MATH3431 Machine Learning and Neural Networks III — Epiphany term 2023

Homework 1: Stochastic learning: Stochastic Gradient Descent

Lecturer: Georgios P. Karagiannis — georgios.karagiannis@durham.ac.uk

As formative assessment, submit the solutions to Exercise 1.2, 1.3, and 1.4.

Exercise 1. ($\star\star\star$) Consider the binary classification problem with inputs $x \in \mathcal{X}$ where $\mathcal{X} := \left\{x \in \mathbb{R}^d : \|x\|_2 \leq L\right\}$ for some given value $L > 0$, target $y \in \mathcal{Y}$ where $\mathcal{Y} := \{-1, +1\}$, and prediction rule $h_w : \mathbb{R}^d \to \{-1, +1\}$ with

$$\begin{aligned} h_w(x) &= \text{sign}\left(w^\top x\right) && (1) \\ &= \text{sign}\left(\sum_{j=1}^{d} w_j x_j\right) && (2) \end{aligned}$$

Let the hypothesis class of prediction rules be

$$\mathcal{H} = \left\{x \to w^\top x : \forall w \in \mathbb{R}^d\right\}$$

In other words, the hypothesis $h_w \in \mathcal{H}$ is parametrized by $w \in \mathbb{R}^d$ it receives an input vector $x \in \mathcal{X} := \mathbb{R}^d$ and it returns the label $y = \text{sign}\left(w^\top x\right) \in \mathcal{Y} := \{\pm 1\}$.

Consider a loss function $\ell : \mathbb{R}^d \to \mathbb{R}_+$ with

$$\ell(w, z = (x, y)) = \max\left(0, 1 - y w^\top x\right) + \lambda \|w\|_2^2 \quad (3)$$

for some given value $\lambda > 0$.

Assume there is available a dataset of examples $S_n = \{z_i = (x_i, y_i)\,;\, i = 1, ..., n\}$ of size n.

Do the following tasks.

Hint-1: We denote

$$\text{sign}(\xi) = \begin{cases} -1, & \text{if } \xi < 0 \\ +1, & \text{if } \xi > 0 \end{cases}$$

Hint-2: The notation ±1 means either -1 or $+1$.

HInt-3: We define $\mathbb{R}_+ := (0, +\infty)$

Hint-4: We denote $\|x\|_2 := \sqrt{\sum_{\forall j} (x_j)^2}$ the Euclidean distance.

1. Show that the function $f : \mathbb{R} \to \mathbb{R}_+$ with $f(x) = \max(0, 1 - x)$ is convex in $\mathbb{R}$; and show that the loss (3) is convex.

   Hint: You may use Example 13 from Handout 1.

2. Show that the loss $\ell(w, z)$ for $\lambda = 0$ (3) is L-Lipschitz (with respect to w) when $x \in \mathcal{X}$ where $\mathcal{X} := \left\{x \in \mathbb{R}^d : \|x\|_2 \leq L\right\}$.

   Hint: You may use the definition of Lipschitz function. Without loss of generality, you can consider any $w_1 \in \mathbb{R}^d$ and $w_2 \in \mathbb{R}^d$ such that $1 - yw_2^\top x \leq 1 - yw_1^\top x$, and then take cases $1 - yw_2^\top x\ >$ or $<\ 0$ and $1 - yw_1^\top x >$ or < 0 to deal with the max.

3. Construct the set of sub-gradients $\partial f(x)$ for $x \in \mathbb{R}$ of the function $f : \mathbb{R} \to \mathbb{R}_+$ with $f(x) = \max(0, 1 - x)$. Show that the vector v with

$$v = \begin{cases} 2\lambda w, & yw^\top x > 1 \\ 2\lambda w, & yw^\top x = 1 \\ -yx + 2\lambda w, & yw^\top x < 1 \end{cases}$$

   is $v \in \partial_w \ell(w, z = (x, y))$, aka a sub-gradient of $\ell(w, z = (x, y))$ at w, for any $w \in \mathbb{R}^d$.

4. Write down the algorithm of online AdaGrad (Adaptive Stochastic Gradient Descent) with learning rate $\eta_t > 0$, batch size m, and termination criterion $t > T_{\max}$ for some $T_{\max} > 0$ in order to discover w^* such as

$$w^* = \arg \min_{\forall w : h_w \in \mathcal{H}} \left(\mathrm{E}_{z \sim g}\left(\ell\left(w, z = (x, y)\right)\right)\right) \tag{4}$$

   The formulas in your algorithm have to be tailored to 3.

5. Use the R code given below in order to generate the dataset of observed examples $S_n = \{z_i = (x_i, y_i)\}_{i=1}^n$ that contains $n = 10^6$ examples with inputs x of dimension $d = 2$. Consider $\lambda = 0$. Use a seed $w^{(0)} = (0, 0)^\top$.

   (a) By using appropriate values for m, η_t and $T_{\max}$, code in R the algorithm you designed in part 4, and run it.

   (b) Plot the trace plots for each of the dimensions of the generated chain $\left\{w^{(t)}\right\}$ against the iteration t.

(c) Report the value of the output w^*_{adaGrad} (any type) of the algorithm as the solution to (4).

(d) To which cluster y (i.e., -1 or 1) $x_{\text{new}} = (1, 0)^\top$ belongs?

```
# R code. Run it before you run anything else
#
data_generating_model <- function(n,w) {
z <- rep( NaN, times=n*3 )
z <- matrix(z, nrow = n, ncol = 3)
z[,1] <- rep(1,times=n)
z[,2] <- runif(n, min = -10, max = 10)
p <- w[1]*z[,1] + w[2]*z[,2] p <- exp(p) / (1+exp(p))
z[,3] <- rbinom(n, size = 1, prob = p)
ind <- (z[,3]==0)
z[ind,3] <- -1
x <- z[,1:2]
y <- z[,3]
return(list(z=z, x=x, y=y))
}
n_obs <- 1000000
w_true <- c(-3,4)
set.seed(2023)
out <- data_generating_model(n = n_obs, w = w_true)
set.seed(0)
z_obs <- out$z #z=(x,y)
x <- out$x
y <- out$y
#z_obs2=z_obs
#z_obs2[z_obs[,3]==-1,3]=0
#w_true <- as.numeric(glm(z_obs2[,3]~ 1+ z_obs2[,2],family = "binomial"
)$coefficients)
```