Optimization for Machine Learning

Optimisation pour l'apprentissage automatique

Clément Royer

Université Paris-Dauphine

Master 2 IASD/ID Apprentissage

French touch

A warning

- This course will be given in English.
- The slides will be in English.
- The instructor is...French.

French touch

A warning

- This course will be given in English.
- The slides will be in English.
- The instructor is...French.

Why?

- It's in the syllabus!
- Latest advances in Machine Learning/Optimization are international;
- Both academic and industrial research are produced in English.

French touch

A warning

- This course will be given in English.
- The slides will be in English.
- The instructor is...French.

Why?

- It's in the syllabus!
- Latest advances in Machine Learning/Optimization are international;
- Both academic and industrial research are produced in English.

Aims of the course

- Present the main optimization tools used in ML;
- Motivate the use of these methods;
- Illustrate on typical ML problems.

Regarding this course

Lecturer : Clément Royer

- *Maître de conférences* at Dauphine since September 2019;
- From 2016 to 2019 : University of Wisconsin-Madison (USA);
- Research : Continuous optimization.

Useful information

- `clement.royer@dauphine.psl.eu`.
- Link to these slides (updated as we go).

URL:
`https://www.lamsade.dauphine.fr/~croyer/docs/courseOptiML.pdf`

Schedule

Three-hour slots

- Week 1: 09/25 (8.30am-11.45am), 09/27 (8.30am-11.45am);
- Week 2: 10/02 (1.45pm-5pm), 10/04 (8.30am-11.45am);
- Week 3: 10/07 (1.45pm-5pm), 10/10 (1.45pm-5pm);
- Week 4: 11/06 (1.45pm-5pm), 11/08 (1.45pm-5pm);
- Week 5: **Exam on 11/15**.

Lab sessions

- On 10/10 and 11/08 for ID (instructor: Clément Royer);
- On 10/10 and 11/07 for IASD (instructor: Laurent Meunier).

Schedule

Three-hour slots

- Week 1: 09/25 (8.30am-11.45am), 09/27 (8.30am-11.45am);
- Week 2: 10/02 (1.45pm-5pm), 10/04 (8.30am-11.45am);
- Week 3: 10/07 (1.45pm-5pm), 10/10 (1.45pm-5pm);
- Week 4: 11/06 (1.45pm-5pm), 11/08 (1.45pm-5pm);
- Week 5: **Exam on 11/15**.

Lab sessions

- On 10/10 and 11/08 for ID (instructor: Clément Royer);
- On 10/10 and 11/07 for IASD (instructor: Laurent Meunier).

I expect to...

- Start/finish on time;
- Be able to hear everyone;
- **Get feedback from you**.

Content of the course

1. Introduction
2. Basics of optimization
3. Unconstrained optimization
4. Constrained optimization
5. Stochastic optimization
6. Nonsmooth optimization
7. Advanced topics

Content of the course

1. Introduction (Some examples)
2. Basics of optimization (Gradients and convexity)
3. Unconstrained optimization (Gradient descent)
4. Constrained optimization (ADMM)
5. Stochastic optimization (Stochastic gradient)
6. Nonsmooth optimization (Proximal methods)
7. Advanced topics (Second-order methods?)

Outline

Terminology

What you may have heard of/read about

- Data Analysis;
- Data Mining;
- Machine Learning (ML);
- Artificial Intelligence;
- Big Data;
- ...

Terminology

What you may have heard of/read about

- Data Analysis;
- Data Mining;
- Machine Learning (ML);
- Artificial Intelligence;
- Big Data;
- ...

What this course is about

- Optimization for ML...
- ...and for all of data science.
- We will focus on **generic principles**.

ML for us

Main goals

- Extract meaning/information from data:
  Statistics, main *features* and *structures*;
- Use this information to predict behavior of yet unseen data.

ML for us

Main goals

- Extract meaning/information from data:
  Statistics, main *features* and *structures*;
- Use this information to predict behavior of yet unseen data.

Components of ML

- Statistics;
- Computer Science (data management, parallel computing, etc);
- **Optimization** for modeling and algorithms.

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.

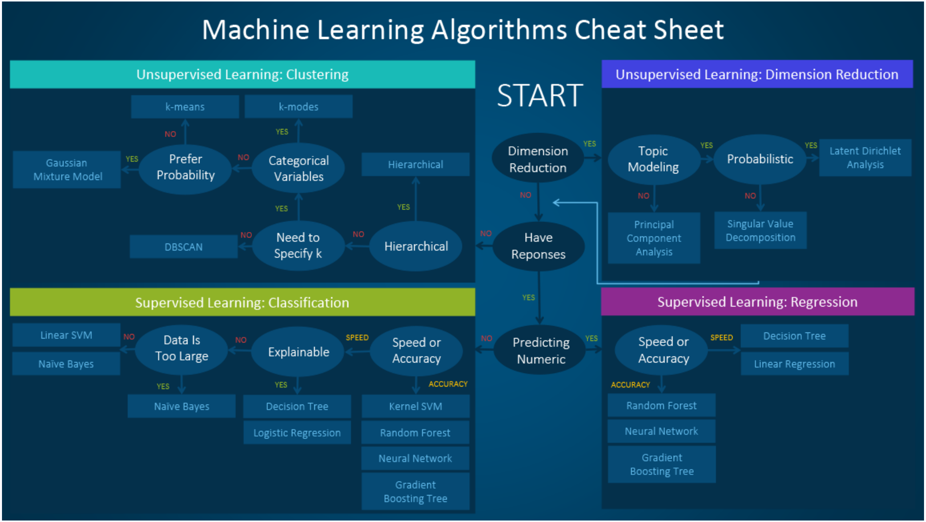
A warning

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.



Source: https://blogs.sas.com/content/subconsciousmusings/2017/04/12/ machine-learning-algorithm-use/

A view

At first there was optimization...

- The rise of optimization: 1970-1980;
- Many algorithms have proven effective in various fields;
- *Standard practice for a physics-motivated problem:* run an interior-point Newton-type method (developed in the 2000s).

...then came ML!

From an optimization point of view:

- ML problems have challenging characteristics;
- The usual solvers are not so efficient in ML problems;
- But other (old) methods have regained interest.

Ubiquitous practice in ML: Run Stochastic Gradient with Momentum (1950s + a 1983 theoretical paper).

What changed?

Big data setting

- Very expensive to compute full derivatives/look at the entire data set;
- First-order methods have proven very effective to reach low accuracies.

What changed?

Big data setting

- Very expensive to compute full derivatives/look at the entire data set;
- First-order methods have proven very effective to reach low accuracies.

Community has changed

- The optimization problem is not everything;
- Interest in statistical properties of the solutions;
- Different analyzes and theoretical results.

Outline

Statistical machine learning approach

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.

Statistical machine learning approach

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

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

Example: text classification

Using d words for classification:

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

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

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

Prediction and classification

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 $\boldsymbol{w} \in \mathbb{R}^d$: $\mathcal{H} = \left\{ h \mid h = h(\cdot; \boldsymbol{w}), \; \boldsymbol{w} \in \mathbb{R}^{\hat{d}} \right\}$;

Prediction and classification

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 $h(x) = \boldsymbol{x}^{\mathrm{T}} \boldsymbol{u} - v$, and we want to choose $\boldsymbol{w}_1, \boldsymbol{w}_0$ such that:

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

Objective of the problem

An objective to optimize over

- Our goal is to penalize values of $\boldsymbol{w} = (\boldsymbol{u}, v)$ for which $h(\boldsymbol{x}_i) \neq y_i$.
- One possibility: the **hinge loss function**

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

About the hinge loss

- $hy > 1 \Rightarrow \ell(h, y) = 0$: no penalty ($h$ and y are of the same sign, $|h| > 1$ so this is a good prediction);
- $hy < -1 \Rightarrow \ell(h, y) > 2$: large penalty ($h$ and y are of opposite sign and $|h| > 1$, this is a bad prediction);
- $|hy| \leq 1 \Rightarrow \ell(h, y) \in [0, 2]$: small penalty ($h$ and y can be of the same sign, but the value of $|h|$ makes the prediction less certain).

Optimization formulation

An optimization problem

$$\min_{\boldsymbol{u},v} \frac{1}{n}\sum_{i=1}^{n} \max\left\{1 - y_i(\boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{u} - v), 0\right\} \qquad .$$

Optimization formulation

An optimization problem

$$\min_{\boldsymbol{u},v} \frac{1}{n} \sum_{i=1}^{n} \max\left\{1 - y_i(\boldsymbol{x}_i^{\mathrm{T}} \boldsymbol{u} - v), 0\right\} \qquad .$$

- Minimize the sum of the losses for all examples;

Optimization formulation

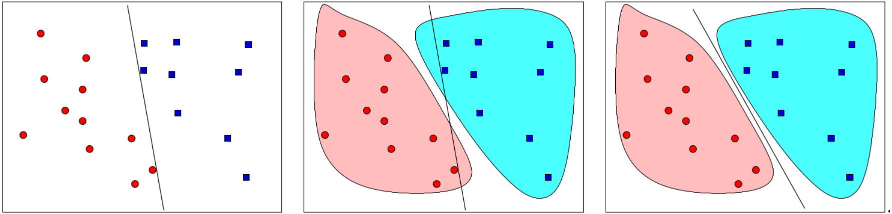
An optimization problem

$$\min_{\boldsymbol{u},v} \frac{1}{n}\sum_{i=1}^{n} \max\left\{1 - y_i(\boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{u} - v), 0\right\} + \frac{\lambda}{2}\|\boldsymbol{u}\|_2^2.$$

for $\lambda \geq 0$.

- Minimize the sum of the losses for all examples;
- A regularizing term is usually added (more on that later).

Different solutions



Source: B. Recht and S. J. Wright, Nonlinear Optimization for Machine Learning (forthcoming).

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

From an optimizer's point of view

$$\min_{\boldsymbol{u},v} \frac{1}{n} \sum_{i=1}^{n} \max\left\{1 - y_i(\boldsymbol{x}_i^{\mathrm{T}} \boldsymbol{u} - v), 0\right\} + \frac{\lambda}{2}\|\boldsymbol{u}\|_2^2$$

Reformulation

- Add variables to replace the max $\Rightarrow$ Convex quadratic program;
- Use duality $\Rightarrow$ Convex quadratic program:

$$\min_{\boldsymbol{\alpha} \in \mathbb{R}^n} \frac{1}{2}\boldsymbol{\alpha}^{\mathrm{T}} \boldsymbol{Q} \boldsymbol{\alpha} - \mathbf{1}^{\mathrm{T}} \boldsymbol{\alpha} \quad \text{subject to} \quad 0 \le \boldsymbol{\alpha} \le \frac{1}{\lambda}\mathbf{1}, \ \boldsymbol{y}^{\mathrm{T}} \boldsymbol{\alpha} = 0$$

with $\boldsymbol{Q} = [y_i\, y_j\, h(x_i)\, h(x_j)]_{ij}$, $\boldsymbol{y} = [y_1 \dots y_n]^{\mathrm{T}}$, $\mathbf{1} = [1 \cdots 1]^{\mathrm{T}} \in \mathbb{R}^n$.

From an optimizer's point of view

$$\min_{\boldsymbol{u},v} \frac{1}{n} \sum_{i=1}^{n} \max\left\{1 - y_i(\boldsymbol{x}_i^{\mathrm{T}} \boldsymbol{u} - v), 0\right\} + \frac{\lambda}{2} \|\boldsymbol{u}\|_2^2$$

Reformulation

- Add variables to replace the max $\Rightarrow$ Convex quadratic program;
- Use duality $\Rightarrow$ Convex quadratic program:

$$\min_{\boldsymbol{\alpha} \in \mathbb{R}^n} \frac{1}{2} \boldsymbol{\alpha}^{\mathrm{T}} \boldsymbol{Q} \boldsymbol{\alpha} - \mathbf{1}^{\mathrm{T}} \boldsymbol{\alpha} \quad \text{subject to} \quad 0 \le \boldsymbol{\alpha} \le \frac{1}{\lambda} \mathbf{1}, \ \boldsymbol{y}^{\mathrm{T}} \boldsymbol{\alpha} = 0$$

with $\boldsymbol{Q} = [y_i \, y_j \, h(x_i) \, h(x_j)]_{ij}$, $\boldsymbol{y} = [y_1 \dots y_n]^{\mathrm{T}}$, $\mathbf{1} = [1 \cdots 1]^{\mathrm{T}} \in \mathbb{R}^n$.

$$\min_{\boldsymbol{u},v} \frac{1}{n}\sum_{i=1}^{n} \max\left\{1 - y_i(\boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{u} - v), 0\right\} + \frac{\lambda}{2}\|\boldsymbol{u}\|_2^2$$

Reformulation

- Add variables to replace the max $\Rightarrow$ Convex quadratic program;
- Use duality $\Rightarrow$ Convex quadratic program:

$$\min_{\boldsymbol{\alpha}\in\mathbb{R}^n} \frac{1}{2}\boldsymbol{\alpha}^{\mathrm{T}}\boldsymbol{Q}\boldsymbol{\alpha} - \mathbf{1}^{\mathrm{T}}\boldsymbol{\alpha} \quad \text{subject to} \quad 0 \le \boldsymbol{\alpha} \le \frac{1}{\lambda}\mathbf{1},\ \boldsymbol{y}^{\mathrm{T}}\boldsymbol{\alpha} = 0$$

with $\boldsymbol{Q} = [y_i\, y_j\, h(x_i)\, h(x_j)]_{ij},\ \boldsymbol{y} = [y_1 \dots y_n]^{\mathrm{T}},\ \mathbf{1} = [1 \cdots 1]^{\mathrm{T}} \in \mathbb{R}^n.$

Optimizers know how to solve this efficiently.

From a data scientist's point of view

$$\min_{\boldsymbol{u},v} \underbrace{\frac{1}{n}\sum_{i=1}^{n} \max\left\{1 - y_i(\boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{u} - v), 0\right\}}_{loss} + \underbrace{\frac{\lambda}{2}\|\boldsymbol{u}\|_2^2}_{regularizer}.$$

The key questions

- Are all solutions with "zero loss" equally good?
- We want to do good not only on our **training set** $\{(\boldsymbol{x}_i, y_i)\}$...
- ...but also on yet unseen data (from a similar distribution)!
- **In our example,** we want our classifier to apply to new text documents.

From a data scientist's point of view

$$\min_{\boldsymbol{u},v} \underbrace{\frac{1}{n}\sum_{i=1}^{n} \max\left\{1 - y_i(\boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{u} - v), 0\right\}}_{loss} + \underbrace{\frac{\lambda}{2}\|\boldsymbol{u}\|_2^2}_{regularizer}.$$

The key questions

- Are all solutions with "zero loss" equally good?
- We want to do good not only on our **training set** $\{(\boldsymbol{x}_i, y_i)\}$...
- ...but also on yet unseen data (from a similar distribution)!
- **In our example,** we want our classifier to apply to new text documents.

Optimizers may not be able to do that efficiently.

Takeaways from the example

- We formulate the optimization problem based on observed data;
- We want the solution to have properties with respect to unseen data;
- Optimization may help but is not the ultimate answer.

Takeaways from the example

- We formulate the optimization problem based on observed data;
- We want the solution to have properties with respect to unseen data;
- Optimization may help but is not the ultimate answer.

Other issues with ML problems

- What if the feature space is large *(all French/English words)*?
- What if the parameter space $\mathbb{R}^n$ is huge *(all Wikipedia articles)*?
- What if linear models do not give good results?

Takeaways from the example

- We formulate the optimization problem based on observed data;
- We want the solution to have properties with respect to unseen data;
- Optimization may help but is not the ultimate answer.

Other issues with ML problems

- What if the feature space is large *(all French/English words)*?
  Reduce dimensionality, look for sparse solutions.
- What if the parameter space $\mathbb{R}^n$ is huge *(all Wikipedia articles)*?
  Sampling/Batch/Stochastic methods.
- What if linear models do not give good results?
  Nonlinear optimization (**kernel SVM**).

In this course

- Methodologies to solve given optimization problems;
- Focus on common structures in ML: finite sum, regularization;
- Discussion on properties of various formulations.

Focus on the optimization side

- Main algorithms and characteristics;
- Some applications, but always from an optimization perspective;
- Plenty of other data science courses in these Master programs!

Outline

Outline

Convention

For simplicity

- Optimization on real variables;
- Finite dimension;
- Canonical vector space structure.

I will use the following

- Scalars: $a, b, c, \ldots$
- Vectors: $\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c}, \ldots$
- Matrices: $\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}, \ldots$
- Sets: $\mathcal{A}, \mathcal{B}, \mathcal{C}, \ldots$

Linear algebra

- $\mathbb{R}^d$: set of vectors with $d \geq 1$ real components;
- For any $\boldsymbol{w} \in \mathbb{R}^d$ and $i \in \{1, \dots, d\}$, $w_i \in \mathbb{R}$ is the i-component of $\boldsymbol{w}$: $\boldsymbol{w} = [w_i]_{1 \leq i \leq d}$;
- Any $\boldsymbol{w} \in \mathbb{R}^d$ will be represented columnwise: $\boldsymbol{w} = \begin{bmatrix} w_1 \\ \vdots \\ w_d \end{bmatrix}$;
- We will use row vectors as "transposed" (from column to row) of their column vectors counterpart: $\boldsymbol{w}^{\mathrm{T}} := [w_1 \cdots w_d]$;

Linear algebra

- $\mathbb{R}^d$: set of vectors with $d \geq 1$ real components;
- For any $\boldsymbol{w} \in \mathbb{R}^d$ and $i \in \{1, \dots, d\}$, $w_i \in \mathbb{R}$ is the i-component of $\boldsymbol{w}$: $\boldsymbol{w} = [w_i]_{1 \leq i \leq d}$;
- Any $\boldsymbol{w} \in \mathbb{R}^d$ will be represented columnwise: $\boldsymbol{w} = \begin{bmatrix} w_1 \\ \vdots \\ w_d \end{bmatrix}$;
- We will use row vectors as "transposed" (from column to row) of their column vectors counterpart: $\boldsymbol{w}^{\mathrm{T}} := [w_1 \cdots w_d]$;

Vector operations

- *Addition in* $\mathbb{R}^d$*:* $\boldsymbol{w} + \boldsymbol{z} := [w_i + z_i]_{1 \leq i \leq d}$;
- *Multiply a vector in* $\mathbb{R}^d$ *by a real number:* $\lambda \boldsymbol{w} := [\lambda w_i]_{1 \leq i \leq d}$.

Linear algebra (2)

Euclidean norm on $\mathbb{R}^d$

The Euclidean norm (or ℓ_2 norm) of a vector $\boldsymbol{w} \in \mathbb{R}^d$ is given by:

$$\|\boldsymbol{w}\| := \sqrt{\sum_{i=1}^{d} w_i^2}.$$

Scalar product on $\mathbb{R}^d$

The scalar product is defined for every $\boldsymbol{w}, \boldsymbol{z} \in \mathbb{R}^d$ by:

$$\boldsymbol{w}^{\mathrm{T}} \boldsymbol{z} := \sum_{i=1}^{d} w_i z_i.$$

One thus has $\boldsymbol{w}^{\mathrm{T}} \boldsymbol{z} = \boldsymbol{z}^{\mathrm{T}} \boldsymbol{w} = \|\boldsymbol{w}\|^2$.

Linear algebra (3)

Matrices

- $\mathbb{R}^{n \times d}$: set of n-by-d matrices;
- $\mathbb{R}^{d \times 1} \simeq \mathbb{R}^d$.

Transposed matrix

Let $\boldsymbol{A} = [\boldsymbol{A}_{ij}] \in \mathbb{R}^{n \times d}$ be a matrix with n rows and d columns.
The *transposed matrix of* $\boldsymbol{A}$, denoted by $\boldsymbol{A}^{\mathrm{T}}$, is the matrix with n rows and m columns such that

$$\forall i = 1, \dots, n,\ \forall j = 1, \dots, d, \qquad \left[\boldsymbol{A}^{\mathrm{T}}\right]_{ij} = \boldsymbol{A}_{ji}.$$

Linear algebra (3)

Matrices

- $\mathbb{R}^{n \times d}$: set of n-by-d matrices;
- $\mathbb{R}^{d \times 1} \simeq \mathbb{R}^d$.

Transposed matrix

Let $\boldsymbol{A} = [\boldsymbol{A}_{ij}] \in \mathbb{R}^{n \times d}$ be a matrix with n rows and d columns.
The *transposed matrix of* $\boldsymbol{A}$, denoted by $\boldsymbol{A}^{\mathrm{T}}$, is the matrix with n rows and m columns such that

$$\forall i = 1, \dots, n, \ \forall j = 1, \dots, d, \qquad \left[\boldsymbol{A}^{\mathrm{T}}\right]_{ij} = \boldsymbol{A}_{ji}.$$

Squared matrix case

- $\boldsymbol{A}^{\mathrm{T}} \in \mathbb{R}^{d \times d}$;
- $\boldsymbol{A}$ is called a *symmetric matrix* if $\boldsymbol{A} = \boldsymbol{A}^{\mathrm{T}}$.

Linear algebra (4)

Matrix inversion

A matrix $\boldsymbol{A} \in \mathbb{R}^{d \times d}$ is *invertible* if it exists $\boldsymbol{B} \in \mathbb{R}^{d \times d}$ such that $\boldsymbol{BA} = \boldsymbol{AB} = \mathbf{I}_d$, where $\mathbf{I}_d$ is the identity matrix of $\mathbb{R}^{d \times d}$.
In this case, $\boldsymbol{B}$ is the unique matrix with this property: $\boldsymbol{B}$ is called the *inverse matrix of* $\boldsymbol{A}$, and is denoted by $\boldsymbol{A}^{-1}$.

Positive (semi-)definiteness

A matrix $\boldsymbol{A} \in \mathbb{R}^{d \times d}$ is *positive semidefinite* if

$$\forall \boldsymbol{x} \in \mathbb{R}^n, \quad \boldsymbol{x}^{\mathrm{T}} \boldsymbol{A} \boldsymbol{x} \geq 0.$$

It is called *positive definite* when $\boldsymbol{x}^{\mathrm{T}} \boldsymbol{A} \boldsymbol{x} > 0$ for every nonzero vector $\boldsymbol{x}$.

Linear algebra (5)

Eigenvalues and eigenvectors

Let $\boldsymbol{A} \in \mathbb{R}^{d \times d}$. A real λ is called an *eigenvalue of* $\boldsymbol{A}$ if

$$\exists \boldsymbol{v} \in \mathbb{R}^d, \|\boldsymbol{v}\| \neq 0, \qquad \boldsymbol{A}\boldsymbol{v} = \lambda \boldsymbol{v}.$$

The vector $\boldsymbol{v}$ is then called an *eigenvector of* $\boldsymbol{A}$ (associated to the eigenvalue λ.

Eigenvalues and eigenvectors

Let $\boldsymbol{A} \in \mathbb{R}^{d \times d}$. A real λ is called an *eigenvalue of $\boldsymbol{A}$* if

$$\exists \boldsymbol{v} \in \mathbb{R}^d, \|\boldsymbol{v}\| \neq 0, \qquad \boldsymbol{A}\boldsymbol{v} = \lambda \boldsymbol{v}.$$

The vector $\boldsymbol{v}$ is then called an *eigenvector of $\boldsymbol{A}$* (associated to the eigenvalue λ.

Any symmetric matrix in $\mathbb{R}^{d \times d}$ possesses d real eigenvalues. Given two symmetric matrices $(\boldsymbol{A}, \boldsymbol{B}) \in \mathbb{R}^{d \times d}$, we introduce the following notations:

- $\lambda_{\min}(\boldsymbol{A})/\lambda_{\max}(\boldsymbol{A})$: smallest/largest eigenvalue of $\boldsymbol{A}$;
- $\boldsymbol{A} \overset{n}{\succeq} \boldsymbol{B} \;\Leftrightarrow\; \lambda_{\min}(\boldsymbol{A}) \geq \lambda_{\max}(\boldsymbol{B})$;
- $\boldsymbol{A} \overset{n}{\succ} \boldsymbol{B} \Leftrightarrow \lambda_{\min}(\boldsymbol{A}) > \lambda_{\max}(\boldsymbol{B})$.

With these notations, $\boldsymbol{A}$ is positive semi-definite (resp. positive definite) if and only if $\boldsymbol{A} \succeq 0$ (resp. $\boldsymbol{A} \succ 0$).

Differential calculus

We consider a smooth function $f : \mathbb{R}^d \to \mathbb{R}$.

We consider a **smooth** function $f : \mathbb{R}^d \to \mathbb{R}$.

First-order derivative

If f is continuously differentiable on $\mathbb{R}^d$, one defines for any $\boldsymbol{w} \in \mathbb{R}^d$ the **gradient of f at w** by

$$\nabla f(\boldsymbol{w}) \;:=\; \left[\frac{\partial f}{\partial w_i}\right]_{1\le i\le d} \in \mathbb{R}^d.$$

The set of continuously differentiable functions will be denoted by $\mathcal{C}^1 \overset{n}{=} \mathcal{C}^1(\mathbb{R}^d, \mathbb{R})$.

Differential calculus

We consider a **smooth** function $f : \mathbb{R}^d \to \mathbb{R}$.

We consider a **smooth** function $f : \mathbb{R}^d \to \mathbb{R}$.

Second-order derivative

If f is *twice* continuously differentiable on $\mathbb{R}^d$, one defines for any $\boldsymbol{w} \in \mathbb{R}^d$ the **Hessian of f at w** by

$$\nabla^2 f(\boldsymbol{w}) \ := \ \left[\frac{\partial^2 f}{\partial w_i \partial w_j} \right]_{1 \le i,j \le d} \in \mathbb{R}^{d \times d}.$$

This matrix is symmetric.
The set of twice continuously differentiable functions will be denoted by $\mathcal{C}^2$.

Differential calculus (2)

First-order Taylor expansions

If $f \in \mathcal{C}^1$, for any $\boldsymbol{w}, \boldsymbol{h} \in \mathbb{R}^d$,

$$\begin{cases} f(\boldsymbol{w}+\boldsymbol{h}) = f(\boldsymbol{w}) + \nabla f(\boldsymbol{w} + t\, h)^\mathrm{T} \boldsymbol{h} & \text{for some } t \in (0,1) \\ f(\boldsymbol{w}+\boldsymbol{h}) = f(\boldsymbol{w}) + \int_0^1 \nabla f(\boldsymbol{w} + t\, h)^\mathrm{T} \boldsymbol{h}\, dt. \end{cases}$$

First-order Taylor expansions

If $f \in \mathcal{C}^1$, for any $\boldsymbol{w}, \boldsymbol{h} \in \mathbb{R}^d$,

$$\begin{cases} f(\boldsymbol{w}+\boldsymbol{h}) = f(\boldsymbol{w}) + \nabla f(\boldsymbol{w} + t\,h)^{\mathrm{T}} \boldsymbol{h} & \text{for some } t \in (0,1) \\ f(\boldsymbol{w}+\boldsymbol{h}) = f(\boldsymbol{w}) + \int_0^1 \nabla f(\boldsymbol{w} + t\,h)^{\mathrm{T}} \boldsymbol{h}\, dt. \end{cases}$$

Second-order Taylor expansions

If $f \in \mathcal{C}^2$, for any $\boldsymbol{w}, \boldsymbol{h} \in \mathbb{R}^d$,

$$\begin{cases} f(\boldsymbol{w}+\boldsymbol{h}) = f(\boldsymbol{w}) + \nabla f(\boldsymbol{w})^{\mathrm{T}} \boldsymbol{h} + \frac{1}{2} \boldsymbol{h}^{\mathrm{T}} \nabla^2 f(\boldsymbol{w} + t\,\boldsymbol{h}) \boldsymbol{h} \\ \qquad \text{for some } t \in (0,1) \\ f(\boldsymbol{w}+\boldsymbol{h}) = f(\boldsymbol{w}) + \nabla f(\boldsymbol{w})^{\mathrm{T}} \boldsymbol{h} + \frac{1}{2} \int_0^1 \boldsymbol{h}^{\mathrm{T}} \nabla^2 f(\boldsymbol{w} + t\,\boldsymbol{h}) \boldsymbol{h}\, dt. \end{cases}$$

Definition

A function $g : \mathbb{R}^d \rightarrow \mathbb{R}^m$ is L-Lipschitz continuous if it exists $L > 0$ such that

$$\forall (\boldsymbol{w}, \boldsymbol{z}) \in (\mathbb{R}^d)^2, \quad \|g(\boldsymbol{w}) - g(\boldsymbol{z})\| \leq L\|\boldsymbol{w} - \boldsymbol{z}\|.$$

The value L is called a Lipschitz constant for g.

- Ex) Any linear function is Lipschitz continuous;
- $\mathcal{C}_L^{1,1}$: set continuously differentiable functions with L-Lipschitz continuous first-order derivative;
- $\mathcal{C}_L^{2,2}$: set of twice continuously differentiable functions with L-Lipschitz continuous second-order derivative.

First-order Taylor bound

Let $f \in \mathcal{C}_L^{1,1}$. For any $\boldsymbol{w}, \boldsymbol{h} \in \mathbb{R}^d$,

$$f(\boldsymbol{w}+\boldsymbol{h}) \le f(\boldsymbol{w}) + \nabla f(\boldsymbol{w})^{\mathrm{T}} \boldsymbol{h} + \frac{L}{2}\|\boldsymbol{h}\|^2.$$

First-order Taylor bound

Let $f \in \mathcal{C}^{1,1}_L$. For any $\boldsymbol{w}, \boldsymbol{h} \in \mathbb{R}^d$,

$$f(\boldsymbol{w}+\boldsymbol{h}) \le f(\boldsymbol{w}) + \nabla f(\boldsymbol{w})^{\mathrm{T}} \boldsymbol{h} + \frac{L}{2}\|\boldsymbol{h}\|^2.$$

$\Rightarrow$ **One of the two key inequalities in optimization.**

Lipschitz continuity and Taylor bounds

First-order Taylor bound

Let $f \in \mathcal{C}_L^{1,1}$. For any $\boldsymbol{w}, \boldsymbol{h} \in \mathbb{R}^d$,

$$f(\boldsymbol{w}+\boldsymbol{h}) \le f(\boldsymbol{w}) + \nabla f(\boldsymbol{w})^{\mathrm{T}}\boldsymbol{h} + \frac{L}{2}\|\boldsymbol{h}\|^2.$$

$\Rightarrow$ **One of the two key inequalities in optimization.**

Second-order Taylor bound

Let $f \in \mathcal{C}_L^{2,2}$. For any $\boldsymbol{w}, \boldsymbol{h} \in \mathbb{R}^d$,

$$f(\boldsymbol{w}+\boldsymbol{h}) \le f(\boldsymbol{w}) + \nabla f(\boldsymbol{w})^{\mathrm{T}}\boldsymbol{h} + \tfrac{1}{2}\boldsymbol{h}^{\mathrm{T}}\nabla^2 f(\boldsymbol{w})\boldsymbol{h} + \frac{L}{6}\|\boldsymbol{h}\|^3,$$

Some references

- Plenty of lecture notes, courses freely available;
- **Appendix material** of many optimization (and some ML) textbooks!

Examples (subject to updates)

- In French:
  https://www.lpsm.paris/pageperso/bolley/poly-cdiff.pdf
  https://www.lpsm.paris/pageperso/bolley/poly-algebre3.pdf
- In English:
  http://vmls-book.stanford.edu/vmls.pdf (Chapters 1-3)
  https://sebastianraschka.com/pdf/books/dlb/appendix_d_calculus.pdf

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.

Formulation of an optimization problem

A **minimization problem** of d real parameters is written as follows :

$$\min_{\boldsymbol{w} \in \mathbb{R}^d} f(x) \quad \text{subject to} \quad \boldsymbol{w} \in \mathcal{F}$$

Formulation of an optimization problem

A **minimization problem** of d real parameters is written as follows :

$$\min_{\boldsymbol{w} \in \mathbb{R}^d} f(x) \quad \text{subject to} \quad \boldsymbol{w} \in \mathcal{F}$$

- $\boldsymbol{w}$ 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;
- $\mathcal{F}$ is the constraint/feasible set.

Formulation of an optimization problem

A **minimization problem** of d real parameters is written as follows :

$$\min_{\boldsymbol{w} \in \mathbb{R}^d} f(x) \quad \text{subject to} \quad \boldsymbol{w} \in \mathcal{F}$$

- $\boldsymbol{w}$ 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;
- $\mathcal{F}$ is the constraint/feasible set.

Maximizing f is equivalent to minimizing $-f$.

Local and global solutions

$$\min_{\boldsymbol{w}\in\mathbb{R}^d} f(\boldsymbol{w}) \quad \text{subject to} \quad \boldsymbol{w} \in \mathcal{F}$$

Local minimum (also called minimizer)

- A point $\boldsymbol{w}^*$ is a **local minimum** of the problem if there exists a neighborhood $\mathcal{N}$ of $\boldsymbol{w}^*$ such that $f(\boldsymbol{w}^*) \le f(\boldsymbol{w})\ \forall \boldsymbol{w} \in \mathcal{N} \cap \mathcal{F}$;
- A local minimum such that $f(\boldsymbol{w}^*) < f(\boldsymbol{w})\ \forall \boldsymbol{w} \in \mathcal{N} \cap \mathcal{F},\ \boldsymbol{w} \neq \boldsymbol{w}^*$ is called a strict local minimum.

Global minimum

A point $\boldsymbol{w}^*$ is a **global minimum** of the problem if $f(\boldsymbol{w}^*) \le f(\boldsymbol{w})\ \forall \boldsymbol{w} \in \mathcal{F}$.

Local and global solutions (2)

- In general, finding global solutions is hard;
- Local solutions can also be hard to find.

Local and global solutions (2)

- In general, finding global solutions is hard;
- Local solutions can also be hard to find.

Tractable cases

- When the objective function behaves nicely;
- Suitable properties of the constraint set (more on that in the constrained optimization lecture).

Unconstrained problem: $\min_{\boldsymbol{w} \in \mathbb{R}^d} f(\boldsymbol{w})$,
f continuously differentiable.

Optimality conditions for unconstrained optimization

Unconstrained problem: $\min_{\boldsymbol{w} \in \mathbb{R}^d} f(\boldsymbol{w})$,
f continuously differentiable.

First-order necessary condition

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

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

Optimality conditions for unconstrained optimization

Unconstrained problem: $\min_{\mathbf{w} \in \mathbb{R}^d} f(\mathbf{w})$,
f continuously differentiable.

First-order necessary condition

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

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

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

Optimality conditions for unconstrained optimization (2)

Unconstrained problem: $\min_{\boldsymbol{w} \in \mathbb{R}^d} f(\boldsymbol{w})$,
f twice continuously differentiable.

Unconstrained problem: $\min_{\boldsymbol{w} \in \mathbb{R}^d} f(\boldsymbol{w})$,
f twice continuously differentiable.

Second-order necessary condition

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

$$\|\nabla f(\boldsymbol{w}^*)\| = 0 \quad \text{and} \quad \nabla^2 f(\boldsymbol{w}^*) \succeq 0.$$

Unconstrained problem: $\min_{\boldsymbol{w} \in \mathbb{R}^d} f(\boldsymbol{w})$,
f twice continuously differentiable.

Second-order necessary condition

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

$$\|\nabla f(\boldsymbol{w}^*)\| = 0 \quad \text{and} \quad \nabla^2 f(\boldsymbol{w}^*) \succeq 0.$$

Second-order sufficient condition

If $\boldsymbol{w}^*$ is such that

$$\|\nabla f(\boldsymbol{w}^*)\| = 0 \quad \text{and} \quad \nabla^2 f(\boldsymbol{w}^*) \succ 0,$$

then it is a local minimum of the problem.

Outline

1. Introduction

2. Basics of optimization
   - Notation and background
   - Optimization problem and optimality
   - Convexity
   - Optimization algorithms

3. Unconstrained optimization

Convex set

A set $\mathcal{C} \in \mathbb{R}^d$ is called **convex** if

$$\forall (\boldsymbol{u}, \boldsymbol{v}) \in \mathcal{C}^2,\ \forall t \in [0,1], \qquad t\boldsymbol{u} + (1-t)\boldsymbol{v} \in \mathcal{C}.$$

First definitions

Convex set

A set $\mathcal{C} \in \mathbb{R}^d$ is called **convex** if

$$\forall (\boldsymbol{u}, \boldsymbol{v}) \in \mathcal{C}^2, \ \forall t \in [0,1], \qquad t\boldsymbol{u} + (1-t)\boldsymbol{v} \in \mathcal{C}.$$

Examples:

- $\mathbb{R}^d$;
- Line segment: $\{t\boldsymbol{w} | t \in \mathbb{R}\}$ for some $\boldsymbol{w} \in \mathbb{R}^d$;
- Sphere: $\left\{\boldsymbol{w} \in \mathbb{R}^d \middle| \sum_i [\boldsymbol{w}_i]^2 \leq 1\right\}$.

Convex function

Generic definition

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

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

Convex function

Generic definition

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

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

Examples:

- Linear function: $f(\boldsymbol{w}) = \boldsymbol{a}^{\mathrm{T}} \boldsymbol{w} + b$;
- Squared Euclidean norm: $f(\boldsymbol{w}) = \|\boldsymbol{w}\|^2 = \boldsymbol{w}^{\mathrm{T}} \boldsymbol{w}$.

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 $f : \mathbb{R}^d \rightarrow \mathbb{R}$ is convex if and only if

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

The other key inequality in optimization.

Smooth convex functions

Convexity and gradient

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

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

The other key inequality in optimization.

Convexity and Hessian

A twice continuously differentiable function $f : \mathbb{R}^d \rightarrow \mathbb{R}$ is convex if and only if for every $\boldsymbol{w} \in \mathbb{R}^d$, $\nabla^2 f(\boldsymbol{w}) \succeq 0$.

Convex optimization problem

$$\min_{\boldsymbol{w} \in \mathcal{X}} f(\boldsymbol{w}), f \text{ convex}, \mathcal{X} \subset \mathbb{R}^d \text{ closed+convex.}$$

Convex optimization problem

$$\min_{\boldsymbol{w} \in \mathcal{X}} f(\boldsymbol{w}), f \text{ convex}, \mathcal{X} \subset \mathbb{R}^d \text{ closed+convex}.$$

Theorem

Every local minimum of a f is a global minimum.

Convex optimization problem

$$\min_{\boldsymbol{w} \in \mathcal{X}} f(\boldsymbol{w}), f \text{ convex}, \mathcal{X} \subset \mathbb{R}^d \text{ closed+convex.}$$

Theorem

Every local minimum of a f is a global minimum.

Corollary

If f is continuously differentiable, every point $\boldsymbol{w}^*$ such that $\|\nabla f(\boldsymbol{w}^*)\| = 0$ is a global minimum.

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 $t \in [0, 1]$,

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

Strong convexity

Definition

A function $f : \mathbb{R}^d \to \mathbb{R}$ 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 $t \in [0,1]$,

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

Theorem

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

Strong convexity (2)

Gradient and strong convexity

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

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

Hessian and strong convexity

Let $f : \mathbb{R}^d \rightarrow \mathbb{R}$, $f \in \mathcal{C}^2$. Then,

$$f \text{ is } \mu\text{-strongly convex} \quad \Longleftrightarrow \quad \nabla^2 f(\boldsymbol{w}) \succeq \mu \mathbf{I} \;\; \forall \boldsymbol{w} \in \mathbb{R}^d.$$

Examples of (strongly) convex problems

Minimize a convex quadratic

$$\min_{\boldsymbol{w} \in \mathbb{R}^d} f(\boldsymbol{w}) := \frac{1}{2}\boldsymbol{w}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{w} + \boldsymbol{b}^{\mathrm{T}}\boldsymbol{w}, \quad \boldsymbol{A} \succeq 0.$$

- $\nabla^2 f(\boldsymbol{w}) = \boldsymbol{A}$;
- Strongly convex if $\boldsymbol{A} \succ 0$, with $\mu = \lambda_{\min}(\boldsymbol{A})$.

Examples of (strongly) convex problems

Minimize a convex quadratic

$$\min_{\boldsymbol{w} \in \mathbb{R}^d} f(\boldsymbol{w}) := \frac{1}{2}\boldsymbol{w}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{w} + \boldsymbol{b}^{\mathrm{T}}\boldsymbol{w}, \quad \boldsymbol{A} \succeq 0.$$

- $\nabla^2 f(\boldsymbol{w}) = \boldsymbol{A}$;
- Strongly convex if $\boldsymbol{A} \succ 0$, with $\mu = \lambda_{\min}(\boldsymbol{A})$.

Projection onto a closed, convex set

$$\min_{\boldsymbol{w} \in \mathcal{X}} \frac{1}{2}\|\boldsymbol{w} - \boldsymbol{a}\|^2, \quad \mathcal{X} \text{ closed, convex.}$$

- The objective is 1-strongly convex $\Rightarrow$ the problem has a unique solution;
- Generalization of the case $\mathcal{X} = \mathbb{R}^d$.

Outline

Three ways to study optimization problems

- **Mathematical :** Prove existence of solutions, well-posedness of a problem. Study complex optimization formulations.
- **Computational :** Write a piece of software to solve specific or generic optimization problems in practice.
- **Algorithmic :** Design algorithms, establish theoretical guarantees and validate their practical implementation.

Three ways to study optimization problems

- **Mathematical :** Prove existence of solutions, well-posedness of a problem. Study complex optimization formulations.
- **Computational :** Write a piece of software to solve specific or generic optimization problems in practice.
- **Algorithmic :** Design algorithms, establish theoretical guarantees and validate their practical implementation.

This course is about the third category.

How to solve an optimization problem?

The ideal approach

- Find the solutions of $\|\nabla f(\boldsymbol{w})\| = 0$;
- Choose the one with the lowest function value.

How to solve an optimization problem?

The ideal approach

- Find the solutions of $\|\nabla f(\boldsymbol{w})\| = 0$;
- Choose the one with the lowest function value.

What's wrong with that?

- Solving a nonlinear equation directly is hard;
- There can be infinitely many solutions;
- The procedure has to be implemented eventually.

How we shall proceed

Iterative procedures

- Driving principle : given the current solution, move towards a (potentially) better point;
- Requires a certain amount of calculation at every iteration.

Our goal in the rest of the course

- Propose several algorithms;
- Analyze their theoretical behavior and guarantees;
- Check their practical appeal (lab sessions).

What do we expect?

In order to solve $\min_{\boldsymbol{w} \in \mathbb{R}^d} f(\boldsymbol{w})$, we hope to achieve one of the following:

1. The iterates should get close to a solution;
2. The function values should get close to the optimum;
3. The optimality conditions should get close to be satisfied.

What do we expect?

In order to solve $\min_{\mathbf{w} \in \mathbb{R}^d} f(\mathbf{w})$, we hope to achieve one of the following:

1. The iterates should get close to a solution;
2. The function values should get close to the optimum;
3. The optimality conditions should get close to be satisfied.

Convergence of iterates

The method generates a sequence of points (iterates) $\{\mathbf{w}_k\}_k$ such that

$$\|\mathbf{w}_k - \mathbf{w}^*\| \to 0 \quad \text{when } k \to \infty,$$

where $\mathbf{w}^*$ is an optimal value of the problem.
(Typical of (strongly) convex functions.)

What do we expect? ('ed)

Convergence in function value

$$f(\boldsymbol{w}_k) \to f^* \quad \text{when } k \to \infty,$$

where f^* is the optimal value of the problem.
(Typical of (strongly) convex functions.)

Convergence in function value

$$f(\mathbf{w}_k) \to f^* \quad \text{when } k \to \infty,$$

where f^* is the optimal value of the problem.
(Typical of (strongly) convex functions.)

Convergence to a stationary point for differentiable f

$$\|\nabla f(\mathbf{w}_k)\| \to 0 \quad \text{when } k \to \infty.$$

More generic condition.

Unlike in theory, in practice:

- We do not know the optimal solution(s);
- We do not know the optimal value.

Why these conditions?

Unlike in theory, in practice:

- We do not know the optimal solution(s);
- We do not know the optimal value.

From an algorithmic standpoint,

- We can measure the behavior of the iterates;
- We can evaluate the objective and try to decrease it iteratively;
- We can evaluate/estimate the gradient norm and measure its decrease to zero.

Remark: Convergence and convergence rates

- In optimization, classical results are asymptotic:

$$\|\nabla f(\boldsymbol{w}_k)\| \rightarrow 0 \quad \text{when } k \rightarrow \infty.$$

Remark: Convergence and convergence rates

- In optimization, classical results are asymptotic:

$$\|\nabla f(\boldsymbol{w}_k)\| \to 0 \quad \text{when } k \to \infty.$$

- **Global convergence rates** are now very popular:

$$\|\nabla f(\boldsymbol{w}_k)\| = \mathcal{O}\left(\frac{1}{k}\right) \quad \Leftrightarrow \quad \exists C > 0, \|\nabla f(\boldsymbol{w}_k)\| \leq \frac{C}{k} \ \forall k.$$

- Common in convex optimization;
- Standard in theoretical computer science/statistics.

Optimizers code in...

- `C/C++/Fortran` (high-performance computing)
- `Matlab`, `Python` (prototyping);
- `Julia`.

On the computational side

Optimizers code in...

- `C/C++/Fortran` (high-performance computing)
- `Matlab`, `Python` (prototyping);
- `Julia`.

Specific optimization modeling languages

- `GAMS`, `AMPL`, `CVX` are broad-spectrum languages;
- `MATPOWER`, `PyTorch` are domain-oriented;
- Can be interfaced with the languages above.

Conclusions: basics of optimization

Modeling framework

- Objective, constraints;
- Characterization of the solutions.

Conclusions: basics of optimization

Modeling framework

- Objective, constraints;
- Characterization of the solutions.

Important tools

- Derivatives and Taylor expansion;
- Convexity.

Conclusions: basics of optimization

Modeling framework

- Objective, constraints;
- Characterization of the solutions.

Important tools

- Derivatives and Taylor expansion;
- Convexity.

Algorithmic principle

- Iterative process: find a sequence of points that leads to a solution;
- Quantify how fast.

Outline

Introduction

Our problem today

$$\min_{\boldsymbol{w} \in \mathbb{R}^d} f(\boldsymbol{w}).$$

Assumptions

- f bounded below by f^*;
- f smooth $\Rightarrow$ derivatives can be used to solve this problem.

Two categories

Least squares

- Heavily relies on linear algebra;
- Can precisely characterize the solution(s);
- Application: Linear regression problems.

Generic smooth unconstrained problems

- One tool from analysis: the gradient;
- Goal: Converge iteratively towards a solution;
- Application(s): Logistic regression (among others).

Aims of this lecture

- Survey classical techniques;
- Illustrate how they may be used in ML problems.

- Highlight the role of the gradient and that of convexity;
- Show how convergence rates are obtained.

Outline

Context

Data

- Dataset with n elements (individuals, trials, samples, etc);
- Every element i is characterized by a vector $\boldsymbol{x}_i \in \mathbb{R}^d$ of features and a label $y_i \in \mathbb{R}$.

$$\Rightarrow \text{Matrix } \boldsymbol{X} = \begin{bmatrix} \boldsymbol{x}_1^{\mathrm{T}} \\ \vdots \\ \boldsymbol{x}_n^{\mathrm{T}} \end{bmatrix} \in \mathbb{R}^{n \times d} \text{ and vector } \boldsymbol{y} = \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix}.$$

Context

Data

- Dataset with n elements (individuals, trials, samples, etc);
- Every element i is characterized by a vector $\boldsymbol{x}_i \in \mathbb{R}^d$ of features and a label $y_i \in \mathbb{R}$.

$$\Rightarrow \text{Matrix } \boldsymbol{X} = \begin{bmatrix} \boldsymbol{x}_1^{\mathrm{T}} \\ \vdots \\ \boldsymbol{x}_n^{\mathrm{T}} \end{bmatrix} \in \mathbb{R}^{n \times d} \text{ and vector } \boldsymbol{y} = \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix}.$$

Goal

We seek a linear predictor function $h : \boldsymbol{x} \mapsto \boldsymbol{x}^{\mathrm{T}} \boldsymbol{w}$ that correctly predicts y_i from $\boldsymbol{x}_i$.

- Linear models often provide a good first approximation;
- Relies on linear algebra, a rich area both theoretically and computationally.

Ideal predictor

- Would achieve $h(\mathbf{x}_i) = \mathbf{x}_i^{\mathrm{T}} \mathbf{w} = y_i$ for every i;
- These n equations can be written under the form of a linear system: $\mathbf{X} \mathbf{w} = \mathbf{y}$.

From linear models to linear systems

Ideal predictor

- Would achieve $h(\mathbf{x}_i) = \mathbf{x}_i^{\mathrm{T}} \mathbf{w} = y_i$ for every i;
- These n equations can be written under the form of a linear system: $\mathbf{X}\mathbf{w} = \mathbf{y}$.

Solving linear systems of equations

- A purely linear algebra problem;
- The solution is completely characterized by the properties of $\mathbf{X}$ and $\mathbf{y}$.

Here's the catch

A dataset

- $\mathbf{x}_1 = \mathbf{x}_2 = \cdots = \mathbf{x}_n = 1$ $(d = 1)$;
- $y_1, \dots, y_n$ are distinct (typical of noisy measurements).

Here's the catch

A dataset

- $\mathbf{x}_1 = \mathbf{x}_2 = \cdots = \mathbf{x}_n = 1$ $(d = 1)$;
- $y_1, \dots, y_n$ are distinct (typical of noisy measurements).

Fitting a linear model

- We seek $\mathbf{w} = w \in \mathbb{R}$ such that $\mathbf{x}_i^\mathrm{T} \mathbf{w} = x_i w = y_i \ \forall i$;
- The corresponding linear system is:

$$\left\{ \begin{array}{rcl} w & = & y_1 \\ w & = & y_2 \\ \vdots & & \\ w & = & y_n \end{array} \right.$$

Here's the catch

A dataset

- $\mathbf{x}_1 = \mathbf{x}_2 = \cdots = \mathbf{x}_n = 1$ $(d = 1)$;
- $y_1, \dots, y_n$ are distinct (typical of noisy measurements).

Fitting a linear model

- We seek $\mathbf{w} = w \in \mathbb{R}$ such that $\mathbf{x}_i^\mathrm{T} \mathbf{w} = x_i w = y_i \ \forall i$;
- The corresponding linear system is:

$$\left\{ \begin{array}{lcl} w & = & y_1 \\ w & = & y_2 \\ \vdots & & \\ w & = & y_n \end{array} \right.$$

- This system does not have a solution!
- Yet it is possible to compute a solution to the "data fitting" problem.

Linear least squares

Problem formulation

Given a data set $\{(\boldsymbol{x}_i, y_i)\}_{1 \le i \le n}$ where $\boldsymbol{x}_i \in \mathbb{R}^d$, compute $\boldsymbol{w}^* \in \mathbb{R}^d$ as a solution of

$$\min_{\boldsymbol{w} \in \mathbb{R}^d} \frac{1}{2} \|\boldsymbol{X}\boldsymbol{w} - \boldsymbol{y}\|^2 = \frac{1}{2}(\boldsymbol{X}\boldsymbol{w} - \boldsymbol{y})^{\mathrm{T}}(\boldsymbol{X}\boldsymbol{w} - \boldsymbol{y}),$$

where $\boldsymbol{X} = \begin{bmatrix} \boldsymbol{x}_1^{\mathrm{T}} \\ \vdots \\ \boldsymbol{x}_n^{\mathrm{T}} \end{bmatrix} \in \mathbb{R}^{n \times d}$ and $\boldsymbol{y} = \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix} \in \mathbb{R}^n$.

Linear least squares

Problem formulation

Given a data set $\{(\boldsymbol{x}_i, y_i)\}_{1 \le i \le n}$ where $\boldsymbol{x}_i \in \mathbb{R}^d$, compute $\boldsymbol{w}^* \in \mathbb{R}^d$ as a solution of

$$\min_{\boldsymbol{w} \in \mathbb{R}^d} \frac{1}{2} \|\boldsymbol{X}\boldsymbol{w} - \boldsymbol{y}\|^2 = \frac{1}{2}(\boldsymbol{X}\boldsymbol{w} - \boldsymbol{y})^{\mathrm{T}}(\boldsymbol{X}\boldsymbol{w} - \boldsymbol{y}),$$

where $\boldsymbol{X} = \begin{bmatrix} \boldsymbol{x}_1^{\mathrm{T}} \\ \vdots \\ \boldsymbol{x}_n^{\mathrm{T}} \end{bmatrix} \in \mathbb{R}^{n \times d}$ and $\boldsymbol{y} = \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix} \in \mathbb{R}^n$.

Characteristics

- Unconstrained optimization problem;
- Nonnegative objective function (values bounded below by 0);
- Smooth: polynomial in the coefficients of $\boldsymbol{w}$.

How to solve linear least squares

$$\min_{\boldsymbol{w} \in \mathbb{R}^d} \frac{1}{2} \left\| \boldsymbol{X} \boldsymbol{w} - \boldsymbol{y} \right\|^2 .$$

How to solve linear least squares

$$\min_{\boldsymbol{w} \in \mathbb{R}^d} \frac{1}{2} \left\| \boldsymbol{X} \boldsymbol{w} - \boldsymbol{y} \right\|^2 .$$

- If $\boldsymbol{w}^*$ is a solution of the linear system $\boldsymbol{X} \boldsymbol{w} = \boldsymbol{y}$, then it is a solution of the least-squares problem!

How to solve linear least squares

$$\min_{\boldsymbol{w} \in \mathbb{R}^d} \frac{1}{2} \|\boldsymbol{X}\boldsymbol{w} - \boldsymbol{y}\|^2 .$$

- If $\boldsymbol{w}^*$ is a solution of the linear system $\boldsymbol{X}\boldsymbol{w} = \boldsymbol{y}$, then it is a solution of the least-squares problem!
- What happens when the system has no solution?

Solving a linear system: the nice case

Squared linear system

$\boldsymbol{X}\,\boldsymbol{w} = \boldsymbol{y}$, avec $\boldsymbol{X} \in \mathbb{R}^{n \times d}$ and $n = d$.

Case 1: $\boldsymbol{X}$ possesses an inverse

$$\boldsymbol{X}\,\boldsymbol{w} = \boldsymbol{y} \;\Leftrightarrow\; \boldsymbol{w} = \boldsymbol{X}^{-1}\boldsymbol{y}.$$

The system possesses a unique solution $\boldsymbol{w}^* = \boldsymbol{X}^{-1}\boldsymbol{y}$, which is also the global minimum of the least-squares problem $\min_{\boldsymbol{w} \in \mathbb{R}^d} \frac{1}{2}\|\boldsymbol{X}\,\boldsymbol{w} - \boldsymbol{y}\|^2$.

Example with $d = n = 2$

$$\left\{ \begin{array}{rcl} w_1 + w_2 & = & 0, \\ 3w_1 + 2w_2 & = & 1. \end{array} \right.$$

The unique solution is $\boldsymbol{w} = [1 \; -1]^{\mathrm{T}}$.

Solving a linear system: the other cases

Squared linear system

$\boldsymbol{X}\,\boldsymbol{w} = \boldsymbol{y}$, avec $\boldsymbol{X} \in \mathbb{R}^{n \times d}$ and $n = d$.

Case 2: $\boldsymbol{X}$ is not invertible

- There could be no solution;
- There could be infinitely many.

In both cases, we can compute a solution in the least-squares sense!

Solving a linear system: the other cases

Squared linear system

$\boldsymbol{X}\,\boldsymbol{w} = \boldsymbol{y}$, avec $\boldsymbol{X} \in \mathbb{R}^{n \times d}$ and $n = d$.

Case 2: $\boldsymbol{X}$ is not invertible

- There could be no solution;
- There could be infinitely many.

In both cases, we can compute a solution in the least-squares sense!

Other cases

- $\boldsymbol{X}\,\boldsymbol{w} = \boldsymbol{y}$, avec $\boldsymbol{X} \in \mathbb{R}^{n \times d}$, $n \neq d$;
- Can have no solution, one or infinitely many!

Linear algebra to the rescue

What we want

- An analogous of the inverse;
- Provides a solution when there exists one (or infinitely many).

Linear algebra to the rescue

What we want

- An analogous of the inverse;
- Provides a solution when there exists one (or infinitely many).

Pseudo-inverse

Given a matrix $\boldsymbol{X} \in \mathbb{R}^{n \times d}$, there exists a matrix $\boldsymbol{A} \in \mathbb{R}^{d \times n}$ that satisfies the Moore-Penrose equations:

$$\begin{cases} \boldsymbol{AXA} &= \boldsymbol{A} \\ \boldsymbol{XAX} &= \boldsymbol{X} \end{cases} \quad \text{and} \quad \begin{cases} (\boldsymbol{AX})^{\mathrm{T}} &= \boldsymbol{AX} \\ (\boldsymbol{XA})^{\mathrm{T}} &= \boldsymbol{XA} \end{cases}$$

This matrix is called the pseudo-inverse of $\boldsymbol{X}$, and we note $\boldsymbol{A} = \boldsymbol{X}^{\dagger}$.
If $\boldsymbol{X}$ is invertible, $\boldsymbol{X}^{\dagger} = \boldsymbol{X}^{-1}$.

Least-squares and pseudo-inverse

$$\boldsymbol{X} \in \mathbb{R}^{n \times d}, \qquad \boldsymbol{y} \in \mathbb{R}^n.$$

Theorem

For any $\boldsymbol{y} \in \mathbb{R}^n$, $\boldsymbol{X}^\dagger \boldsymbol{y}$ is the solution of the least-squares problem

$$\min_{\boldsymbol{w} \in \mathbb{R}^d} f(\boldsymbol{w}) = \frac{1}{2}\|\boldsymbol{X}\boldsymbol{w} - \boldsymbol{y}\|^2.$$

with minimal norm. That is, for any $\hat{\boldsymbol{w}}$ solution of the problem:

$$\boldsymbol{X} \in \mathbb{R}^{n \times d}, \qquad \boldsymbol{y} \in \mathbb{R}^n.$$

Theorem

For any $\boldsymbol{y} \in \mathbb{R}^n$, $\boldsymbol{X}^\dagger \boldsymbol{y}$ is the solution of the least-squares problem

$$\min_{\boldsymbol{w} \in \mathbb{R}^d} f(\boldsymbol{w}) = \frac{1}{2}\|\boldsymbol{X}\boldsymbol{w} - \boldsymbol{y}\|^2.$$

with minimal norm. That is, for any $\hat{\boldsymbol{w}}$ solution of the problem:

- $f\left(\boldsymbol{X}^\dagger \boldsymbol{y}\right) = f(\hat{\boldsymbol{w}})$;
- $\|\boldsymbol{X}^\dagger \boldsymbol{y}\| \leq \|\hat{\boldsymbol{w}}\|$;
- $\boldsymbol{X}^\dagger \boldsymbol{y}$ can be represented using less information than $\hat{\boldsymbol{w}}$.

Back to our example

$$\boldsymbol{X}\boldsymbol{w} = \boldsymbol{y}, \qquad \boldsymbol{X} = \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix}, \ \boldsymbol{y} = \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix}.$$

Finding a solution

- The least-squares problem $\min_{\boldsymbol{w} \in \mathbb{R}^d} \frac{1}{2}\|\boldsymbol{X}\boldsymbol{w} - \boldsymbol{y}\|^2$ has infinitely many solutions;
- Among them, $\boldsymbol{w}^* = \boldsymbol{X}^\dagger \boldsymbol{y}$ is the one with minimal norm;
- This solution turns out to be the mean $\boldsymbol{w}^* = \frac{1}{n}\sum_{i=1}^n y_i$!

Key points

- Computing the pseudo-inverse;
- Or an approximation thereof!

What linear algebra solvers can do

- Put the data matrix $\boldsymbol{X}$ in a nicer form (QR, LU, SVD, etc) easier to (pseudo-)invert;
- Use iterative linear algebra routines (LSQR, LSLQ, etc) to compute an approximate solution, paying attention to round-off errors;
- **Run in parallel/distributed environments**.

Two applications among many more

Least-squares formulations

- Naturally arise in a plurality of fields that try to minimize the error between a model and some data
  Ex) weather forecasting, statistics, economy.
- Some problems can also be formulate or reformulated as linear least squares.

Two applications among many more

Least-squares formulations

- Naturally arise in a plurality of fields that try to minimize the error between a model and some data
  Ex) weather forecasting, statistics, economy.
- Some problems can also be formulate or reformulated as linear least squares.

Two illustrations

1. Rewrite an optimization problem as a linear least-squares problem;
2. Form a linear least-squares formulation of a problem.

Illustration: Minimization of quadratic functions

Problem

Given a symmetric matrix $\boldsymbol{A} \in \mathbb{R}^{d \times d}$ and a vector $\boldsymbol{b} \in \mathbb{R}^d$, solve

$$\min_{\boldsymbol{w} \in \mathbb{R}^d} \frac{1}{2} \boldsymbol{w}^{\mathrm{T}} \boldsymbol{A} \boldsymbol{w} - \boldsymbol{b}^{\mathrm{T}} \boldsymbol{w}.$$

Solving this problem

- No solution if $\boldsymbol{A}$ has negative eigenvalues! (In that case, the problem is unbounded);
- We will always assume that $\boldsymbol{A} \succeq 0$.

Illustration: Minimization of quadratic functions

Problem

Given a symmetric matrix $\boldsymbol{A} \in \mathbb{R}^{d \times d}$ and a vector $\boldsymbol{b} \in \mathbb{R}^d$, solve

$$\min_{\boldsymbol{w} \in \mathbb{R}^d} \frac{1}{2} \boldsymbol{w}^{\mathrm{T}} \boldsymbol{A} \boldsymbol{w} - \boldsymbol{b}^{\mathrm{T}} \boldsymbol{w}.$$

- Unconstrained problem;
- Smooth: objective is a degree-2 polynomial in the components of $\boldsymbol{w}$.

Solving this problem

- No solution if $\boldsymbol{A}$ has negative eigenvalues! (In that case, the problem is unbounded);
- We will always assume that $\boldsymbol{A} \succeq 0$.

Square root of a matrix matrix

For any symmetric positive semidefinite matrix $\boldsymbol{A}$, there exists a matrix $\boldsymbol{B}$ such that $\boldsymbol{B}^2 = \boldsymbol{B} \times \boldsymbol{B} = \boldsymbol{A}$.

The matrix $\boldsymbol{B}$ is called the square root of $\boldsymbol{A}$; we write $\boldsymbol{B} = \boldsymbol{A}^{1/2}$.

Illustration: Minimization of quadratic functions (2)

Square root of a matrix matrix

For any symmetric positive semidefinite matrix $\boldsymbol{A}$, there exists a matrix $\boldsymbol{B}$ such that $\boldsymbol{B}^2 = \boldsymbol{B} \times \boldsymbol{B} = \boldsymbol{A}$.
The matrix $\boldsymbol{B}$ is called the square root of $\boldsymbol{A}$; we write $\boldsymbol{B} = \boldsymbol{A}^{1/2}$.

Reformulations

The problem $\min_{\boldsymbol{w} \in \mathbb{R}^d} \frac{1}{2} \boldsymbol{w}^{\mathrm{T}} \boldsymbol{A} \boldsymbol{w} - \boldsymbol{b}^{\mathrm{T}} \boldsymbol{w}$ is equivalent to

$$\min_{\boldsymbol{w} \in \mathbb{R}^d} \frac{1}{2} \left\| \boldsymbol{A}^{1/2} \boldsymbol{w} - \boldsymbol{c} \right\|^2,$$

where $\boldsymbol{b} = \boldsymbol{A}^{1/2} \boldsymbol{c}$ ($\boldsymbol{c}$ may not be unique).

A useful example

$$\min_{z \in \mathbb{R}^d} \varphi(z) = \boldsymbol{g}^{\mathrm{T}} \boldsymbol{z} + \frac{m}{2} \|\boldsymbol{z} - \boldsymbol{w}\|^2 \quad \text{where } \boldsymbol{g} \in \mathbb{R}^d, m \geq 0.$$

Illustration: Minimization of quadratic functions (3)

A useful example

$$\min_{\mathbf{z}\in\mathbb{R}^d} \varphi(z) = \mathbf{g}^{\mathrm{T}}\mathbf{z} + \frac{m}{2}\|\mathbf{z}-\mathbf{w}\|^2 \quad \text{where } \mathbf{g}\in\mathbb{R}^d, m \geq 0.$$

Reformulation

1. Expand the least-squares formula:

$$\varphi(z) = \frac{1}{2}z^{\mathrm{T}} \underbrace{(m\mathbf{I}_d)}_{Identity\ \ matrix} \mathbf{z} + (\mathbf{g}-m\mathbf{w})^{\mathrm{T}}\mathbf{z} + \underbrace{\frac{m}{2}\|\mathbf{w}\|^2}_{Independent\ \ of\ \ \mathbf{z}}.$$

A useful example

$$\min_{z \in \mathbb{R}^d} \varphi(z) = \mathbf{g}^{\mathrm{T}} z + \frac{m}{2}\|z - \mathbf{w}\|^2 \quad \text{where } \mathbf{g} \in \mathbb{R}^d, m \geq 0.$$

Reformulation

1. Expand the least-squares formula:

$$\varphi(z) = \frac{1}{2} z^{\mathrm{T}} \underbrace{(m\mathbf{I}_d)}_{\textit{Identity matrix}} z + (\mathbf{g} - m\mathbf{w})^{\mathrm{T}} z + \underbrace{\frac{m}{2}\|\mathbf{w}\|^2}_{\textit{Independent of } z}.$$

2. Equivalent linear least-squares problem:

$$\min_{z \in \mathbb{R}^d} \frac{1}{2}\left\| z - \left(w - \frac{1}{m} g \right) \right\|^2.$$

A useful example

$$\min_{\boldsymbol{z}\in\mathbb{R}^d} \varphi(z) = \boldsymbol{g}^{\mathrm{T}}\boldsymbol{z} + \frac{m}{2}\|\boldsymbol{z}-\boldsymbol{w}\|^2 \quad \text{where } \boldsymbol{g}\in\mathbb{R}^d, m \geq 0.$$

Reformulation

1. Expand the least-squares formula:

$$\varphi(z) = \frac{1}{2}z^{\mathrm{T}} \underbrace{(m\mathbf{I}_d)}_{Identity\ matrix} \boldsymbol{z} + (\boldsymbol{g} - m\boldsymbol{w})^{\mathrm{T}}\boldsymbol{z} + \underbrace{\frac{m}{2}\|\boldsymbol{w}\|^2}_{Independent\ of\ \boldsymbol{z}}.$$

2. Equivalent linear least-squares problem:

$$\min_{\boldsymbol{z}\in\mathbb{R}^d} \frac{1}{2}\left\|\boldsymbol{z} - \left(\boldsymbol{w} - \frac{1}{m}g\right)\right\|^2.$$

3. The global minimum of the problem is $\boldsymbol{z}^* = \boldsymbol{w} - \frac{1}{m}g$.

Illustration: Linear regression

- Data $\{(\boldsymbol{x}_i, y_i)\}_i$, $\boldsymbol{x}_i \in \mathbb{R}^d$, $y_i \in \mathbb{R}$.
- Goal: compute a linear model $h(\boldsymbol{x}) = \boldsymbol{w}^{\mathrm{T}} \boldsymbol{x}$ such that $h(\boldsymbol{x}_i) \approx y_i$ for $i = 1, \dots, n$.
- Objective: Minimize the squares of the errors $|h(\boldsymbol{x}_i) - y_i|$.

Illustration: Linear regression

- Data $\{(\boldsymbol{x}_i, y_i)\}_i$, $\boldsymbol{x}_i \in \mathbb{R}^d$, $y_i \in \mathbb{R}$.
- Goal: compute a linear model $h(\boldsymbol{x}) = \boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}$ such that $h(\boldsymbol{x}_i) \approx y_i$ for $i = 1, \dots, n$.
- Objective: Minimize the squares of the errors $|h(\boldsymbol{x}_i) - y_i|$.

Linear regression

$$\min_{\boldsymbol{w}\in\mathbb{R}^d} \frac{1}{2n} \sum_{i=1}^{n} (h(\boldsymbol{x}_i) - y_i)^2 = \frac{1}{2} \|\boldsymbol{X}\boldsymbol{w} - \boldsymbol{y}\|^2 .$$

- This is a linear least-squares problem!

Illustration: Linear regression (2)

Quality of the least-squares solution

- Best possible approximation in terms of errors;
- Fixed solution when $\{(\boldsymbol{x}_i, y_i)\}_i$ are deterministic.

Illustration: Linear regression (2)

Quality of the least-squares solution

- Best possible approximation in terms of errors;
- Fixed solution when $\{(\mathbf{x}_i, y_i)\}_i$ are deterministic.

In presence of random data

- True linear regression: consider the distribution of the data;
- Statistical interpretation of the least-squares solution: maximum likelihood estimator.

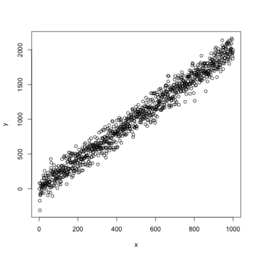
Illustration: Linear regression (2)

Quality of the least-squares solution

- Best possible approximation in terms of errors;
- Fixed solution when $\{(\mathbf{x}_i, y_i)\}_i$ are deterministic.

In presence of random data

- True linear regression: consider the distribution of the data;
- Statistical interpretation of the least-squares solution: maximum likelihood estimator.



In short: linear least squares

Aims

- Find a linear relationship between features and labels in your data;
- Work even when an exact linear model does not exist!

Techniques

- Look for solutions of the associated linear system;
- In practice, can use direct linear algebra solvers (even better when you choose one matching your problem characteristics);
- If too costly, can think of iterative methods.

Outline

Back to the general problem

$$\min_{\boldsymbol{w}\in\mathbb{R}^d} f(\boldsymbol{w}).$$

Assumptions: f smooth, bounded below.

Back to the general problem

$$\min_{\boldsymbol{w} \in \mathbb{R}^d} f(\boldsymbol{w}).$$

Assumptions: f smooth, bounded below.

Key properties

- Smoothness: We will exploit the gradient of f;
- Convexity: Will allow for fast convergence (with the right method).

Example: Logistic regression

Context

- Data set $\{(\mathbf{x}_i, y_i)\}_i$, $\mathbf{x}_i \in \mathbb{R}^d$, $y_i \in \{-1, +1\}$;
- Goal: Classification through a linear classifier $\mathbf{x} \mapsto \mathbf{w}^{\mathrm{T}}\mathbf{x}$;
- **Difference with SVM:** we want probabilities of belonging to a class!

Example: Logistic regression

Context

- Data set $\{(\boldsymbol{x}_i, y_i)\}_i$, $\boldsymbol{x}_i \in \mathbb{R}^d$, $y_i \in \{-1,+1\}$;
- Goal: Classification through a linear classifier $\boldsymbol{x} \mapsto \boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}$;
- **Difference with SVM:** we want probabilities of belonging to a class!

A probabilistic measure

- We define an odds-like function

$$p(\boldsymbol{x};\boldsymbol{w}) = (1 + e^{\boldsymbol{x}^{\mathrm{T}}\boldsymbol{w}})^{-1} \in (0,1).$$

- The parameters $\boldsymbol{w}$ should be chosen such that

$$\begin{cases} p(\boldsymbol{x}_i;\boldsymbol{w}) \approx 1 & \text{if } y_i = +1; \\ p(\boldsymbol{x}_i;\boldsymbol{w}) \approx 0 & \text{if } y_i = -1. \end{cases}$$

Example: Logistic regression (2)

Towards an objective function

$$p(\boldsymbol{x};\boldsymbol{w}) = (1 + e^{\boldsymbol{x}^{\mathrm{T}}\boldsymbol{w}})^{-1},$$

- Penalize cases where
  - $y_i = +1$ and $p(\boldsymbol{x}_i;\boldsymbol{w})$ is small;
  - $y_i = +1$ and $p(\boldsymbol{x}_i;\boldsymbol{w})$ is close to 1;
- Use logarithm of the $p(\boldsymbol{x}_i;\boldsymbol{w})$ in the cost function:
  - Motivation: Statistical interpretation (joint distribution);
  - Mathematical interest for gradient calculations.

Example: Logistic regression (2)

Towards an objective function

$$p(\boldsymbol{x};\boldsymbol{w}) = (1+e^{\boldsymbol{x}^{\mathrm{T}}\boldsymbol{w}})^{-1},$$

- Penalize cases where
  - $y_i = +1$ and $p(\boldsymbol{x}_i;\boldsymbol{w})$ is small;
  - $y_i = +1$ and $p(\boldsymbol{x}_i;\boldsymbol{w})$ is close to 1;
- Use logarithm of the $p(\boldsymbol{x}_i;\boldsymbol{w})$ in the cost function:
  - Motivation: Statistical interpretation (joint distribution);
  - Mathematical interest for gradient calculations.

Resulting function: logistic loss

$$f(\boldsymbol{w}) = \frac{1}{n}\left\{\sum_{y_i=-1}\ln\left(1+e^{-\boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{w}}\right)+\sum_{y_i=+1}\ln\left(1+e^{\boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{w}}\right)\right\}.$$

Ex: Logistic regression (3)

Logistic loss problem

$$\min_{\boldsymbol{w}\in\mathbb{R}^n} \frac{1}{n}\left\{\sum_{y_i=-1}\ln\left(1+e^{-\boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{w}}\right)+\sum_{y_i=+1}\ln\left(1+e^{\boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{w}}\right)\right\}$$

Ex: Logistic regression (3)

Logistic loss problem

$$\min_{\boldsymbol{w} \in \mathbb{R}^n} \frac{1}{n} \left\{ \sum_{y_i=-1} \ln\left(1 + e^{-\boldsymbol{x}_i^{\mathrm{T}} \boldsymbol{w}}\right) + \sum_{y_i=+1} \ln\left(1 + e^{\boldsymbol{x}_i^{\mathrm{T}} \boldsymbol{w}}\right) \right\}$$

- The logistic loss is convex (but not strongly);
- To make it convex, possible to add a regularizing term $\frac{\mu}{2}\|\boldsymbol{w}\|^2$
  $\Rightarrow$ The problem becomes μ-strongly convex!

Example: Nonlinear regression

Context

- Data set $\{(\boldsymbol{x}_i, y_i)\}_i$, $\boldsymbol{x}_i \in \mathbb{R}^d$, $y_i \in \{0,1\}$;
- Goal: Classification through a linear classifier $\boldsymbol{x} \mapsto \boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}$.

Example: Nonlinear regression

Context

- Data set $\{(\mathbf{x}_i, y_i)\}_i$, $\mathbf{x}_i \in \mathbb{R}^d$, $y_i \in \{0, 1\}$;
- Goal: Classification through a linear classifier $\mathbf{x} \mapsto \mathbf{w}^{\mathrm{T}}\mathbf{x}$.

A loss function

- We use a sigmoid function: $\phi(\mathbf{x}_i; \mathbf{w}) = \left(1 + e^{-\mathbf{x}_i^{\mathrm{T}}\mathbf{w}}\right)$;
- Our goal is now to penalize the squared error $(y_i - \phi(\mathbf{x}_i; \mathbf{w}))^2$.

Example: Nonconvex loss function (2)

The optimization problem

$$\min_{\boldsymbol{w} \in \mathbb{R}^d} f(\boldsymbol{w}) = \frac{1}{n} \sum_{i=1}^{n} \left(y_i - \frac{1}{1+e^{-x_i^{\mathrm{T}} w}} \right)^2 .$$

Example: Nonconvex loss function (2)

The optimization problem

$$\min_{\boldsymbol{w} \in \mathbb{R}^d} f(\boldsymbol{w}) = \frac{1}{n} \sum_{i=1}^{n} \left(y_i - \frac{1}{1+e^{-x_i^{\mathrm{T}} w}} \right)^2 .$$

- Nonconvex problem;
- **Nonlinear** least-squares structure;
- Smooth: can apply gradient descent.