OPTIMIZATION FOR MACHINE LEARNING

September 16, 2024

Today:
* Logistics
* Introduction to optimization

Logistics

Course webpage: https://www.lamsade.dauphine.fr/~croyer/teachOID.html

My email: clement.royer@lamsade.dauphine.fr

Schedule:

Sep 16	8.30 - 11.45
Sep 23	8.30 - 11.45
Sep 24	8.30 - 11.45
Sep 30	8.30 - 11.45
Oct 1	17.15 - 20.30
Dec 16	13.45 - 17
Dec 17	17.15 - 20.30
Dec 20	13.45 - 17

Grading

60% Exam (January 29)

40% Homework (done in part in Oct 1st / Dec 17 sessions)

INTRODUCTION TO OPTIMIZATION

Optimization : Field of study concerned with making the best decision out of a set of alternatives

$\Rightarrow$ Matured in the 1980s as an area of computational mathematics

$\Rightarrow$ Success stories : Finance, Supply chain, Scheduling, Vaccine dispatch, ...

↳ Shift over the past 20 years: data-driven optimization

- changed the problems people are interested in
  $\Rightarrow$ Specific formulations
- changed the algorithms people develop and use
  $\Rightarrow$ stochastic gradient methods are the most widely used algorithms in ML
  $\Rightarrow$ other methods that were highly popular in 1980s - 2000s fell out of fashion (Newton's method)

Typical "optimization for machine learning problem"

ML (Machine Learning) : Extract information from data

↳ Data is typically available under the form of samples

$$D = \{ (\underset{\text{input}}{x_i}, \underset{\text{output}}{y_i}) \}_{i=1..m} \qquad m \text{ samples}$$

↳ Goal : Learn a mapping between the x_i's and the y_i's

$\Rightarrow$ We assume that this mapping is defined (or parameterized) using a vector $w \in \mathbb{R}^d$ of "weights" or parameters

$w = \begin{bmatrix} w_1 \\ \vdots \\ w_d \end{bmatrix} d$

$w_j \in \mathbb{R} \ \forall j$

$\Rightarrow$ Notation: $h(\cdot\,; w) : x \mapsto h(x; w)$

Learning problem / Training problem

Find the best parameter values given the data

We represent this problem as follows:

$$\underset{w \in \mathbb{R}^d}{\text{minimize}} \left\{ f_D(w) + \lambda r(w) \right\}$$

Optimization problem

Decision variables

Nature of the problem (minimize / maximize)
Here, want to find the w that yields the lowest value of $f_D(w) + \lambda r(w)$

objective function: characterizes how good a particular choice of w is

Decision variables: Parameters we can choose to affect the objective function

NB: the optimization problem reads "Minimize $f_D(w) + \lambda r(w)$ over $w \in \mathbb{R}^d$"

Specific objective function structure in ML: $f_D(w) + \lambda r(w)$

- $f_D(w)$: training loss / empirical loss / data-fitting term
  $\Rightarrow$ Depends on the data D, measures how good the choice of w is at learning / fitting relationships between

inputs and outputs

$\Rightarrow$ Generic form:

$$D = \{(x_i, y_i)\}_{i=1..m} \qquad f_D(w) = \underbrace{\frac{1}{m}\sum_{i=1}^{m}}_{\text{Average/Mean over all data points}} \overset{\text{Loss function}}{\ell}\Big(\underbrace{h(x_i; w)}_{\substack{\text{Prediction of } y_i \text{ given } x_i \\ \text{made using the mapping} \\ h(\cdot; w) \text{ parametrized by } w}}, y_i\Big)$$

x_i: Input data

y_i: output data

ℓ: loss function quantifies how close $h(x_i; w)$ and y_i are

- $r(w)$: regularization term, penalizes choices of w that do not satisfy desired properties $\Rightarrow$ Not data-dependent!
- $\lambda > 0$: tradeoff parameter
  - $\lambda \gg 1$: more weight on the regularization term, less on data
  - $\lambda \ll 1$: more weight on the data-fitting term, less on regularization

Examples

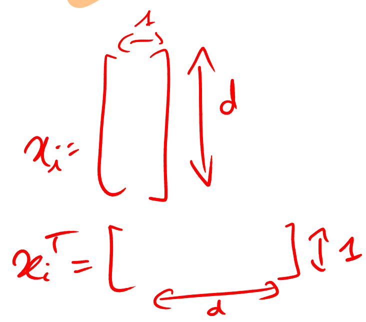
① Linear least squares / Linear regression

Data: $\{(x_i, y_i)\}_{i=1..m}$ $\quad x_i \in \mathbb{R}^d$, $y_i \in \mathbb{R}$

$\Rightarrow$ Representation: $X = \begin{bmatrix} x_1^T \\ \vdots \\ x_m^T \end{bmatrix} \in \mathbb{R}^{m \times d}$ (matrices with m rows and d columns)

$\longleftarrow d \longrightarrow$

$$y = \begin{bmatrix} y_1 \\ | \\ y_m \end{bmatrix} \in \mathbb{R}^m$$



Goal: Find a linear relationship between x_i's and y_i's

Optimization problem:

Norm $\|v\|^2 = \sum_{i=1}^{m} v_i^2 \quad \forall v \in \mathbb{R}^m$

$$\underset{w \in \mathbb{R}^d}{\text{minimize}} \quad \frac{1}{2m} \|Xw - y\|^2$$

where $\|Xw - y\|^2 = \sum_{i=1}^{m} (x_i^T w - y_i)^2$

loss function

Mapping

$x \mapsto x^T w = \sum_{j=1}^{d} x_j w_j$

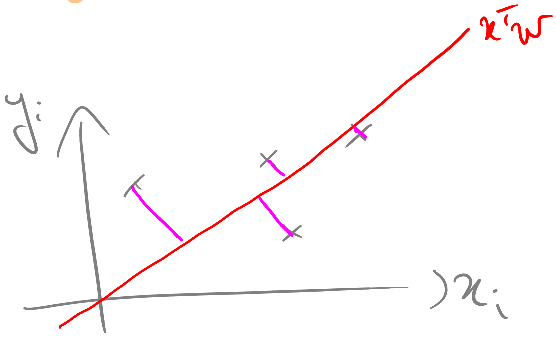
"scalar product" / "inner product"

↳ Fits the general framework with $r(w) = 0 \ \forall w$

and

$$f_D(w) = \frac{1}{m} \sum_{i=1}^{m} \frac{1}{2} (x_i^T w - y_i)^2$$

NB: x^T = "x transpose"



↳ Variants with regularization

* Ridge regression

$$\underset{w \in \mathbb{R}^d}{\text{minimize}} \quad \frac{1}{2m} \|Xw - y\|^2 + \lambda \underbrace{\|w\|^2}_{\ell_2 \text{ regularization}} \qquad \lambda > 0$$

$\|w\|^2 = \sum_{j=1}^{d} w_j^2$

ℓ_2 regularization (reduces overfitting)

* LASSO

$$\underset{w \in \mathbb{R}^d}{\text{minimize}} \quad \frac{1}{2m} \|Xw - y\|^2 + \lambda \|w\|_1$$

ℓ_1 regularization

$\|w\|_1 = \sum_{j=1}^{d} |w_j|$

→ Produces solutions (w) with many zero coefficients

$\Rightarrow$ Sparse solutions / Feature selection

② Support vector machines (SVM)

<u>Data</u>: $\{(x_i, y_i)\}_{i=1..n}$ $\quad x_i \in \mathbb{R}^d$ $\quad y_i \in \{-1, +1\}$

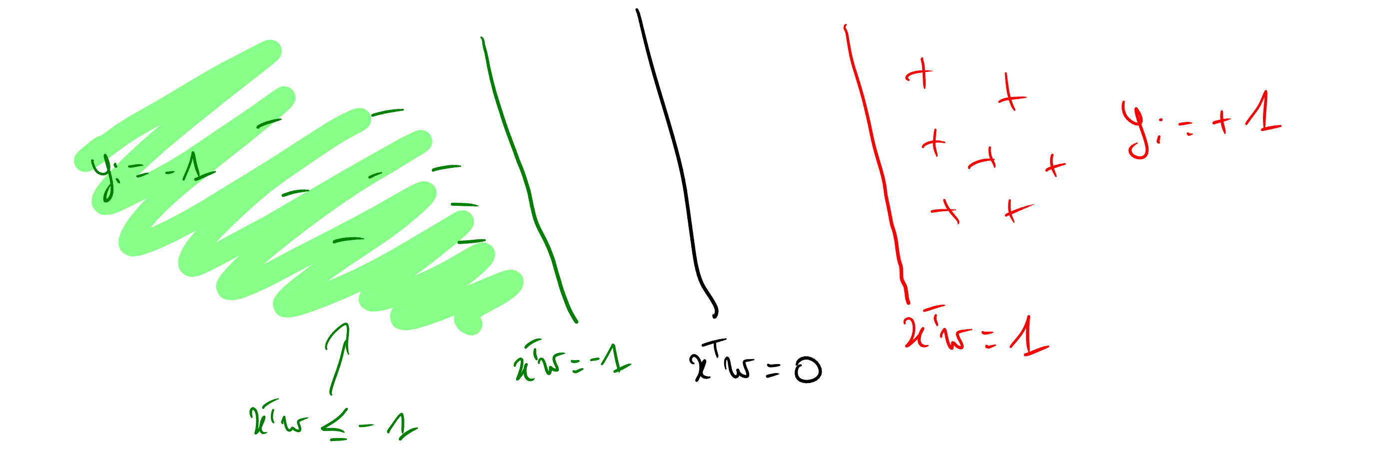
Binary label

<u>Goal</u> (classification): Compute y_i given x_i through a linear model $x \longmapsto x^T w$

<u>Idea</u>: For (x_i, y_i), we should have

$$x_i^T w \geq 1 \quad \text{if} \quad y_i = 1$$

and $\quad x_i^T w \leq -1 \quad \text{if} \quad y_i = -1$



Optimization problem

$$\underset{w \in \mathbb{R}^d}{\text{minimize}} \ \frac{1}{n}\left\{\sum_{i=1}^{n} \max(1 - y_i\, x_i^T w, 0)\right\} + \lambda \|w\|_1$$

hinge loss

ℓ_1 regularization for robustness to outliers

<u>Extensions</u>

* Use ℓ_2 regularization instead of ℓ_1
* Use a nonlinear function of x (kernel SVM)

(3) Deep neural networks

Illustration: Feed-forward NN, multiclass classification

<u>Data</u>: $\{(x_i, y_i)\}_{i=1..n}$ $\qquad x_i \in \mathbb{R}^{d_0}$

$y_i \in \mathbb{R}^K \qquad y_i = [y_{ik}]_{k=1..K}$

K classes, y_i: 1-hot encoding of the class of x_i $\qquad y_i = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$ $\to$ class of x_i

<u>Goal</u>: Classification (between K classes)

<u>Mapping</u>: Neural network with L layers

$$x^0 \in \mathbb{R}^{d_0} \mapsto x^1 \in \mathbb{R}^{d_1} \mapsto \dots \mapsto x^L \in \mathbb{R}^{d_L}$$

$$\forall \ell = 1..L, \qquad x^\ell = \sigma\Big(\overbrace{W^\ell x^{\ell-1} + b^\ell}^{\text{linear transformation of } x^{\ell-1}}\Big)$$

weight matrix in $\mathbb{R}^{d_\ell \times d_{\ell-1}}$ $\qquad$ bias vector in $\mathbb{R}^{d_\ell}$

σ: nonlinear activation applied component-wise

Ex) ReLU, $\quad \sigma(x^{\ell-1}) = \begin{bmatrix} \max(x^{\ell-1}_1, 0) \\ \vdots \\ \max(x^{\ell-1}_{d_{\ell-1}}, 0) \end{bmatrix} \in \mathbb{R}^{d_{\ell-1}}$

Given one input x_i as x^0, the NN outputs $x^L \in \mathbb{R}^{d_L}$ with $d_L = K$ (number of classes)

↳ the parameters of the network are the weight matrices (W^ℓ) and the bias vectors (b^ℓ)

$\Rightarrow$ The parameters are gathered in a vector $w \in \mathbb{R}^d$

with $d = \underbrace{d_1}_{b^1} + \underbrace{d_1 \times d_0}_{W^1} + \dots + \underbrace{d_L}_{b^L} + \underbrace{d_L \times d_{L-1}}_{W^L}$

Optimization problem

$\rightarrow$ Based on the negative log likelihood as a loss function : minimizing this loss is equivalent to maximizing the probability of correct classification

$\rightarrow$ Given x_i, the output of the NN is denoted by $x^L(x_i; w)$

k-th component of $x^L(x_i;w) \in \mathbb{R}^K$

$$\underset{w \in \mathbb{R}^d}{\text{minimize}} \ \frac{1}{m} \sum_{i=1}^{m} \left\{ \log\left(\sum_{k=1}^{K} \exp\left([x^L(x_i;w)]_k \right) \right) - \sum_{k=1}^{K} y_{ik} [x^L(x_i;w)]_k \right\} + \underbrace{\lambda \|w\|^2}_{\text{For generalization}}$$

Course goals

- Identify structure within an optimization problem
- choose an algorithm that is well-suited for that problem

NB: For homework assignment, we'll look at shallow neural networks 2 layers

$$x \longmapsto W^2 \sigma(W^1 x + b^1) + b^2$$

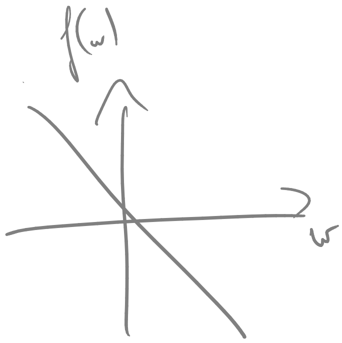
BASICS OF OPTIMIZATION

We consider a general optimization problem

$$(P) \quad \underset{w \in \mathbb{R}^d}{\text{minimize}} \ f(w) \qquad f : \mathbb{R}^d \to \mathbb{R}$$

<u>Definitions</u>:

- $\min_{w \in \mathbb{R}^d} f(w)$: optimal value of (P)
  best value of f (smallest) obtained among all w in $\mathbb{R}^d$



In general, $\min_{w \in \mathbb{R}^d} f(w) \in \mathbb{R} \cup \{-\infty, +\infty\}$

$\min_{w \in \mathbb{R}^d} f(w) = -\infty$: (P) is unbounded

$\min_{w \in \mathbb{R}^d} f(w) = +\infty$: (P) is infeasible (only occurs for special fs)

- $\underset{w \in \mathbb{R}^d}{\operatorname{argmin}} f(w) \subseteq \mathbb{R}^d$: set of solutions of (P)

$$\underset{w \in \mathbb{R}^d}{\operatorname{argmin}} f(w) = \left\{ \bar{w} \in \mathbb{R}^d \;\middle|\; f(\bar{w}) = \min_{w \in \mathbb{R}^d} f(w) \right\}$$

$\hookrightarrow$ this set can be empty (No solutions), finite (1 or more solutions, but finitely many) or infinite

Ex) $f : w \mapsto 0$ $\qquad \underset{w \in \mathbb{R}^d}{\operatorname{argmin}} f(w) = \mathbb{R}^d$

Properties: $\forall \alpha > 0$,

$$\min_{w\in\mathbb{R}^d}\{\alpha \times f(w)\} = \alpha \times \min_{w\in\mathbb{R}^d} f(w)$$

$$\operatorname{argmin}_{w\in\mathbb{R}^d}\{\alpha f(w)\} = \operatorname{argmin}_{w\in\mathbb{R}^d} f(w)$$

$\Rightarrow$ We say that (P) is equivalent to $\underset{w\in\mathbb{R}^d}{\text{minimize}} \{\alpha f(w)\}$ for any $\alpha > 0$

Ex) $\underset{w\in\mathbb{R}^d}{\text{minimize}} \frac{1}{n}\sum_{i=1}^{n} \dots$ equivalent to $\underset{w\in\mathbb{R}^d}{\text{minimize}} \sum_{i=1}^{n} \dots$

⚠ In general, it is intractable to compute the set of solutions directly $\Rightarrow$ we need to use algorithms to compute approximate solutions in finite time

Def:

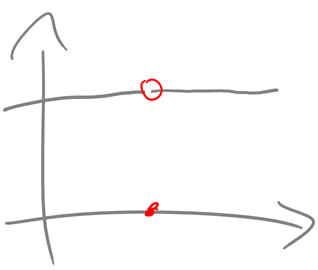
- A solution $\bar{w} \in \operatorname{argmin}_{w\in\mathbb{R}^d} f(w)$ is called a global minimum

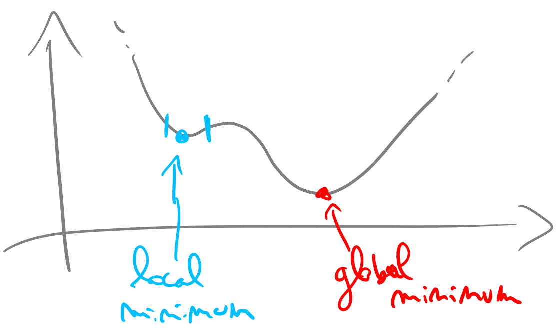
$$\forall w \in \mathbb{R}^d, \quad f(\bar{w}) \leq f(w)$$

- A vector $\bar{w} \in \mathbb{R}^d$ is called a local minimum if

$\forall w \in \mathbb{R}^d$ such that $\|w - \bar{w}\|$ is small enough,

$$f(\bar{w}) \leq f(w)$$





NB

Global

$\Downarrow$

Local

↳ Most optimization algorithms in this course are designed to find local minima

↳ But in general, finding local minima is not tractable ⟹ Need more assumptions on f

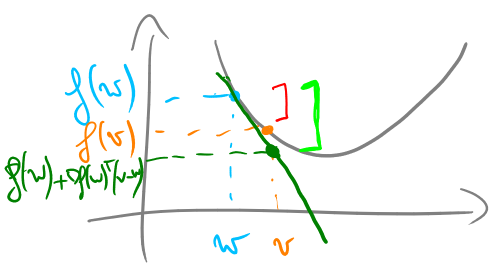
↳ With differentiability and convexity, we can
- characterize local minima (or approximations thereof)
- in finite time / in an easy way

a) Differentiability

Def: $f: \mathbb{R}^d \to \mathbb{R}$ is C^1 if its derivative exists for any $w \in \mathbb{R}^d$ and the derivative is a continuous function

⟹ $\forall w \in \mathbb{R}^d$, there exists a vector called the gradient of f at w, denoted by $\nabla f(w) \in \mathbb{R}^d$ that represents the derivative of f at w

Fundamental property: If f is C^1 and $w \in \mathbb{R}^d$, then for any $v \in \mathbb{R}^d$ such that $\|v - w\|$ is small enough, then



$$f(v) - f(w) \underset{\text{"approximates"}}{\approx} \underbrace{\nabla f(w)}_{\in \mathbb{R}^d}{}^T \underbrace{(v - w)}_{\in \mathbb{R}^d}$$

↳ Computing $f(v) - f(w)$ involves computing f

↳ Computing $\nabla f(w)^T(v-w)$ $\underline{\underline{\forall v}}$ involves only 1 calculation of $\nabla f(w)$

→ The fact that $f(v) \simeq f(w) + \nabla f(w)^T(v-w)$ for small $\|v-w\|$ is a local property that has connections to local optimality

Theorem : [First-order necessary conditions]

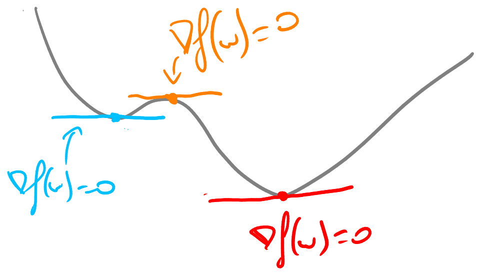
Let $f : \mathbb{R}^d \to \mathbb{R}$ be C^1.

$\underline{\text{If}}$ $\bar{w} \in \mathbb{R}^d$ is a local minimum of f,

$\underline{\text{Then}}$ $\nabla f(\bar{w}) = 0$.

$$\nabla f(\bar{w}) = \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix}$$

⚠



* There can be $\bar{w} \in \mathbb{R}^d$ that are not local minima but for which $\nabla f(\bar{w}) = 0$ (Ex : local maxima)
* This is why the condition is called $\underline{\text{necessary}}$ (IF ... THEN ...)

↳ Even if the condition is only necessary, it gives a way to check whether a vector is not a local minimum in finite time (time required to check $\nabla f(w) = 0$)

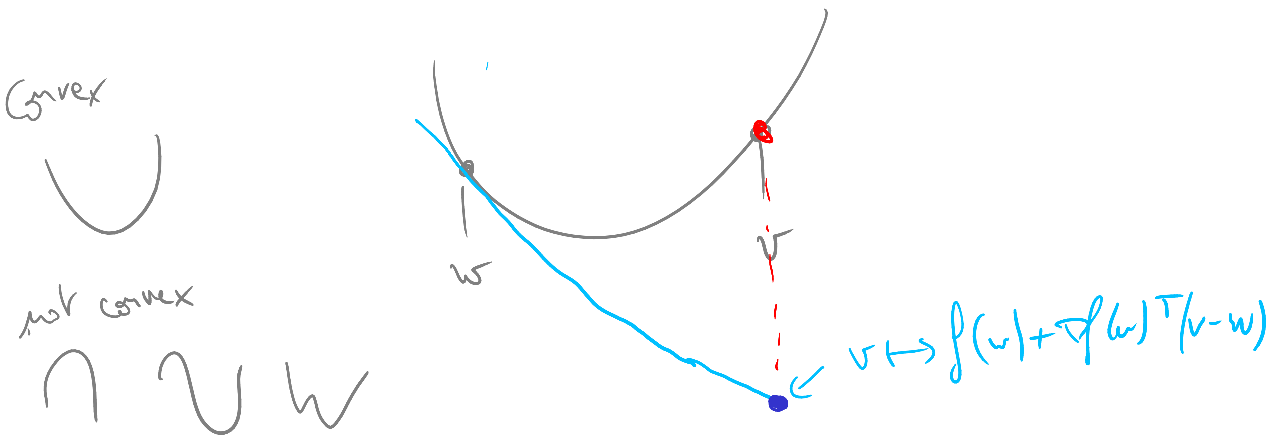
IF local minimum THEN $\nabla f(w) = 0$

IF $\nabla f(w) \neq 0$ THEN NOT local minimum

b) Convexity

Def: $f:\mathbb{R}^d \to \mathbb{R}$ C^1 is convex if

$$\forall (w,v)\in(\mathbb{R}^d)^2, \quad f(v) \geq f(w) + \nabla f(w)^T (v-w)$$



Theorem: Let f be C^1 convex. Then

(i) $[\bar{w} \in \mathbb{R}^d \text{ local minimum}] \iff [\nabla f(\bar{w}) = 0]$

(ii) $[\bar{w} \in \mathbb{R}^d \text{ local minimum}] \iff [\bar{w} \in \mathbb{R}^d \text{ global minimum}]$

Takeaway:

- Finding global minima is intractable in practice...
- ...but not for C^1 and convex functions (like the objective function of linear least squares). For those functions, checking the gradient is enough to determine whether a point is a global minimum