Optimization for Machine Learning

M2 IDA, September 21, 2023

Today

Lecture (8.30am - 10am)
- Convex optimization problems
- Gradient descent (time permitting)

Exercises (10.15am - 11.45am, F. le Bronnec)
- Manipulate notions of convexity and optimality
- Solutions available online afterwards

Next session (Monday, Sep 25)
- 8.30am - 10am: Exercises on convex optimization problems
- 10.15am - 11.45am: Lecture on gradient descent

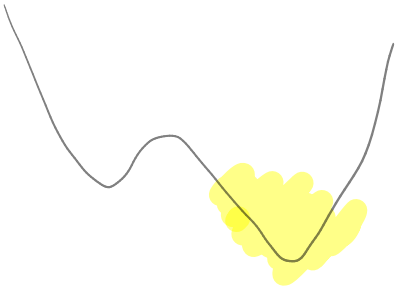
1) MORE ABOUT CONVEX OPTIMIZATION PROBLEMS

(Chapter 2 of lecture notes)

↳ Many instances of ML problems are formulated as convex optimization problems

⇒ Every local solution is a global solution (and solution here means minimum)

⇒ Can find those solutions by solving $\nabla f(w) = 0$ where f is the function to minimize ($f: \mathbb{R}^d \to \mathbb{R}$, C^1)



↳ There exist very efficient algorithms for solving convex problems (even with constraints, even without C^1 objective, ...)

⇒ In convex optimization, the challenge is about modeling your problem as a nice convex optimization problem

Ex) linearize everything ⇒ LP relaxation

LP ≡ linear program

Problem data

(LP)

$$c = \begin{bmatrix} c_1 \\ \vdots \\ c_d \end{bmatrix} \in \mathbb{R}^d$$

$A \in \mathbb{R}^{m \times d}$

$b \in \mathbb{R}^m$

minimize $c^T w$ s.t. $Aw = b$, $w \geq 0$

$w \in \mathbb{R}^d$

linear objective function

linear constraints

$w_i \geq 0$

$$f(w) = c^T w = \sum_{i=1}^{d} c_i w_i$$

Constraint set: $\{ w \in \mathbb{R}^d \mid Aw = b, w \geq 0 \}$

(*) $Aw = b$ ($A \in \mathbb{R}^{m \times d}$, $w \in \mathbb{R}^d$, $b \in \mathbb{R}^m$): linear system of m equations in d variables

If $A = [A_{ij}]_{\substack{1 \leq i \leq m \\ 1 \leq j \leq d}}$, $b = [b_i]_{1 \leq i \leq m}$, $w = [w_j]_{1 \leq j \leq d}$

(*) $\Leftrightarrow$

$$\begin{cases} A_{11} w_1 + \dots + A_{1d} w_d = b_1 \\ \vdots \\ A_{m1} w_1 + \dots + A_{md} w_d = b_m \end{cases}$$

NB. We write (LP) in so-called standard form but there are other possible formulations,

minimize $c^T w$ s.t. $w \geq 0$ / minimize $c^T w$ s.t. $Aw \leq b$

$w \in \mathbb{R}^d$ / $w \in \mathbb{R}^d$

↳ Not every optimization problem can be written as a linear program!

But

Relaxation
- often possible to write an LP that approximates the problem of interest

Reformulation
- Some nonlinear optimization problems can be reformulated as LPs!

Relaxation: $w \mapsto (w+1)^2$ from $\mathbb{R}$ to $\mathbb{R}$

locally, $(w+1)^2 \simeq 0 + 2(w+1)$

⇒ Idea behind gradient descent

⇒ Used in many applications, e.g. Optimal Power Flow (power distribution in a network)

↳ Nonlinear problem

↳ Approximated by a linear program using a solver and the current data (demand/supply/...) updated every 15 minutes

Reformulation: Replace the problem by an equivalent linear program

Example: Robust linear regression

1) Setup: Data $\{(x_i, y_i)\}_{i=1..m}$

$x_i \in \mathbb{R}^d$ $\quad y_i \in \mathbb{R}$

We define

$$X = \begin{bmatrix} x_1^T \\ \vdots \\ x_m^T \end{bmatrix} \in \mathbb{R}^{m \times d}$$ ("feature matrix" / input)

$$y = \begin{bmatrix} y_1 \\ \vdots \\ y_m \end{bmatrix} \in \mathbb{R}^m$$ ("label vector" / output)

Goal: Find $w \in \mathbb{R}^d$ such that Xw is close to y

Ideally, would like $Xw = y$ (perfect fit to the data) but when the data is noisy this is not possible

- One possibility is to compute w by solving a linear least-squares problem:

$$\underset{w \in \mathbb{R}^d}{\text{minimize}} \ \frac{1}{2m} \|Xw - y\|^2 = \frac{1}{2m} \sum_{i=1}^{m} (x_i^T w - y_i)^2$$

(average over data points)

Called "l_2 regression", "linear regression", "ordinary least squares", ...

(Having $\frac{1}{2}, \frac{1}{2m}, 1$ in front of the sum does not

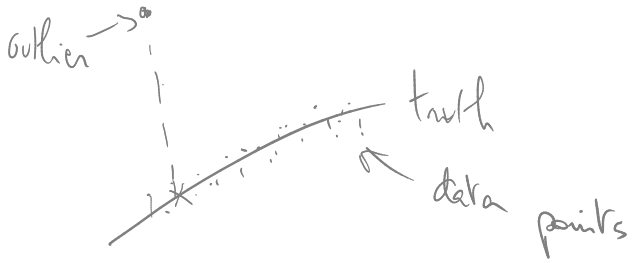
Change the set of solutions

$\forall \lambda > 0$, $\underset{w \in \mathbb{R}^d}{\operatorname{argmin}} f(w) = \underset{w \in \mathbb{R}^d}{\operatorname{argmin}} \lambda \times f(w)$

$\Rightarrow$ Multiplying by $\lambda > 0$ is one way to build a problem reformulation!

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

↳ The solution of the least-squares problems is known to be sensitive to outliers in the data



$y = \underbrace{X w^*}_{\text{truth}} + \underbrace{\varepsilon}_{\text{noise}}$

truth + noise

↳ One way to tackle this issue is to use another norm $\Rightarrow$ ℓ_1 linear regression

$$\underset{w \in \mathbb{R}^d}{\text{minimize}} \; \|Xw - y\|_1 = \sum_{i=1}^{n} |x_i^T w - y_i|$$

Nonlinear function
Not C^1
Convex

One way to solve this $\Rightarrow$ Reformulation as an LP!

(1) $\underset{w \in \mathbb{R}^d}{\text{minimize}} \; \|Xw - y\|_1 = \sum_{i=1}^{n} |x_i^T w - y_i|$

$|t| = \begin{cases} t & \text{if } t \geq 0 \\ -t & \text{if } t \leq 0 \end{cases}$

$\hookrightarrow |x_i^T w - y_i| = t_i^+ + t_i^-$

$x_i^T w - y_i = t_i^+ - t_i^-$

$t_i^+ \geq 0$

$t_i^- \geq 0$

is equivalent to the linear program

(2)

minimize $\begin{bmatrix} w \\ t^+ \\ t^- \end{bmatrix} \in \mathbb{R}^{d+2n}$

$\sum_{i=1}^{n} (t_i^+ + t_i^-)$ → linear function of w, t_i^+, t_i^-

s.t.

$x_i^T w - y_i = t_i^+ - t_i^- \quad \forall i = 1..n$

$t_i^+ \geq 0 \quad \forall i = 1..n$

$t_i^- \geq 0 \quad \forall i = 1..n$

linear constraints with respect to w, t^+, t^-

$\forall a \in \mathbb{R}$

$a = a^+ - a^-$

$|a| = a^+ + a^-$

$a^+ = \max(a, 0)$

$a^- = \max(-a, 0)$

(2) is an LP ⇒ can be solved efficiently

Get solution $\begin{bmatrix} w^* \\ (t^+)^* \\ (t^-)^* \end{bmatrix}$

and w^* will be a solution of (1)