin SVMs

Soft-margin SVM optimization problem defined by training data $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$:

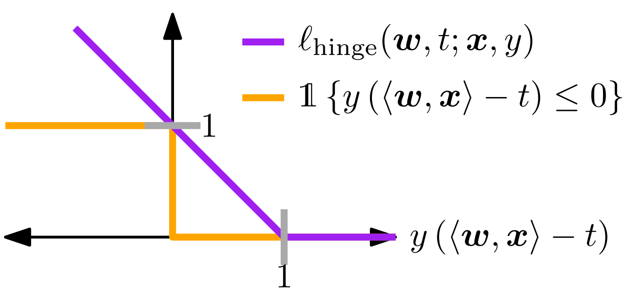
$$\min_{\boldsymbol{w}\in\mathbb{R}^d, t\in\mathbb{R}} \quad \frac{\lambda}{2}\|\boldsymbol{w}\|_2^2 + \frac{1}{n}\sum_{i=1}^n \Big[1 - y_i\big(\langle \boldsymbol{w}, \boldsymbol{x}_i\rangle - t\big)\Big]_+ .$$

Compare with **Empirical Risk Minimization** (i.e., minimize training error rate):

$$\min_{\boldsymbol{w}\in\mathbb{R}^d, t\in\mathbb{R}} \quad \frac{1}{n}\sum_{i=1}^n \mathbb{1}\Big\{y_i\big(\langle \boldsymbol{w}, \boldsymbol{x}_i\rangle - t\big) \le 0\Big\} .$$

In both cases, i-th term in summation is function of $y_i(\langle \boldsymbol{w}, \boldsymbol{x}_i\rangle - t)$.

Zero-one loss vs. hinge loss



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

$$\mathbb{1}\Big\{y\big(\langle \boldsymbol{w}, \boldsymbol{x}\rangle - t\big) \le 0\Big\} \ \le\ \Big[1 - y\big(\langle \boldsymbol{w}, \boldsymbol{x}\rangle - t\big)\Big]_+ \ =:\ \ell_{\text{hinge}}(\boldsymbol{w}, t; \boldsymbol{x}, y) .$$

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

General form

Soft-margin SVM:

$$\min_{\boldsymbol{w}\in\mathbb{R}^d, t\in\mathbb{R}} \quad \frac{\lambda}{2}\|\boldsymbol{w}\|_2^2 + \frac{1}{n}\sum_{i=1}^n \ell_{\text{hinge}}(\boldsymbol{w}, t; \boldsymbol{x}_i, y_i) .$$

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

$$\min_{\boldsymbol{w}\in\mathbb{R}^d, t\in\mathbb{R}} \quad \frac{1}{n}\sum_{i=1}^n \mathbb{1}\Big\{y_i\big(\langle \boldsymbol{w}, \boldsymbol{x}_i\rangle - t\big) \le 0\Big\} .$$

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 term**: encodes "learning bias" (e.g., prefer large margins).
- **Training loss term**: how poor is the "fit" to the training data.

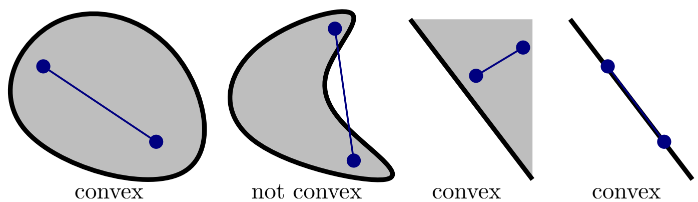
Learning via optimization

- Many different choices for regularization and loss.
  - $R(\boldsymbol{w}) = \lambda \|\boldsymbol{w}\|_2^2/2$: encourage large margins
  - $R(\boldsymbol{w}) = \lambda \|\boldsymbol{w}\|_1$: encourage $\boldsymbol{w}$ to be sparse
  - $R(\boldsymbol{w}) = \lambda \sum_{i=1}^{d-1} |w_{i+1} - w_i|$: useful for ordered features.
  - $\ell(\boldsymbol{w}; \boldsymbol{x}, y) = [\langle \boldsymbol{w}, \boldsymbol{x} \rangle - y]_+^2$
  - $\ell(\boldsymbol{w}, t; \boldsymbol{x}, y) = \ln\big(1 + \exp(-y\langle \boldsymbol{w}, \boldsymbol{x} \rangle)\big)$ (logistic regression)
  - ...
  - Also used beyond classification (e.g., regression, parameter estimation, clustering)
- For classification problems, often want ℓ to be an upper-bound on zero-one loss—i.e., a **surrogate loss**.
- Trade-off parameter $\lambda > 0$: usually determine using cross validation.
- Computationally easier when overall objective function is **convex**: possible to efficiently find global minimizer in polynomial time.

Introduction to convexity

Convex sets

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



Examples:

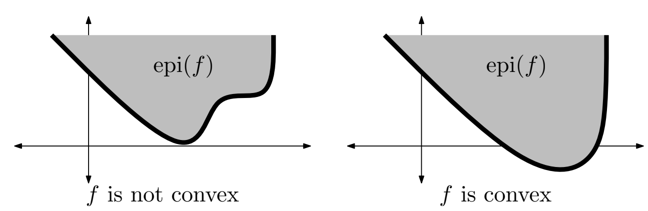
- All of $\mathbb{R}^d$.
- Empty set.
- Affine hyperplanes.
- Half-spaces: $\{\boldsymbol{a} \in \mathbb{R}^d : \langle \boldsymbol{a}, \boldsymbol{x} \rangle - b \leq 0\}$.
- Intersections of convex sets.
- Convex hulls:
  $\operatorname{conv}(S) := \{\sum_{i=1}^k \alpha_i \boldsymbol{x}_i : k \in \mathbb{N},\ \boldsymbol{x}_i \in S,\ \alpha_i \geq 0,\ \sum_{i=1}^k \alpha_i = 1\}$.

Convex functions

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

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

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



Examples:

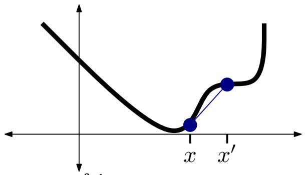
- $f(x) = c^x$ for any $c > 0$ (on $\mathbb{R}$)
- $f(x) = |x|^c$ for any $c \geq 1$ (on $\mathbb{R}$)
- $f(\boldsymbol{x}) = c$ for any constant $c \in \mathbb{R}$.
- $f(\boldsymbol{x}) = \langle \boldsymbol{a}, \boldsymbol{x} \rangle$ for any $\boldsymbol{a} \in \mathbb{R}^d$.
- $f(\boldsymbol{x})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

Jensen's inequality

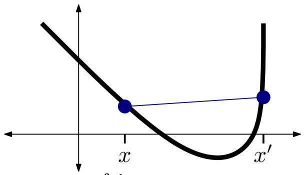
Equivalent definition of convex functions

A function $f\colon \mathbb{R}^d \to \mathbb{R}$ is convex iff 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

Jensen's inequality

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

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

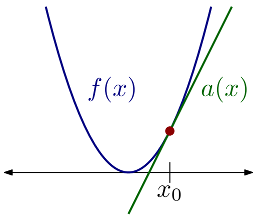
Convexity of differentiable functions

Differentiable functions

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

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

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 $\boldsymbol{x}$).

Convex optimization problems

Optimization problems

A typical optimization problem (in standard form) is written as

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

- $f_0\colon \mathbb{R}^d \to \mathbb{R}$ is the **objective function**;
- $f_1, f_2, \dots, f_n\colon \mathbb{R}^d \to \mathbb{R}$ are the **constraint functions**;
- inequalities $f_i(\boldsymbol{x}) \le 0$ are **constraints**;
- $A := \Big\{\boldsymbol{x} \in \mathbb{R}^d : f_i(\boldsymbol{x}) \le 0 \text{ for all } i = 1, 2, \dots, n\Big\}$ is the **feasible region**.
- **The goal**: Find $\boldsymbol{x} \in A$ so that $f_0(\boldsymbol{x})$ is as small as possible.
- **(Optimal) value** of the optimization problem is the smallest such value of $f_0(\boldsymbol{x})$ achieved by a feasible point $\boldsymbol{x} \in A$.
- 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**:

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

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

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

Example: Soft-margin SVM

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

Bringing to standard form:

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

Local minimizers

Consider an optimization problem (not necessarily convex):

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

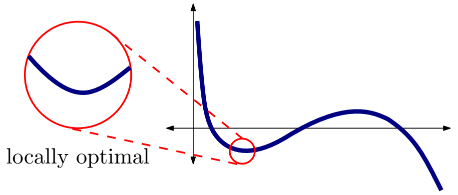
We say $\tilde{\boldsymbol{x}} \in A$ is a **local minimizer** if there is an open ball

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

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

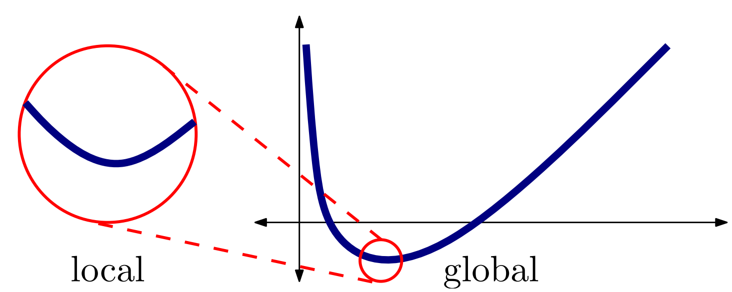
$$\begin{aligned} \min_{\boldsymbol{x}\in\mathbb{R}^d} \quad & f_0(\boldsymbol{x}) \\ \text{s.t.} \quad & \boldsymbol{x} \in A \cap U\,. \end{aligned}$$

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



Local minimizers of convex opt. problems

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



Local optimization algorithms

Unconstrained convex optimization

Unconstrained convex optimization problem

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

(f is the convex objective function; feasible region is $\mathbb{R}^d$).

Optimality condition for differentiable convex objectives

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

Unfortunately, can't always find closed-form solution to system of equations $\nabla f(\boldsymbol{x}) = \mathbf{0}$. $\longrightarrow$ Resort to iterative methods to find a solution.

Local optimization for convex objectives

Locally change $\boldsymbol{x} \to \boldsymbol{x} + \boldsymbol{\delta}$.
Hopefully improve objective value $f(\boldsymbol{x}) \to f(\boldsymbol{x} + \boldsymbol{\delta})$.

How to pick δ?

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

$$f(\boldsymbol{x} + \boldsymbol{\delta}) \geq f(\boldsymbol{x}) .$$ **Clearly a bad direction.**

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

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.

Step sizes via line search

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 $\eta := \frac{1}{2}\eta$.

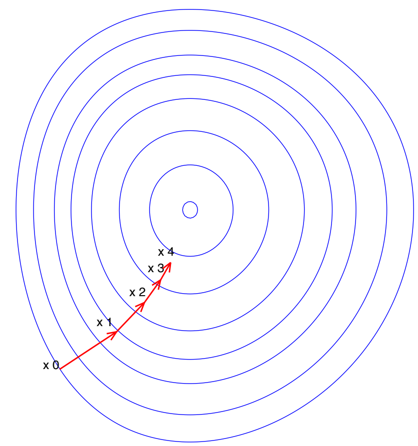
Main idea: $f(\boldsymbol{x} - \eta\boldsymbol{\lambda}) \approx f(\boldsymbol{x}) - \eta\|\boldsymbol{\lambda}\|_2^2$ when η is small, so can optimistically hope to decrease value by about $\eta\|\boldsymbol{\lambda}\|_2^2$.

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

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

Many other line search methods are possible.

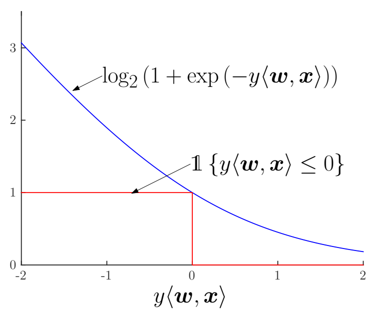
Illustration of gradient descent



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: logistic regression ($y \in \{-1, +1\}$)

$$\min_{\boldsymbol{w}\in\mathbb{R}^d} \quad f(\boldsymbol{w}) \quad \text{where} \quad f(\boldsymbol{w}) \;:=\; \frac{1}{|S|}\sum_{(\boldsymbol{x},y)\in S} \ln\Big(1 + \exp\big(-y\langle\boldsymbol{w},\boldsymbol{x}\rangle\big)\Big)\,.$$



Easy to check that f is convex.

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

Example: logistic regression ($y \in \{-1, +1\}$)

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

$$\begin{aligned}
\nabla f(\boldsymbol{w}) &= -\frac{1}{|S|}\sum_{(\boldsymbol{x},y)\in S} \frac{1}{1 + e^{y\langle\boldsymbol{w},\boldsymbol{x}\rangle}} y\boldsymbol{x} \\
&= -\frac{1}{|S|}\sum_{(\boldsymbol{x},y)\in S} \big(1 - P_{\boldsymbol{w}}(Y{=}y \mid \boldsymbol{X}{=}\boldsymbol{x})\big) y\boldsymbol{x}\,.
\end{aligned}$$

Gradient descent algorithm for logistic regression:

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

$$\begin{aligned}
\boldsymbol{w}^{(t+1)} &:= \boldsymbol{w}^{(t)} - \eta_t \nabla f(\boldsymbol{w}^{(t)}) \\
&= \boldsymbol{w}^{(t)} + \eta_t \frac{1}{|S|}\sum_{(\boldsymbol{x},y)\in S} \big(1 - P_{\boldsymbol{w}^{(t)}}(Y{=}y \mid \boldsymbol{X}{=}\boldsymbol{x})\big) y\boldsymbol{x}.
\end{aligned}$$

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 ($\|\nabla f(\boldsymbol{x})\|_2 \leq \epsilon$ 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 rate.

- Running gradient descent to convergence not strictly necessary: **may be beneficial to stop early (e.g., when hold-out error rate starts to increase significantly).**

Key takeaways

1. Formulate learning with a general loss function and regularizer as an optimization problem.
2. Convex sets, convex functions, ways to check convexity of a function.
3. Standard form of convex optimization problems, concept of local and global minimizers.
4. Gradient descent algorithm (for unconstrained problems with differentiable objectives); high-level idea of backtracking line search.
5. Stopping condition for machine learning applications.