MATH3431 Machine Learning and Neural Networks III — Epiphany term 2025

Exercise sheet

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

Part 1. Convex learning problems

Exercise 1. ($\star$)Let $f:\mathbb{R}^d\rightarrow\mathbb{R}$ such that $f(w)=g(<w,x>+y)$ or some $x\in\mathbb{R}^d$, $y\in\mathbb{R}$. Show that: If g is convex function then $f$is convex function.

Exercise 2. ($\star$)Let functions g_1 be ρ_1-Lipschitz and g_2 be ρ_2-Lipschitz. Then, show that, f with $f(x)=g_1(g_2(x))$ is $\rho_1\rho_2$-Lipschitz.

Exercise 3. ($\star$)Let $f:\mathbb{R}^d\rightarrow\mathbb{R}$ with $f(w)=g(\langle w,x\rangle+y)$ $x\in\mathbb{R}^d$ and $y\in\mathbb{R}$. Let $g:\mathbb{R}\rightarrow\mathbb{R}$ be a β-smooth function. Then show that f is a $\left(\beta\left\Vert x\right\Vert^2\right)$-smooth.

Hint:: You may use Cauchy-Schwarz inequality $\langle y,x\rangle\leq\left\Vert y\right\Vert\left\Vert x\right\Vert$

Exercise 4. ($\star$)Show that $f:S\rightarrow\mathbb{R}$ is ρ-Lipschitz over an open convex set S if and only if for all $w\in S$ and $v\in\partial f(w)$ it is $\left\Vert v\right\Vert\leq\rho$.

Hint:: You may use Cauchy-Schwarz inequality $\langle y,x\rangle\leq\left\Vert y\right\Vert\left\Vert x\right\Vert$

Exercise 5. ($\star$)Let $g_1(w),...,g_r(w)$ be r convex functions, and let $f(\cdot)=\max_{\forall j}(g_j(\cdot))$. Show that for some w it is $\nabla g_k(w)\in\partial f(w)$ where $k=\arg\max_j(g_j(w))$ is the index of function $g_j(\cdot)$ presenting the greatest value at w.

Exercise 6. ($\star$)Consider the regression learning problem $(\mathcal{H},\mathcal{Z},\ell)$ with predictor rule $h(x)=\langle w,x\rangle$ labeled by some unknown parameter $w\in\mathcal{W}$, loss function $\ell(w,(x,y))=(\langle w,x\rangle-y)^2$, feature $x\in\mathcal{X}$, and target $y\in\mathbb{R}$. Let $\mathcal{W}=\mathcal{X}=\left\{\omega\in\mathbb{R}^d:|\omega|\leq\rho\right\}$ for some $\rho>0$.

(1) Show that the resulting learning problem is Convex-Lipschitz-Bounded learning problem.
(2) Specify the parameters of Lipschitnzess.

Exercise 7. ($\star$)If $f$is λ-strongly convex and u is a minimizer of f then for any w

$$f(w) - f(u) \geq \frac{\lambda}{2} \|w-u\|^2$$

Hint:: Use the definition, and set $\alpha \to 0$.

Exercise 8. ($\star$) Let $f : \mathbb{R}^d \to \mathbb{R}$ be a convex and β-smooth function.

(1) Show that for $v, w \in \mathbb{R}^d$

$$f(v) - f(w) \in \left(\langle \nabla f(w), v-w \rangle, \langle \nabla f(w), v-w \rangle + \frac{\beta}{2} \|v-w\|^2 \right)$$

(2) Show that for $v, w \in \mathbb{R}^d$ such that $v = w - \frac{1}{\beta}\nabla f(w)$, it is

$$\frac{1}{2\beta} \|\nabla f(w)\|^2 \leq f(w) - f(v)$$

(3) Additionally assume that $f(x) > 0$ for all $x \in \mathbb{R}^d$. Show that for $w \in \mathbb{R}^d$,

$$\|\nabla f(w)\| \leq \sqrt{2\beta f(w)}$$

Exercise 9. ($\star$)Let $f : \mathbb{R}^d \to \mathbb{R}$ be a λ-strongly convex function. Assume that w^* is a minimizer of f i.e.

$$w^* = \arg\min_w \{f(w)\}$$

Show that for any $w \in \mathbb{R}^d$ it holds

$$f(w) - f(w^*) \geq \frac{\lambda}{2} \|w-w^*\|^2$$

Hint: Use the definition of λ-strongly convex function, properly rearrange it, and ...

Exercise 10. ($\star$)Show that the function $J(x;\lambda) = \lambda \|x\|^2$ is 2λ-strongly convex

---

Exercise 11. ($\star\star\star$)Consider a learning problem $(\mathcal{H}, \mathcal{Z}, \ell)$ with $\mathcal{H} \subset \mathbb{R}^d$, $d > 0$, and loss function $\ell : \mathcal{H} \times \mathcal{Z} \to \mathbb{R}_+$ which is convex, β-smooth and non-negative. Let $\mathfrak{A}$ be a learning algorithm with output $\mathfrak{A}(\mathcal{S})$ trained against training dataset $\mathcal{S} = \{z_1, ..., z_m\}$ of IID samples $z_1, ..., z_m \sim g$ where g is a data generating distribution. In particular, consider that $\mathfrak{A}(\mathcal{S})$ is the Regularized Loss Minimization learning rule that outputs a hypothesis in

$$\min_w \left\{ \hat{R}_{\mathcal{S}}(w) + \lambda \|w\|_2^2 \right\}$$

for $\lambda \geq \frac{2\beta}{m}$ where $\hat{R}_{\mathcal{S}}(w) = \frac{1}{m}\sum_{i=1}^m \ell(w, z_i)$ for all $w \in \mathcal{H}$.

(1) Prove that

$$\mathrm{E}_{\mathcal{S}\sim g}\left(\hat{R}_{\mathcal{S}}\left(\mathfrak{A}\left(\mathcal{S}\right)\right)\right) \leq R_g\left(w\right) + \lambda \left\|w\right\|_2^2$$

for all $w \in \mathcal{H}$. $R_g(\cdot)$ denotes the risk function under the real data generating distribution g.

(2) Prove that

$$\mathrm{E}_{\mathcal{S}\sim g}\left(R_g\left(\mathfrak{A}\left(\mathcal{S}\right)\right) - \hat{R}_{\mathcal{S}}\left(\mathfrak{A}\left(\mathcal{S}\right)\right)\right) \leq \frac{48\beta}{\lambda m}\mathrm{E}_{\mathcal{S}\sim g}\left(\hat{R}_{\mathcal{S}}\left(\mathfrak{A}\left(\mathcal{S}\right)\right)\right).$$

Hint:: If needed you can use the following:

Let $\mathcal{S}^{(i)} = \{z_1, ..., z_{i-1}, z', z_{i+1}, ..., z_m\}$ be a set resulting from $\mathcal{S}$ by replacing its i-th element z_i with an independently drawn $z' \sim g$. Then

$$24\beta\ell\left(\mathfrak{A}\left(\mathcal{S}\right), z_i\right) + \lambda m\ell\left(\mathfrak{A}\left(\mathcal{S}\right), z_i\right) + 24\beta\ell\left(\mathfrak{A}\left(\mathcal{S}^{(i)}\right), z'\right) - \lambda m\ell\left(\mathfrak{A}\left(\mathcal{S}^{(i)}\right), z_i\right) \geq 0$$

(3) Show that the learning algorithm $\mathfrak{A}$ is on-average-replace-one-stable with rate ε. Specify that rate ε as a function of β, λ, m and possibly any other user specified constants if needed. Explain how the shrinkage parameter λ, the training dataset size m, and the smoothness parameter β affect the stability of the learning algorithm $\mathfrak{A}$.

(4) Show that the expected risk is bounded as follows

$$\mathrm{E}_{\mathcal{S}\sim g}\left(R_g\left(\mathfrak{A}\left(\mathcal{S}\right)\right)\right) \leq \left(1 + \frac{48\beta}{\lambda m}\right)\left(R_g\left(w\right) + \lambda \left\|w\right\|_2^2\right)$$

for all $w \in \mathcal{H}$.

Part 2. Stochastic learning

Exercise 12. ($\star$) Assume a Bayesian model

$$\begin{cases} z_i|w & \overset{\text{ind}}{\sim} f\left(z_i|w\right), \ i = 1, ..., n \\ w & \sim f\left(w\right) \end{cases}$$

and consider that our objective is the discovery of MAP estimate w^* i.e.

$$w^* = \arg\min_{\forall w\in\Theta}\left(-\log\left(L_n\left(w\right)\right) - f\left(w\right)\right) = \arg\min_{\forall w\in\Theta}\left(-\sum_{i=1}^{n}\log\left(f\left(z_i|w\right)\right) - \log\left(f\left(w\right)\right)\right)$$

by using SGD with update

$$w^{(t+1)} = w^{(t)} + \eta_t\left(\frac{n}{m}\sum_{j\in\mathcal{J}^{(t)}}\nabla_w \log\left(f\left(z_j|w^{(t)}\right)\right) + \nabla_w \log\left(f\left(w^{(t)}\right)\right)\right)$$

for some randomly selected set $\mathcal{J}^{(t)} \subseteq \{1, ..., n\}^m$ of m integers from 1 to n via simple random sampling (SRS) with replacement. Show that

$$\mathrm{E}_{\mathcal{J}^{(t)} \sim \text{simple-random-sampling}} \left(\frac{n}{m} \sum_{j \in \mathcal{J}^{(t)}} \nabla_w \log \left(f \left(z_j | w^{(t)} \right) \right) \right) = \sum_{i=1}^{n} \nabla_w \log \left(f \left(z_i | w^{(t)} \right) \right)$$

Exercise 13. ($\star$) Let $\{v_t; t = 1, ..., T\}$ be a sequence of vectors with $v_t \in \mathbb{R}^d$ and $d \in \mathbb{N} - \{0\}$. Consider an algorithm producing $\left\{w^{(t)}; t = 1, 2, 3, ...\right\}$ with

$$w^{(1)} = 0$$
$$w^{(t+1)} = w^{(t)} - \eta v_t$$

$w_t \in \mathbb{R}^d$ and $d \in \mathbb{N} - \{0\}$. Show that

(1) it is

$$\langle w^{(t)} - w^*, v_t \rangle = \frac{1}{2\eta} \left(- \left\| w^{(t+1)} - w^* \right\|^2 + \left\| w^{(t)} - w^* \right\|^2 \right) + \frac{\eta}{2} \left\| v_t \right\|^2$$

Hint:: Recall that

$$\|x + y\|_2^2 = \|x\|_2^2 + \|y\|_2^2 + 2 \langle x, y \rangle, \ \forall x, y \in \mathbb{R}^d, d \in \mathbb{N} - \{0\}$$

(2) it is

$$\sum_{t=1}^{T} \langle w^{(t)} - w^*, v_t \rangle = \frac{1}{2\eta} \sum_{t=1}^{T} \left(- \left\| w^{(t+1)} - w^* \right\|^2 + \left\| w^{(t)} - w^* \right\|^2 \right) + \frac{\eta}{2} \sum_{t=1}^{T} \left\| v_t \right\|^2$$

(3) (continue) it is

$$\sum_{t=1}^{T} \langle w^{(t)} - w^*, v_t \rangle \leq \frac{\|w^*\|^2}{2\eta} + \frac{\eta}{2} \sum_{t=1}^{T} \|v_t\|^2$$

Exercise 14. ($\star$) Let $\{v_t; t = 1, ..., T\}$ be a sequence of vectors. Consider an algorithm producing $\left\{w^{(t)}; t = 1, 2, 3, ...\right\}$ with

$$w^{(1)} = 0$$
$$w^{\left(t + \frac{1}{2}\right)} = w^{(t)} - \eta v_t$$
$$w^{(t+1)} = \arg\min_{w \in \mathcal{H}} \left(\left\| w - w^{\left(t + \frac{1}{2}\right)} \right\| \right)$$

for $t = 1, ..., T$.

Hint: You can use the following Lemma

(Projection Lemma): Let $\mathcal{H}$ be a closed convex set and let v be the projection of w onto $\mathcal{H}$,i.e.

$$v = \arg\min_{x \in \mathcal{H}} \|x - w\|^2$$

then for every $u \in \mathcal{H}$ it is

$$\|v-u\|^2 \leq \|w-u\|^2$$

Show that

(1) it is

$$\langle w^{(t)}-w^*, v_t\rangle \leq \frac{1}{2\eta}\left(-\left\|w^{(t+1)}-w^*\right\|^2+\left\|w^{(t)}-w^*\right\|^2\right)+\frac{\eta}{2}\|v_t\|^2$$

(2) it is

$$\sum_{t=1}^{T}\langle w^{(t)}-w^*, v_t\rangle \leq \frac{1}{2\eta}\sum_{t=1}^{T}\left(-\left\|w^{(t+1)}-w^*\right\|^2+\left\|w^{(t)}-w^*\right\|^2\right)+\frac{\eta}{2}\sum_{t=1}^{T}\|v_t\|^2$$

(3) (continue) it is

$$\sum_{t=1}^{T}\langle w^{(t)}-w^*, v_t\rangle \leq \frac{\|w^*\|^2}{2\eta}+\frac{\eta}{2}\sum_{t=1}^{T}\|v_t\|^2$$

Comment: Above we show that Lemma **??** from "Handout **??**: Gradient descent" holds even when a projection step is included. Hence, even if a projection step is included after the update step of the recursion of GD algorithm or the SGD algorithm the analysis in Section **??** in "Handout **??**: Gradient descent" holds. Hence, even if a projection step is included after the update step of the recursion of SGD algorithm or the SGD algorithm the analysis in Section **??** in "Handout **??**: Stochastic gradient descent" holds.

Exercise 15. ($\star$) [1]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

$$h_w(x) = \operatorname{sign}\left(w^\top x\right) \tag{0.1}$$

$$= \operatorname{sign}\left(\sum_{j=1}^{d} w_j x_j\right) \tag{0.2}$$

Let the hypothesis class is

$$\mathcal{H} = \left\{x \to w^\top x : \forall w \in \mathbb{R}^d\right\} \tag{0.3}$$

[1]We use standard notation

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

± 1 means either -1 or $+1$, $\mathbb{R}_+ := (0,+\infty)$, and $\|x\|_2 := \sqrt{\sum_{\forall j}(x_j)^2}$ for the Euclidean distance.

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 $y = \text{sign}\left(w^\top x\right) \in \mathcal{Y} := \{\pm 1\}$ where

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

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

$$\ell(w, z = (x, y)) = \max\left(0, 1 - y w^\top x\right) + \lambda \|w\|_2^2 \tag{0.4}$$

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:

(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 (0.4) is convex.

Hint:: You may use Proposition **??** from Handout **??**: Elements of convex learning problems.

(2) Show that the loss $\ell(w, z)$ for $\lambda = 0$ (0.4) 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 $1 - y w_2^\top x \leq 1 - y w_1^\top x$, and then take cases $1 - y w_2^\top x >$ or < 0 and $1 - y w_1^\top x >$ 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

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

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

$$w^* = \arg \min_{\forall w : h_w \in \mathcal{H}} \left(\mathrm{E}_{z \sim g}\left(\ell(w, z = (x, y))\right)\right) \tag{0.5}$$

The formulas in your algorithm shoud be implemented for the above learning problem and tailored to 0.1, 0.3, and 0.4.

(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 $\{w^{(t)}\}$ against the iteration t.

(c) Report the value of the output w^*_{adaGrad} (any type) of the algorithm as the solution to (0.5).

(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)
```

Part 3. Support Vector Machines

Part 4. The kernel trick

Part 5. **Multi-class classification**

Part 6. **Artificial Neural Networks**

Part 7. **Gaussian process regression**

Part 8. **Revision**