Introduction to Machine Learning Course

Short HW4 – Optimization, Regression, and Boosting

Submitted individually by Thursday, 12.01.23, at 23:59.

You may answer in Hebrew or English and write on a computer or by hand (but be clear).

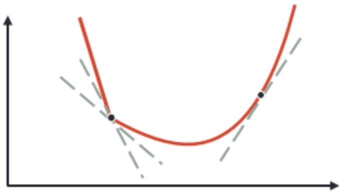
Please submit a PDF file named like your ID number, e.g., 123456789.pdf.

Bonus (maximal grade is 100): Writing on a computer (using LyX/LaTeX, Word + Equation tool, etc.) = 2 pts.

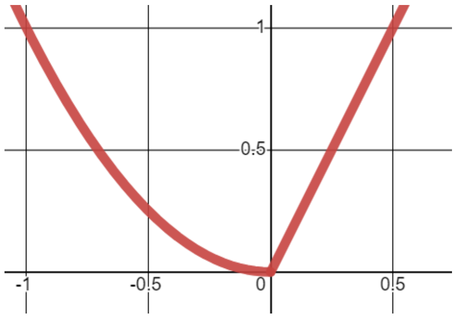
Part A – Optimization

As we saw in Tutorial 08, subgradients generalize gradients to convex functions which are not necessarily differentiable. Notice: you can solve this exercise even before watching Tutorial 08.

Definition: the set of subgradients of $f: V \to \mathbb{R}$ at point $\boldsymbol{u} \in V$ is:



$$\partial f(\boldsymbol{u}) \triangleq \{\boldsymbol{q} \in V \mid \forall \boldsymbol{v} \in V: f(\boldsymbol{v}) \geq f(\boldsymbol{u}) + \boldsymbol{q}^\top(\boldsymbol{v} - \boldsymbol{u})\}.$$



1. Let $f(x) = \begin{cases} x^2, & x < 0 \\ 2x, & x \geq 0 \end{cases}$.
   1.1. Is f convex? No need to explain.
   1.2. Propose a sub-derivative function g for f. That is, $g \in \partial f$.
   Use the above definition to prove that $g(u) \in \partial f(u), \forall u \in \mathbb{R}$.
   1.3. Set a learning rate of $\eta = 0.25$ and a starting point $x_0 = -1$.
   Running subgradient descent, will the algorithm converge to a minimum?
   Prove you answer by filling the following table like we did in Tutorial 07 using as many rows as needed.

i	x_i	$f(x_i)$	$\frac{\partial}{\partial x} f(x_i) = g(x_i)$
0	-1	1	
1			
$\vdots$			

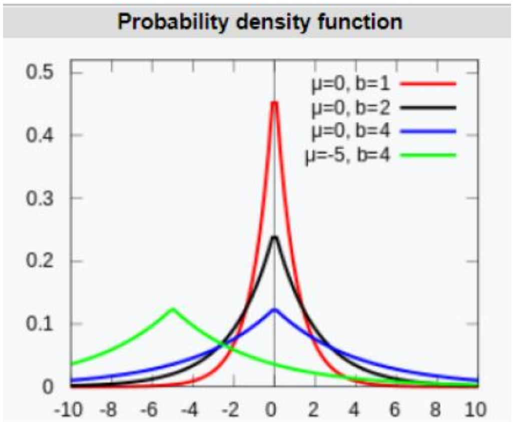
   1.4. Repeat 1.3 with $\eta = 1,\ x_0 = -1$.

Part B – Regression

2. Consider a noisy linear model where $y = \langle \boldsymbol{w}, x \rangle + \varepsilon$, for:
    - Given examples $\boldsymbol{x} \in \mathbb{R}^d$
    - An unknown weight vector $\boldsymbol{w} \in \mathbb{R}^d$
    - Random i.i.d noise ε

In Lecture 09, we showed that when $\varepsilon \sim \mathcal{N}(0, \sigma^2)$, the solution of the least squares formulation is a Maximum-Likelihood Estimator (MLE) of the unknown $\boldsymbol{w}$.

Prove that when $\varepsilon_i \sim \text{Laplace}(0, b)$, the MLE for $\boldsymbol{w}$ corresponds to the solution of the least absolute deviation problem:



$$\text{argmin}_{\boldsymbol{w}} \underbrace{\frac{1}{m}\sum_{i=1}^{m} |\boldsymbol{w}^\top \boldsymbol{x}_i - y_i|}_{\mathcal{L}_{\text{abs}}(\boldsymbol{w})}.$$

That is, prove that:

$$\underbrace{\text{argmax}_{\boldsymbol{w}} \prod_{i=1}^{m} P(y_i, \boldsymbol{x}_i; \boldsymbol{w})}_{\text{Maximum-Likelihood Estimator}} = \underbrace{\text{argmin}_{\boldsymbol{w}} \frac{1}{m}\sum_{i=1}^{m} |\boldsymbol{w}^\top \boldsymbol{x}_i - y_i|}_{\text{Least absolute deviation}}.$$

The steps in your proof should be briefly explained.

Reminder: The Laplacian pdf's is $p(y_i|\mu, b) = \frac{1}{2b} \exp\left\{-\frac{|y_i - \mu|}{b}\right\}$.

Its statistics are $\mathbb{E}[y_i] = \mu = \boldsymbol{w}^\top \boldsymbol{x}_i$ and $\text{Var}[y_i] = 2b^2$.

Part C – Boosting

3. Given the following data with binary labels ("+", "-").

We run AdaBoost with Decision stumps as weak classifiers.
The sizes of the shapes in the figures indicate the probabilities that the algorithm assigns to each sample (high probability = large shape). Initially, the algorithm starts from a uniform distribution.

Only one of the following figures depicts a possible distribution that can be obtained after <u>one</u> iteration of AdaBoost. **Which one?** Answer and propose a weak classifier that can lead to that figure (use a <u>clear</u> drawing or a short description of that classifier).