Optimization for Machine Learning

Introduction and gradient descent

Clément Royer

CIMPA School "Control, Optimization and Model Reduction in Machine Learning"

February 24, 2025

About the instructor

Who I am: Clément Royer

- *Maître de conférences* at Dauphine since 2019.
- Research topics: Optimization and applications.
- Email: `clement.royer@lamsade.dauphine.fr`
- Webpage: https://www.lamsade.dauphine.fr/~croyer

About these lectures

Repository: M https://tinyurl.com/3etmd46y

Learning goals

- Have an optimization toolbox for ML;
- Know the theoretical underpinnings;
- Practical experience.

Outline

Outline

What's optimization?

- Operations research;
- Decision-making;
- Decision sciences;
- Mathematical programming;
- Mathematical optimization.

$\Rightarrow$ All of these can be considered as optimization.

What's optimization?

- Operations research;
- Decision-making;
- Decision sciences;
- Mathematical programming;
- Mathematical optimization.

$\Rightarrow$ All of these can be considered as optimization.

My definition

The purpose of optimization is to make the best decision out of a set of alternatives.

A warning

Optimization $\not\subset$ Machine Learning

- Optimization is a mathematical tool;
- Used in many areas: Economics, Chemistry, Physics, Social sciences,...
- Appears in other branches of (applied) mathematics: Linear Algebra, PDEs, Statistics, etc.

A warning

Optimization $\not\subset$ Machine Learning

- Optimization is a mathematical tool;
- Used in many areas: Economics, Chemistry, Physics, Social sciences,...
- Appears in other branches of (applied) mathematics: Linear Algebra, PDEs, Statistics, etc.

Machine Learning $\not\subset$ Optimization

- Optimization targets a certain problem;
- ML is not just about this problem;
- Other features of ML (data cleaning, hardware,...) will not appear in the optimization.

Formulation of an (unconstrained) optimization problem

$$\underset{\boldsymbol{w}\in\mathbb{R}^d}{\text{minimize}}\, f(\boldsymbol{w})$$

Formulation of an (unconstrained) optimization problem

$$\operatorname*{minimize}_{\boldsymbol{w} \in \mathbb{R}^d} f(\boldsymbol{w})$$

- $\boldsymbol{w}$ represents the optimization variable(s);
- d is the dimension of the problem (we will assume $d \geq 1$);
- $f(\cdot)$ is the objective/cost/loss function.

Formulation of an (unconstrained) optimization problem

$$\operatorname*{minimize}_{\boldsymbol{w} \in \mathbb{R}^d} f(\boldsymbol{w})$$

- $\boldsymbol{w}$ represents the optimization variable(s);
- d is the dimension of the problem (we will assume $d \geq 1$);
- $f(\cdot)$ is the objective/cost/loss function.

Maximizing f is equivalent to minimizing $-f$.

Example: SVM Classification

Given: A dataset $\{(\boldsymbol{x}_1, y_1), \dots, (\boldsymbol{x}_n, y_n)\}$.

- $\boldsymbol{x}_i$ is a feature vector in $\mathbb{R}^d$;
- y_i is a label.

Example: SVM Classification

Given: A dataset $\{(\boldsymbol{x}_1, y_1), \dots, (\boldsymbol{x}_n, y_n)\}$.

- $\boldsymbol{x}_i$ is a feature vector in $\mathbb{R}^d$;
- y_i is a label.

Motivation: text classification

Using d words for classification:

- $\boldsymbol{x}_i$ represents the words contained in a text document:

$$[\boldsymbol{x}_i]_j = \begin{cases} 1 & \text{if word } j \text{ is in document } i, \\ 0 & \text{otherwise.} \end{cases}$$

- y_i is equal to $+1$ if the document addresses a certain topic of interest, to -1 otherwise.

Example: SVM Classification (2)

Learning process

- Given $\{(\boldsymbol{x}_i, y_i)\}_i$, discover a function $h : \mathbb{R}^d \rightarrow \mathbb{R}$ such that $h(\boldsymbol{x}_i) \approx y_i \ \forall i = 1, \dots, n$.
- Choose the predictor function h among a set $\mathcal{H}$ parameterized by a vector $\boldsymbol{w} \in \mathbb{R}^d$: $\mathcal{H} = \left\{ h \mid h = h(\cdot; \boldsymbol{w}), \ \boldsymbol{w} \in \mathbb{R}^{\hat{d}} \right\}$;

Example: SVM Classification (2)

Learning process

- Given $\{(\boldsymbol{x}_i, y_i)\}_i$, discover a function $h : \mathbb{R}^d \to \mathbb{R}$ such that $h(\boldsymbol{x}_i) \approx y_i \ \forall i = 1, \dots, n$.
- Choose the predictor function h among a set $\mathcal{H}$ parameterized by a vector $\boldsymbol{w} \in \mathbb{R}^d$: $\mathcal{H} = \left\{ h \mid h = h(\cdot; \boldsymbol{w}),\ \boldsymbol{w} \in \mathbb{R}^{\hat{d}} \right\}$;

Linear model for text classification

- We seek a hyperplane in $\mathbb{R}^d$ separating the feature vectors associated with $y_i = +1$ and those associated with $y_i = -1$;
- This corresponds to a linear model $h(x) = \boldsymbol{x}^{\mathrm{T}} \boldsymbol{w}$, and we want to choose $\boldsymbol{w}$ such that:

$$\forall i = 1, \dots, n, \qquad \begin{cases} \boldsymbol{x}_i^{\mathrm{T}} \boldsymbol{w} \geq 1 & \text{if } y_i = +1 \\ \boldsymbol{x}_i^{\mathrm{T}} \boldsymbol{w} \leq -1 & \text{if } y_i = -1. \end{cases}$$

Example: SVM Classification (3)

An objective to optimize over

- Our goal: penalize values of $\boldsymbol{w}$ for which $h(\boldsymbol{x}_i)$ does not predict y_i well enough.
- We use the **hinge loss function**

$$\forall (h, y) \in \mathbb{R}^2, \quad \ell(h, y) = \max\{1 - yh, 0\}.$$

About the hinge loss

- $hy > 1 \Rightarrow \ell(h, y) = 0$: h and y are of the same sign, $|h| > 1$ so good prediction;
- $hy < -1 \Rightarrow \ell(h, y) > 2$: h and y are of opposite sign and $|h| > 1$ bad prediction);
- $|hy| \le 1 \Rightarrow \ell(h, y) \in [0, 2]$: small penalty (value of $|h|$ makes the prediction less certain).

Example: SVM Classification (4)

An optimization problem

$$\underset{\boldsymbol{w}\in\mathbb{R}^d}{\text{minimize}} \frac{1}{n}\sum_{i=1}^{n} \max\left\{1 - y_i(\boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{w}), 0\right\} \qquad .$$

Example: SVM Classification (4)

An optimization problem

$$\underset{\boldsymbol{w}\in\mathbb{R}^d}{\text{minimize}}\ \frac{1}{n}\sum_{i=1}^{n}\max\left\{1-y_i(\boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{w}),0\right\} \qquad .$$

- Minimize the sum of the losses for all examples;

Example: SVM Classification (4)

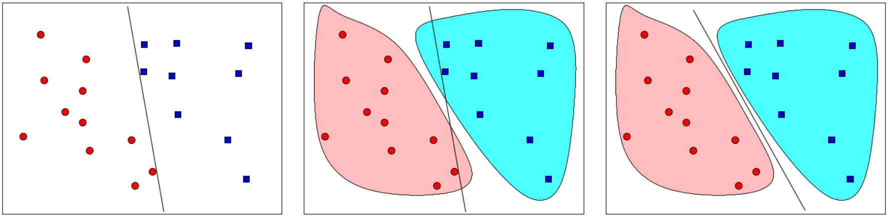
An optimization problem

$$\operatorname*{minimize}_{\boldsymbol{w}\in\mathbb{R}^d} \frac{1}{n}\sum_{i=1}^{n} \max\left\{1 - y_i(\boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{w}), 0\right\} + \frac{\lambda}{2}\|\boldsymbol{w}\|_2^2.$$

for $\lambda \geq 0$.

- Minimize the sum of the losses for all examples;
- Regularizing term to promote small-norm solutions (more on that later).

Example: SVM Classification (4)



Source: S. J. Wright & B. Recht, Optimization for Data Analysis, 2022.

- Red/Blue dots: data points labeled +1/-1;
- Red/Blue clouds: distribution of the text documents;
- Two linear classifiers;
- Rightmost plot: maximal-margin solution.

Typical optimization problem for ML

- **Data**, e.g. $\{\boldsymbol{x}_i, \boldsymbol{y}_i\}_{i=1}^n$.
- **Model class** $\mathcal{H} = \{\boldsymbol{h}(\cdot; \boldsymbol{w}), \boldsymbol{w} \in \mathbb{R}^d\}$
- **Loss function** ℓ.

Empirical risk minimization

$$\operatorname*{minimize}_{\boldsymbol{w} \in \mathbb{R}^d} \underbrace{\frac{1}{n} \sum_{i=1}^n \ell(\boldsymbol{h}(\boldsymbol{x}_i, \boldsymbol{w}), \boldsymbol{y}_i)}_{f(\boldsymbol{w})} + \lambda \Omega(\boldsymbol{w})$$

- f: Data-fitting term.
- Ω: Regularization term.

A few more examples

Linear regression

$$\operatorname*{minimize}_{\boldsymbol{w}\in\mathbb{R}^d} \frac{1}{2n}\|\boldsymbol{X}\boldsymbol{w}-\boldsymbol{y}\|_2^2 = \frac{1}{2n}\sum_{i=1}^n (\boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{w}-y_i)^2.$$

- Simplest data analysis task possible.
- $\boldsymbol{x}_i \in \mathbb{R}^d$, $y_i \in \mathbb{R}$.
- Nontrivial to solve when $n, d \gg 1$.

A few more examples

Linear regression

$$\operatorname*{minimize}_{\boldsymbol{w} \in \mathbb{R}^d} \frac{1}{2n} \|\boldsymbol{X}\boldsymbol{w} - \boldsymbol{y}\|_2^2 = \frac{1}{2n} \sum_{i=1}^{n} (\boldsymbol{x}_i^{\mathrm{T}} \boldsymbol{w} - y_i)^2.$$

- Simplest data analysis task possible.
- $\boldsymbol{x}_i \in \mathbb{R}^d$, $y_i \in \mathbb{R}$.
- Nontrivial to solve when $n, d \gg 1$.

Alternate losses for linear regression

- ℓ_1 loss: $\|\boldsymbol{X}\boldsymbol{w} - \boldsymbol{y}\|_1 = \sum_{i=1}^n |\boldsymbol{x}_i^{\mathrm{T}} \boldsymbol{w} - y_i|$
- Chebyshev loss: $\|\boldsymbol{X}\boldsymbol{w} - \boldsymbol{y}\|_\infty = \max_{1 \le i \le n} |\boldsymbol{x}_i^{\mathrm{T}} \boldsymbol{w} - y_i|$.
- And more!

A few more examples ('ed)

Binary classification (using CNNs)

$$\underset{\boldsymbol{w}\in\mathbb{R}^d}{\text{minimize}} \frac{1}{n}\sum_{i=1}^{n} \log(1+\exp(-y_i \mathrm{CNN}(\boldsymbol{x}_i)) + \lambda\|\boldsymbol{w}\|_1.$$

- Cross-entropy/Logistic loss.
- $\boldsymbol{x}_i \in \mathbb{R}^{d_0\times d_0\times c_0}$ (image), $y_i \in \{-1,1\}$ (class).
- $\mathrm{CNN} : \boldsymbol{x}_i = \boldsymbol{z}^{(0)} \mapsto \boldsymbol{z}^{(1)} \mapsto \cdots \mapsto \boldsymbol{z}^{(L)}$, where

$$\boldsymbol{z}_{ijk}^{(l)} = \phi\left(\sum_{m,n,p} \boldsymbol{W}_{m,n,p,k}^{(l-1)} \boldsymbol{z}_{i-m,j-n,p}^{(l-1)} + \boldsymbol{b}_k^{(l-1)}\right).$$

$\phi(\boldsymbol{z}) = [\max(\boldsymbol{z}_i, 0)]_i$ (ReLU activation).

- $\boldsymbol{w}$ concatenates all $(\boldsymbol{W}^l, \boldsymbol{b}^l)_{l=0\dots(L-1)}$.

Takeaways

Generic form: $\text{minimize}_{\boldsymbol{w} \in \mathbb{R}^d} f(\boldsymbol{w}) + \lambda \Omega(\boldsymbol{w})$.

Common traits

- Defined based on data.
- Use continuous functions (linear, ReLU, log/exp).

Distinctive aspects

- Model complexity/Number of parameters.
- Nonlinearity of operations.
- Regularization/Lack thereof.

Outline

Local and global solutions

$$\underset{\boldsymbol{w}\in\mathbb{R}^d}{\text{minimize}}\, f(\boldsymbol{w})$$

- $\text{argmin}_{\boldsymbol{w}\in\mathbb{R}^d} f(\boldsymbol{w})$: Set of solutions.
- $\min_{\boldsymbol{w}\in\mathbb{R}^d} f(\boldsymbol{w})$: Optimal value (can be infinite).

Local and global solutions

$$\underset{\boldsymbol{w} \in \mathbb{R}^d}{\text{minimize}}\, f(\boldsymbol{w})$$

- $\text{argmin}_{\boldsymbol{w} \in \mathbb{R}^d} f(\boldsymbol{w})$: Set of solutions.
- $\min_{\boldsymbol{w} \in \mathbb{R}^d} f(\boldsymbol{w})$: Optimal value (can be infinite).

Global and local minima

- $\boldsymbol{w}^*$ is a solution or a **global minimum** of f if $f(\boldsymbol{w}^*) \leq f(\boldsymbol{w})\ \forall \boldsymbol{w} \in \mathbb{R}^d$.
- $\boldsymbol{w}^*$ is a **local minimum** of f if $f(\boldsymbol{w}^*) \leq f(\boldsymbol{w})\ \forall \boldsymbol{w}, \|\boldsymbol{w} - \boldsymbol{w}^*\|_2 \leq \epsilon$ for some $\epsilon > 0$.

Local and global solutions

$$\underset{\boldsymbol{w} \in \mathbb{R}^d}{\text{minimize}}\, f(\boldsymbol{w})$$

- $\text{argmin}_{\boldsymbol{w} \in \mathbb{R}^d} f(\boldsymbol{w})$: Set of solutions.
- $\min_{\boldsymbol{w} \in \mathbb{R}^d} f(\boldsymbol{w})$: Optimal value (can be infinite).

Global and local minima

- $\boldsymbol{w}^*$ is a solution or a **global minimum** of f if $f(\boldsymbol{w}^*) \leq f(\boldsymbol{w})\ \forall \boldsymbol{w} \in \mathbb{R}^d$.
- $\boldsymbol{w}^*$ is a **local minimum** of f if $f(\boldsymbol{w}^*) \leq f(\boldsymbol{w})\ \forall \boldsymbol{w}, \|\boldsymbol{w} - \boldsymbol{w}^*\|_2 \leq \epsilon$ for some $\epsilon > 0$.

- Finding global/local minima is hard in general!
- Regularity of f is needed.

First notion of regularity: Smoothness

Class of $\mathcal{C}^1$ functions

$f : \mathbb{R}^d \to \mathbb{R}$ is continuously differentiable/$\mathcal{C}^1$ if

- For any $\boldsymbol{w} \in \mathbb{R}^d$, the **gradient** $\nabla f(\boldsymbol{w})$ exists.
- $\nabla f : \mathbb{R}^d \to \mathbb{R}^d$ is continuous.

First notion of regularity: Smoothness

Class of $\mathcal{C}^1$ functions

$f : \mathbb{R}^d \to \mathbb{R}$ is continuously differentiable/$\mathcal{C}^1$ if

- For any $\boldsymbol{w} \in \mathbb{R}^d$, the **gradient** $\nabla f(\boldsymbol{w})$ exists.
- $\nabla f : \mathbb{R}^d \to \mathbb{R}^d$ is continuous.

Class of $\mathcal{C}^{1,1}_L$ functions ($L > 0$)

f is $\mathcal{C}^{1,1}_L$ if it is $\mathcal{C}^1$ and ∇f is L-Lipschitz continuous, i.e.

$$\forall (\boldsymbol{v}, \boldsymbol{w}) \in (\mathbb{R}^d)^2, \qquad \|\nabla f(\boldsymbol{v}) - \nabla f(\boldsymbol{w})\| \leq L\|\boldsymbol{v} - \boldsymbol{w}\|.$$

Smoothness and optimality conditions

Problem: $\text{minimize}_{\boldsymbol{w}\in\mathbb{R}^d} f(\boldsymbol{w})$, $f\mathcal{C}^1$.

First-order necessary condition

If $\boldsymbol{w}^*$ is a local minimum of the problem, **then**

$$\|\nabla f(\boldsymbol{w}^*)\| \;=\; 0.$$

- This condition is only necessary;
- A point such that $\|\nabla f(\boldsymbol{w}^*)\| = 0$ can also be a local maximum or a saddle point.

Smoothness and optimality conditions

Problem: $\text{minimize}_{\boldsymbol{w} \in \mathbb{R}^d} f(\boldsymbol{w})$, $f \mathcal{C}^1$.

First-order necessary condition

If $\boldsymbol{w}^*$ is a local minimum of the problem, **then**

$$\|\nabla f(\boldsymbol{w}^*)\| = 0.$$

- This condition is only necessary;
- A point such that $\|\nabla f(\boldsymbol{w}^*)\| = 0$ can also be a local maximum or a saddle point.

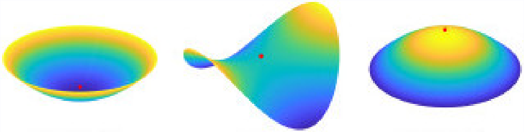
Problem: $\text{minimize}_{\boldsymbol{w} \in \mathbb{R}^d} f(\boldsymbol{w})$, $f \mathcal{C}^1$.

First-order necessary condition

If $\boldsymbol{w}^*$ is a local minimum of the problem, **then**

$$\|\nabla f(\boldsymbol{w}^*)\| = 0.$$

- This condition is only necessary;
- A point such that $\|\nabla f(\boldsymbol{w}^*)\| = 0$ can also be a local maximum or a saddle point.

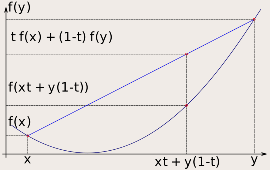


Picture from (Wright and Ma '22).

Generic definition (+Wikicommons picture)

A function $f : \mathbb{R}^d \to \mathbb{R}$ is **convex** if

$$\forall (\boldsymbol{u}, \boldsymbol{v}) \in (\mathbb{R}^d)^2, \ \forall t \in [0,1],$$
$$f(t\boldsymbol{u} + (1-t)\boldsymbol{v}) \leq t\, f(\boldsymbol{u}) + (1-t)\, f(\boldsymbol{v}).$$

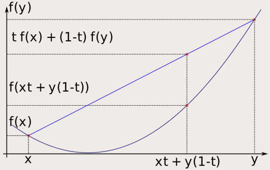


Another notion of regularity: Convexity

Generic definition (+Wikicommons picture)

A function $f : \mathbb{R}^d \to \mathbb{R}$ is **convex** if

$$\forall (\boldsymbol{u}, \boldsymbol{v}) \in (\mathbb{R}^d)^2, \ \forall t \in [0,1],$$
$$f(t\boldsymbol{u} + (1-t)\boldsymbol{v}) \le t\, f(\boldsymbol{u}) + (1-t)\, f(\boldsymbol{v}).$$



Examples in ML

- Linear function $\boldsymbol{w} \mapsto \boldsymbol{a}^{\mathrm{T}}\boldsymbol{w} + b$
- Norms $\|\boldsymbol{w}\|_2$, $\|\boldsymbol{w}\|_1$, $\|\boldsymbol{w}\|_2^2$.
- Logistic loss.

Smooth convex functions

Convexity and gradient

A continuously differentiable function $f : \mathbb{R}^d \to \mathbb{R}$ is convex if and only if

$$\forall \boldsymbol{u}, \boldsymbol{v} \in \mathbb{R}^d, \quad f(v) \geq f(u) + \nabla f(u)^{\mathrm{T}}(v - u).$$

Convexity and gradient

A continuously differentiable function $f : \mathbb{R}^d \to \mathbb{R}$ is convex if and only if

$$\forall \boldsymbol{u}, \boldsymbol{v} \in \mathbb{R}^d, \quad f(v) \geq f(u) + \nabla f(u)^{\mathrm{T}}(v - u).$$

A key inequality in optimization.

Convex optimization problem

$$\operatorname*{minimize}_{\boldsymbol{w} \in \mathbb{R}^d} f(\boldsymbol{w}), \ f \text{ convex.}$$

Convex optimization problem

$$\operatorname*{minimize}_{\boldsymbol{w}\in\mathbb{R}^d} f(\boldsymbol{w}), \ f \text{ convex.}$$

Theorem

Every local minimum of f is a global minimum.

Convex optimization problem

$$\operatorname*{minimize}_{\boldsymbol{w} \in \mathbb{R}^d} f(\boldsymbol{w}), \ f \text{ convex.}$$

Theorem

Every local minimum of f is a global minimum.

Corollary

If f is $\mathcal{C}^1$,

$$\operatorname*{argmin}_{\boldsymbol{w} \in \mathbb{R}^d} f(\boldsymbol{w}) = \left\{ \bar{\boldsymbol{w}} \mid \|\nabla f(\bar{\boldsymbol{w}})\| = 0 \right\}.$$

Any point with a zero gradient is a global minimum!

Strong convexity

Definition

A function $f : \mathbb{R}^d \to \mathbb{R}$ in $\mathcal{C}^1$ is μ-strongly convex (or *strongly convex of modulus* $\mu > 0$) if for all $(\boldsymbol{u}, \boldsymbol{v}) \in (\mathbb{R}^d)^2$ and $t \in [0, 1]$,

$$f(t\boldsymbol{u} + (1-t)\boldsymbol{v}) \leq t\, f(\boldsymbol{u}) + (1-t) f(\boldsymbol{v}) - \frac{\mu}{2} t(1-t)\|\boldsymbol{v} - \boldsymbol{u}\|^2.$$

Strong convexity

Definition

A function $f \ : \mathbb{R}^d \rightarrow \mathbb{R}$ in $\mathcal{C}^1$ is μ-strongly convex (or *strongly convex of modulus* $\mu > 0$) if for all $(\boldsymbol{u}, \boldsymbol{v}) \in (\mathbb{R}^d)^2$ and $t \in [0, 1]$,

$$f(t\boldsymbol{u} + (1-t)\boldsymbol{v}) \ \leq \ t\, f(\boldsymbol{u}) + (1-t)f(\boldsymbol{v}) - \frac{\mu}{2} t(1-t)\|\boldsymbol{v} - \boldsymbol{u}\|^2.$$

Theorem

Any strongly convex function in $\mathcal{C}^1$ has a unique global minimizer.

Gradient and strong convexity

Let $f : \mathbb{R}^d \rightarrow \mathbb{R}$, $f \in \mathcal{C}^1$. Then,

$$\forall \boldsymbol{u}, \boldsymbol{v} \in \mathbb{R}^d, \quad f(\boldsymbol{v}) \geq f(\boldsymbol{u}) + \nabla f(\boldsymbol{u})^{\mathrm{T}}(\boldsymbol{v} - \boldsymbol{u}) + \frac{\mu}{2}\|\boldsymbol{v} - \boldsymbol{u}\|^2.$$

Outline

General optimization problem

$$\operatorname*{minimize}_{\boldsymbol{w}\in\mathbb{R}^d} f(\boldsymbol{w}).$$

Assumptions: f smooth ($\mathcal{C}^1$), bounded below.

General optimization problem

$$\operatorname*{minimize}_{\boldsymbol{w}\in\mathbb{R}^d} f(\boldsymbol{w}).$$

Assumptions: f smooth ($\mathcal{C}^1$), bounded below.

Key properties

- Smoothness: We will exploit the gradient of f.
- In presence of convexity, get better guarantees.

Negative gradient direction

$$\underset{\boldsymbol{w}\in\mathbb{R}^d}{\text{minimize}}\, f(\boldsymbol{w}), \quad f \in \mathcal{C}_L^{1,1}.$$

Consider any $\boldsymbol{w} \in \mathbb{R}^d$. Then, one of the two assertions below holds:

1. Either $\boldsymbol{w}$ is a local minimum and $\nabla f(\boldsymbol{w}) = 0$;
2. Or the function f decreases locally from $\boldsymbol{w}$ in the direction of -$\nabla f(\boldsymbol{w})$.

Negative gradient direction

$$\underset{\boldsymbol{w}\in\mathbb{R}^d}{\text{minimize}}\, f(\boldsymbol{w}), \quad f \in \mathcal{C}^{1,1}_L.$$

Consider any $\boldsymbol{w} \in \mathbb{R}^d$. Then, one of the two assertions below holds:

1. Either $\boldsymbol{w}$ is a local minimum and $\nabla f(\boldsymbol{w}) = 0$;
2. Or the function f decreases locally from $\boldsymbol{w}$ in the direction of -$\nabla f(\boldsymbol{w})$.

Key argument (Taylor expansion)

$$f(\boldsymbol{v}) \approx f(\boldsymbol{w}) + \nabla f(\boldsymbol{w})^{\mathrm{T}}(\boldsymbol{v} - \boldsymbol{w}) \quad \text{for } \boldsymbol{v} \text{ close to } \boldsymbol{w}.$$

Gradient descent method

Inputs: $\boldsymbol{w}_0 \in \mathbb{R}^d, \ \alpha_0 > 0, \ k = 0.$

1. Evaluate $\nabla f(\boldsymbol{w}_k)$.
2. Set $\boldsymbol{w}_{k+1} = \boldsymbol{w}_k - \alpha_k \nabla f(\boldsymbol{w}_k)$.
3. Increment k by 1 and go to Step 1.

Gradient descent method

Inputs: $\boldsymbol{w}_0 \in \mathbb{R}^d$, $\alpha_0 > 0$, $k = 0$.

1. Evaluate $\nabla f(\boldsymbol{w}_k)$.
2. Set $\boldsymbol{w}_{k+1} = \boldsymbol{w}_k - \alpha_k \nabla f(\boldsymbol{w}_k)$.
3. Increment k by 1 and go to Step 1.

Stopping criterion

- Convergence criterion (optional): Stop when $\|\nabla f(\boldsymbol{w}_k)\| < \varepsilon$;
- Budget criterion (optional): Stop when $k = k_{\max}$.

Choosing the stepsize

Constant stepsize

If $f \in \mathcal{C}_L^{1,1}$, set $\alpha_k = \frac{1}{L}$:

- Guaranteed decrease at every iteration;
- But requires knowledge of L.

Decreasing stepsize

Choose α_k such that $\alpha_k \to 0$.

- Guarantees that f will decrease eventually (for small stepsizes);
- But steps get smaller and smaller.

Choosing the stepsize (2)

What's done in optimization

- Line search: At every iteration, α_k is obtained by *backtracking* on a subset of values (ex: $1, \frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \dots,$).
- The chosen value must satisfy certain conditions (ex: decreasing the function value).

What's done in optimization for ML

- Start with a fixed value until the method starts stalling (gradient gets small);
- Decrease the step size, then repeat.

Analyzing gradient descent

$$\underset{\boldsymbol{x}\in\mathbb{R}^d}{\text{minimize}}\, f(\boldsymbol{x}), \qquad f \in \mathcal{C}^{1,1}_L.$$

Gradient descent

- Iteration: $\boldsymbol{w}_{k+1} = \boldsymbol{w}_k - \alpha_k \nabla f(\boldsymbol{w}_k)$, stop if $\nabla f(\boldsymbol{w}_k) = 0$.
- Typical choice in theory : $\alpha_k = \frac{1}{L}$.

Theoretical analysis

- Convergence: Show that $\|\nabla f(\boldsymbol{w}_k)\| \to 0$;
- Convergence rate: Look at how fast $\|\nabla f(\boldsymbol{w}_k)\|$ decreases.
- Worst-case complexity: Equivalent to convergence rate, measures the cost of satisfying $\|\nabla f(\boldsymbol{w}_k)\| \le \epsilon$ for $\epsilon > 0$.

Convergence rates: Nonconvex case

Theorem

If $f \in \mathcal{C}^{1,1}_L$ and $\alpha_k = \frac{1}{L}$,

$$\min_{0 \le k \le K-1} \|\nabla f(\boldsymbol{w}_k)\| \le \mathcal{O}\left(\frac{1}{\sqrt{K}}\right)$$

after $K \ge 1$ iterations.

Convergence rates: Nonconvex case

Theorem

If $f \in \mathcal{C}^{1,1}_L$ and $\alpha_k = \frac{1}{L}$,

$$\min_{0 \le k \le K-1} \|\nabla f(\boldsymbol{w}_k)\| \le \mathcal{O}\left(\frac{1}{\sqrt{K}}\right)$$

after $K \ge 1$ iterations.

A key inequality for the proof

$$\forall (\boldsymbol{v}, \boldsymbol{w}), \quad f(\boldsymbol{v}) \le f(\boldsymbol{w}) + \nabla f(\boldsymbol{w})^{\mathrm{T}}(\boldsymbol{v} - \boldsymbol{w}) + \frac{L}{2}\|\boldsymbol{v} - \boldsymbol{w}\|_2^2.$$

- Another key inequality in optimization.
- With $\boldsymbol{v} = \boldsymbol{w}_{k+1}$ and $\boldsymbol{w} = \boldsymbol{w}_k$, gives decrease in $\mathcal{O}(\|\nabla f(\boldsymbol{w}_k)\|^2)$.

Theorem

Let $f \in \mathcal{C}^{1,1}_L$ be convex and $\alpha_k = \frac{1}{L}$ in GD. Then, for $K \geq 1$,

1. If f is convex,

$$f(\boldsymbol{w}_K) - f^* \leq \mathcal{O}\left(\frac{1}{K}\right).$$

2. If f is μ-strongly convex,

$$f(\boldsymbol{w}_K) - f^* \leq \mathcal{O}\left(\left(1 - \frac{\mu}{L}\right)^K\right).$$

Convergence rates (convex case)

Theorem

Let $f \in \mathcal{C}_L^{1,1}$ be convex and $\alpha_k = \frac{1}{L}$ in GD. Then, for $K \geq 1$,

1. If f is convex,

$$f(\boldsymbol{w}_K) - f^* \leq \mathcal{O}\left(\frac{1}{K}\right).$$

2. If f is μ-strongly convex,

$$f(\boldsymbol{w}_K) - f^* \leq \mathcal{O}\left(\left(1 - \frac{\mu}{L}\right)^K\right).$$

Interpretation

Nonconvex	Convex	Strongly convex
$\mathcal{O}(1/\sqrt{K})$	$\mathcal{O}(1/K)$	$\mathcal{O}(t^K)$

Stronger guarantees for convex problems at lower cost.

Conclusion: Gradient descent

A versatile algorithm

- Applies as long as f has a gradient.
- Various implementations (stepsizes).
- Theoretical guarantees for convex/nonconvex problems.

Going further

- What if the function does not have a gradient?
- What about the problem structure?

Outline

Back to where we started

The linear SVM problem

$$\min_{\mathbf{w}\in\mathbb{R}^d} \frac{1}{n}\sum_{i=1}^{n} \max\{1 - y_i \mathbf{x}_i^{\mathrm{T}} \mathbf{w}, 0\} + \frac{\lambda}{2}\|\mathbf{w}\|_2^2.$$

- The hinge loss is not continuously differentiable!
- But it is continuous and convex...

Nonsmooth functions

Definition

A function is called **nonsmooth** if it is not differentiable everywhere.
NB: Nonsmooth $\neq$ Discontinuous.

Example of nonsmooth functions

- $w \mapsto |w|$ from $\mathbb{R}$ to $\mathbb{R}$;
- $w \mapsto \|w\|_1$ from $\mathbb{R}^d$ to $\mathbb{R}$;
- ReLU: $w \mapsto \max\{w, 0\}$ from $\mathbb{R}^d$ to $\mathbb{R}$.

Subgradients for nonsmooth convex problems

Definition

Let $f : \mathbb{R}^n \rightarrow \mathbb{R}$ be a convex function. A vector $\boldsymbol{g} \in \mathbb{R}^n$ is called a *subgradient* of f at $\boldsymbol{w} \in \mathbb{R}^n$ if

$$\forall \boldsymbol{z} \in \mathbb{R}^n, \qquad f(\boldsymbol{z}) \ \geq \ f(\boldsymbol{w}) + \boldsymbol{g}^{\mathrm{T}}(\boldsymbol{z} - \boldsymbol{w}).$$

The set of all subgradients of f at $\boldsymbol{w}$ is called the *subdifferential* of f at $\boldsymbol{w}$, and denoted by $\partial f(\boldsymbol{w})$.

- If f differentiable at $\boldsymbol{w}$, $\partial f(\boldsymbol{w}) = \{\nabla f(\boldsymbol{w})\}$;
- $0 \in \partial f(\boldsymbol{w}) \Leftrightarrow \boldsymbol{w}$ minimum of f!

Example: Let $f : \mathbb{R} \rightarrow \mathbb{R}$, $f(w) = |w|$.

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

Subgradient method

Iteration for nonsmooth convex f

$$\boldsymbol{w}_{k+1} = \boldsymbol{w}_k - \alpha_k \boldsymbol{g}_k, \qquad \boldsymbol{g}_k \in \partial f(\boldsymbol{w}_k).$$

- Depends on the subgradient: a subgradient can be a direction of increase!
- α_k typically constant or decreasing.

Subgradient method

Iteration for nonsmooth convex f

$$\mathbf{w}_{k+1} = \mathbf{w}_k - \alpha_k \mathbf{g}_k, \qquad \mathbf{g}_k \in \partial f(\mathbf{w}_k).$$

- Depends on the subgradient: a subgradient can be a direction of increase!
- α_k typically constant or decreasing.

Guarantees

Let $\bar{\mathbf{w}}_K = \frac{1}{\sum_{k=0}^{K-1}} \sum_{k=0}^{K-1} \alpha_k \mathbf{w}_k$. Then,

$$f(\bar{\mathbf{w}}_K) - f^* \leq \mathcal{O}\left(\frac{1}{\sqrt{K}}\right).$$

Worst rate than gradient descent but a lot more general!

Outline

Back to where we started

The linear SVM problem

$$\min_{\boldsymbol{w}\in\mathbb{R}^d} \frac{1}{n}\sum_{i=1}^{n} \max\{1 - y_i \boldsymbol{x}_i^{\mathrm{T}} \boldsymbol{w}, 0\} + \frac{\lambda}{2}\|\boldsymbol{w}\|_2^2.$$

- The problem is **regularized** (by a data-independent term);
- The purpose of regularization is to enforce specific properties/structure on a solution.

General form of a regularized problem

$$\min_{\boldsymbol{w} \in R^d} \underbrace{f(\boldsymbol{w})}_{loss\ \ function} + \underbrace{\lambda \Omega(\boldsymbol{w})}_{regularization\ \ term}.$$

where $\lambda > 0$ is called a regularization parameter.

Example: Ridge regularization

$$\min_{\boldsymbol{w} \in \mathbb{R}^d} f(\boldsymbol{w}) + \frac{\lambda}{2}\|\boldsymbol{w}\|_2^2.$$

Interpretations:

- Equivalent to enforcing a constraint on $\|\boldsymbol{w}\|_2^2 = \sum_{i=1}^d w_i^2$;
- Penalizes $\boldsymbol{w}$s with large components;
- The variance of the solution w. r. t. the data is reduced;
- The objective function is strongly convex.

Solving regularized problems

Setup: Composite optimization

$$\operatorname*{minimize}_{\boldsymbol{w}\in\mathbb{R}^d} f(\boldsymbol{w}) + \lambda\Omega(\boldsymbol{w}).$$

- $f \in \mathcal{C}^{1,1}$;
- Ω convex but nonsmooth.

Proximal approach

- Classical optimization paradigm: replace a problem by a sequence of easier (sub)problems;
- Exploit smoothness of f, use the structure of Ω to solve the subproblems;
- Those should be solvable **efficiently**.

Proximal Gradient Descent (PGD)

Iteration of PGD

$$\boldsymbol{w}_{k+1} = \operatorname*{argmin}_{\boldsymbol{w} \in \mathbb{R}^d} \left\{ f(\boldsymbol{w}_k) + \nabla f(\boldsymbol{w}_k)^{\mathrm{T}}(\boldsymbol{w} - \boldsymbol{w}_k) + \frac{1}{2\alpha_k}\|\boldsymbol{w} - \boldsymbol{w}_k\|_2^2 + \lambda\Omega(\boldsymbol{w}) \right\}.$$

- If $\Omega \equiv 0$, the solution is $\boldsymbol{w}_{k+1} = \boldsymbol{w}_k - \alpha_k \nabla f(\boldsymbol{w}_k)$: **This is the Gradient Descent iteration!**
- In general, the cost of an iteration is 1 gradient call + **1 proximal subproblem solve**.

Properties

- Complexity bounds exist for nonconvex and mostly for convex f;
- Stepsize choices can be designed based on those for GD;

Illustration: ISTA (2)

Sparsity-inducing regularizes

- Want solution $\boldsymbol{w} \in \mathbb{R}^d$ with few nonzero components.
- For linear models, amounts to feature selection.
- In general, can help with model compression.

A better approach: LASSO regularization

LASSO=Least Absolute Shrinkage and Selection Operator

$$\operatorname*{minimize}_{\boldsymbol{w} \in \mathbb{R}^d} f(\boldsymbol{w}) + \lambda \|\boldsymbol{w}\|_1, \quad \|\boldsymbol{w}\|_1 = \sum_{i=1}^{d} |w_i|.$$

- $\|\cdot\|_1$ is convex, continuous, and a norm;
- It is nonsmooth but subgradients can be computed.

Illustration: ISTA

Context

- Solve $\mathrm{minimize}_{\boldsymbol{w}\in\mathbb{R}^d} f(\boldsymbol{w}) + \lambda\|\boldsymbol{w}\|_1$;
- Common problem in image processing;
- Explicit form of the proximal subproblem solution.

Illustration: ISTA

Context

- Solve $\mathrm{minimize}_{\boldsymbol{w} \in \mathbb{R}^d} f(\boldsymbol{w}) + \lambda \|\boldsymbol{w}\|_1$;
- Common problem in image processing;
- Explicit form of the proximal subproblem solution.

Iteration of ISTA: Iterative Soft-Thresholding Algorithm

Define $\boldsymbol{w}_{k+1}$ componentwise: for any $i \in \{1, \dots, d\}$,

$$[\boldsymbol{w}_{k+1}]_i = \begin{cases} [\boldsymbol{w}_k - \alpha_k \nabla f(\boldsymbol{w}_k)]_i + \alpha_k \lambda & \text{if } [\boldsymbol{w}_k - \alpha_k \nabla f(\boldsymbol{w}_k)]_i < -\alpha_k \lambda \\ [\boldsymbol{w}_k - \alpha_k \nabla f(\boldsymbol{w}_k)]_i - \alpha_k \lambda & \text{if } [\boldsymbol{w}_k - \alpha_k \nabla f(\boldsymbol{w}_k)]_i > \alpha_k \lambda \\ 0 & \text{if } [\boldsymbol{w}_k - \alpha_k \nabla f(\boldsymbol{w}_k)]_i \in [-\alpha_k \lambda, \alpha_k \lambda]. \end{cases}$$

Summary

Optimization problems in ML

- Common feature: Depend on data.
- Distinctive features: Convexity, smoothness, regularization.

Gradient descent

- The basic block for optimization.
- Applies to convex and nonconvex functions.
- Some freedom in the implementation (see lab session).

Beyond gradient descent

- Nonsmoothness$\Rightarrow$ Subgradient methods!
- Regularization$\Rightarrow$ Proximal methods!
- Data dependency? $\Rightarrow$ See next lecture.

References

Textbooks:

- A. Beck, *First-order methods in optimization*, MOS-SIAM Series on Optimization, 2017.
  $\Rightarrow$ *Chapter 10* is related to proximal methods, and contains many examples of explicit proximal step calculations.
- J. Wright and Y. Ma, *High-Dimensional Data Analysis with Low-Dimensional Models: Principles, Com- putation, and Applications*, Cambridge University Press, 2022.
  $\Rightarrow$ Numerous applications, freely available online.
- S. J. Wright and B. Recht, *Optimization for Data Analysis*, Cambridge University Press, 2022.
  $\Rightarrow$ Textbook with full analysis for gradient descent.