

Exercise sheet

Lecturer: Georgios P. Karagiannis

georgios.karagiannis@durham.ac.uk

Part 1. Stochastic learning

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

Exercise 2. (★) 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. (★) 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 $(\beta \|x\|^2)$ -smooth.

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

Exercise 4. (★) 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 $\|v\| \leq \rho$.

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

Exercise 5. (★) Let $g_1(w), \dots, g_r(w)$ be r convex functions, and let $f(\cdot) = \max_{j \in [r]} (g_j(\cdot))$. 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. (★) 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 $\mathcal{W} = \mathcal{X} = \{\omega \in \mathbb{R}^d : |\omega| \leq \rho\}$ for some $\rho > 0$.

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

The following is given as a homework (Formative assessment 1)

Exercise 7. (★) Consider the binary classification problem with inputs $x \in \mathcal{X}$ where $\mathcal{X} := \{x \in \mathbb{R}^d : \|x\|_2 \leq L\}$ for some given value $L > 0$, target $y \in \mathcal{Y}$ where $\mathcal{Y} := \{-1, +1\}$, and prediction rule $h_w : \mathbb{R}^d \rightarrow \{-1, +1\}$ with

$$(1) \quad h_w(x) = \text{sign}(w^\top x)$$

$$(2) \quad = \text{sign}\left(\sum_{j=1}^d w_j x_j\right)$$

Let the hypothesis class of prediction rules be

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

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

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

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

for some given value $\lambda > 0$.

Assume there is available a dataset of examples $S_n = \{z_i = (x_i, y_i) ; i = 1, \dots, n\}$ 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 $\|x\|_2 := \sqrt{\sum_{j=1}^d (x_j)^2}$ the Euclidean distance.

- (1) Show that the function $f : \mathbb{R} \rightarrow \mathbb{R}_+$ 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 $\mathcal{X} := \{x \in \mathbb{R}^d : \|x\|_2 \leq L\}$.

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 $1 - yw_2^\top x > \text{or} < 0$ and $1 - yw_1^\top x > \text{or} < 0$ to deal with the max.

- (3) Construct the set of sub-gradients $\partial f(x)$ for $x \in \mathbb{R}$ of the function $f : \mathbb{R} \rightarrow \mathbb{R}_+$ 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

$$(4) \quad w^* = \arg \min_{w: h_w \in \mathcal{H}} (\mathbb{E}_{z \sim g} (\ell(w, z = (x, y))))$$

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 $\{w^{(t)}\}$ 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)

```

Exercise 8. (★) Assume a Bayesian model

$$\begin{cases} z_i|w & \stackrel{\text{ind}}{\sim} f(z_i|w), \quad i = 1, \dots, n \\ w & \sim f(w) \end{cases}$$

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

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

by using SGD with update

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

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

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

Part 2. Artificial Neural Networks

Exercise 9. (★) Consider the regression problem, with a predictive rule $h_w : \mathbb{R}^d \rightarrow \mathbb{R}$, as a classification probability, that receives values $x \in \mathbb{R}^d$ returns values in \mathbb{R} . Let $h_w(x)$ be modeled as an ANN

$$h(x) = \sigma_2 \left(\sum_{j=1}^c w_{2,1,j} \sigma_1 \left(\sum_{i=1}^d w_{1,j,i} x_i \right) \right)$$

and let the associated activation function be

$$\sigma_2(a) = a \Phi(a)$$

where $\Phi(x) = \int_{-\infty}^x \phi(t) dt$ is considered as known function, and $\phi(t) = \frac{1}{\sqrt{2\pi}} \exp(-\frac{1}{2}t^2)$ and

$$\sigma_1(a) = \exp(-a^2)$$

Consider a loss

$$\ell(w, z = (x, y)) = \frac{1}{2} (y - h_w(x))^2$$

at w and example $z = (x, y)$, where $x \in \mathbb{R}^d$ is the input vector (features), and y is the output vector (targets) with $y \in \mathbb{R}$. Consider that d , c , and q are known integers.

- (1) Perform the forward pass of the back-propagation procedure to compute the activations which may be denoted as $\{a_{t,i}\}$ and outputs which may be denoted as $\{o_{t,i}\}$ at each layer t .
 - (2) Perform the backward pass of the back-propagation procedure in order to compute the elements of the gradient $\nabla_w \ell(w, (x, y))$.
-

Exercise 10. (★) Students are encouraged to practice on the Exercises 5.1-5.28 from the textbook

- Bishop, C. M. (2006). Pattern recognition and machine learning (Vol. 4, No. 4, p. 738). New York: Springer.

available from

- <https://www.microsoft.com/en-us/research/uploads/prod/2006/01/Bishop-Pattern-Recognition-and-Machine-Learning-2006.pdf>
- <https://www.microsoft.com/en-us/research/uploads/prod/2006/01/Bishop-Pattern-Recognition-and-Machine-Learning-2006.pdf>

The solutions are available from

- <https://www.microsoft.com/en-us/research/wp-content/uploads/2016/05/prml-web-sol-2009-09-08.pdf>
- <https://www.maths.dur.ac.uk/users/georgios.karagiannis/temp/Pattern%20recognition%20and%20machine%20learning-solutions-1.pdf>
- <https://www.maths.dur.ac.uk/users/georgios.karagiannis/temp/Pattern%20recognition%20and%20machine%20learning-solutions-2.pdf>

The following is given as a homework (Formative assessment 2)

Exercise 11. (★) Consider the multi-class classification problem, with a predictive rule $h_w : \mathbb{R}^d \rightarrow \mathcal{P}$, as a classification probability i.e, $h_{w,k}(x) = \Pr(x \text{ belongs to class } k)$, that receives values $x \in \mathbb{R}^d$ returns vales in $\mathcal{P} = \left\{ p \in (0, 1)^q : \sum_{j=1}^q p_j = 1 \right\}$. Let $h_w = (h_{w,1}, \dots, h_{w,q})^\top$, let $h_w(x)$ be modeled as an ANN

$$h_k(x) = \sigma_2 \left(\sum_{j=1}^c w_{2,k,j} \sigma_1 \left(\sum_{i=1}^d w_{1,j,i} x_i \right) \right)$$

for $k = 1, \dots, q$, and let the associated activation functions be

$$\sigma_2(a_k) = \frac{\exp(a_k)}{\sum_{k'=1}^q \exp(a_{k'})}, \text{ for } k = 1, \dots, q$$

(called softmax function) and $\sigma_1(a) = \arctan(a)$. Consider a loss

$$\ell(w, z = (x, y)) = - \sum_{k=1}^q y_k \log(h_{w,k}(x))$$

at w and example $z = (x, y)$, where $x \in \mathbb{R}^d$ is the input vector (features), and $y = (y_1, \dots, y_q)$ is the output vector (labels) with $y \in \{0, 1\}^q$ and $\sum_{k=1}^q y_k = 1$. Consider that d , c , and q are known integers.

Hint: You may use

$$\frac{d}{dx} \arctan(x) = \frac{1}{1+x^2}$$

- (1) Perform the forward pass of the back-propagation procedure to compute the activations which may be denoted as $\{a_{t,i}\}$ and outputs which may be denoted as $\{o_{t,i}\}$ at each layer t .
- (2) Show that

$$\frac{\partial}{\partial a_k} \sigma_2(a_j) = \sigma_2(a_j) (1(j=k) - \sigma_2(a_k))$$

$$\text{for } k = 1, \dots, q. \text{ Let } 1(j = k) = \begin{cases} 1 & j = k \\ 0 & j \neq k \end{cases}.$$

- (3) Perform the backward pass of the back-propagation procedure in order to compute the elements of the gradient $\nabla_w \ell(w, (x, y))$.

Part 3. Support Vector Machines

The following is given as a homework (Formative assessment 3)

Exercise 12. (★★) Consider a training data set $\mathcal{D} = \{z_i = (x_i, y_i)\}_{i=1}^m$. Consider the Soft-SVM Algorithm that requires the solution of the following quadratic minimization problem (in a slightly modified but equivalent form to what we have discussed)

Primal problem:

$$(5) \quad (w^*, b^*, \xi^*) = \arg \min_{(w, b, \xi)} \left(\frac{1}{2} \|w\|_2^2 + C \sum_{i=1}^m \xi_i \right)$$

$$(6) \quad \text{subject to: } y_i (\langle w, x_i \rangle + b) \geq 1 - \xi_i, \quad \forall i = 1, \dots, m$$

$$(7) \quad \xi_i \geq 0, \quad \forall i = 1, \dots, m$$

for some user-specified fixed parameter $C > 0$.

- (1) Specify the Lagrangian function L associated to the above primal quadratic minimization problem, where $\{\alpha_i\}$ are the Lagrange coefficients wrt (6), and $\{\beta_i\}$ are the Lagrange coefficients wrt (7). Write down any possible restrictions on the Lagrange coefficients.
- (2) Compute the dual Lagrangian function denoted as \tilde{L} as a function of the Lagrange coefficients and the data points \mathcal{D} .
- (3) Apply the Karush–Kuhn–Tucker (KKT) conditions to the above problem, and write them down.
- (4) Derive and write down the dual Lagrangian quadratic maximization problem, along with the inequality and equality constraints, where you seek to find $\{\alpha_i\}$.
- (5) Justify why the i -th point x_i lies on the margin boundary when $\alpha_i \in (0, C)$ (beware it is $\alpha_i \neq C$), and why the i -th point x_i lies inside the margin when $\alpha_i = C$.
- (6) Given optimal values $\{\alpha_i^*\}$ for Lagrangian coefficients $\{\alpha_i\}$ as they are derived by solving the dual Lagrangian maximization problem in part 4, derive the optimal values w^* and b^* for the parameters w and b as function of the support vectors. Regarding parameter b it should be derived in the form

$$b^* = \frac{1}{|\mathcal{M}|} \sum_{i \in \mathcal{M}} \left(y_i - \sum_{j \in \mathcal{S}} \alpha_j^* y_j \langle x_j, x_i \rangle \right)$$

where you determine the sets \mathcal{M} and \mathcal{S} .

- (7) Report the halfspace predictive rule $h_{w,b}(x)$ of the above problem as a function of α^* and b^* .
-

Exercise 13. (★★) Show that K with

$$K(x, y) = \frac{\sin\left(2\pi\left(N + \frac{1}{2}\right)(x - y)\right)}{\sin(\pi(x - y))}$$

is a valid kernel.

Hint-1: You may use that $\sum_{n=0}^r z^n = \frac{1-z^{r+1}}{1-z}$

Hint-2: You may use that $e^{ix} = \cos(x) + i \sin(x)$

Exercise 14. (★) Students are encouraged to practice on the Exercises 6.1-6.19 from the textbook

- Bishop, C. M. (2006). Pattern recognition and machine learning (Vol. 4, No. 4, p. 738). New York: Springer.

available from

- <https://www.microsoft.com/en-us/research/uploads/prod/2006/01/Bishop-Pattern-Recognition-pdf>
 - <https://www.microsoft.com/en-us/research/uploads/prod/2006/01/Bishop-Pattern-Recognition-and-Machine-Learning-2006.pdf>

The solutions are available from

- <https://www.microsoft.com/en-us/research/wp-content/uploads/2016/05/prml-web-sol-2009-09-pdf>
 - <https://www.microsoft.com/en-us/research/wp-content/uploads/2016/05/prml-web-sol-2009-09-08.pdf>
 - <https://www.maths.dur.ac.uk/users/georgios.karagiannis/temp/Pattern%20recognition%20and%20machine%20learning-solutions-1.pdf>
 - <https://www.maths.dur.ac.uk/users/georgios.karagiannis/temp/Pattern%20recognition%20and%20machine%20learning-solutions-1.pdf>
 - <https://www.maths.dur.ac.uk/users/georgios.karagiannis/temp/Pattern%20recognition%20and%20machine%20learning-solutions-2.pdf>
 - <https://www.maths.dur.ac.uk/users/georgios.karagiannis/temp/Pattern%20recognition%20and%20machine%20learning-solutions-2.pdf>
-

Part 4. Gaussian process regression

Exercise 15. (★) Students are encouraged to practice on the Exercises 6.19-6.27 from the textbook

- Bishop, C. M. (2006). Pattern recognition and machine learning (Vol. 4, No. 4, p. 738). New York: Springer.

available from

- <https://www.microsoft.com/en-us/research/uploads/prod/2006/01/Bishop-Pattern-Recognition-and-Machine-Learning-2006.pdf>
 - <https://www.microsoft.com/en-us/research/uploads/prod/2006/01/Bishop-Pattern-Recognition-and-Machine-Learning-2006.pdf>

The solutions are available from

- <https://www.microsoft.com/en-us/research/wp-content/uploads/2016/05/prml-web-sol-2009-09-08.pdf>
 - <https://www.microsoft.com/en-us/research/wp-content/uploads/2016/05/prml-web-sol-2009-09-08.pdf>
 - <https://www.maths.dur.ac.uk/users/georgios.karagiannis/temp/Pattern%20recognition%20and%20machine%20learning-solutions-1.pdf>
 - <https://www.maths.dur.ac.uk/users/georgios.karagiannis/temp/Pattern%20recognition%20and%20machine%20learning-solutions-1.pdf>
 - <https://www.maths.dur.ac.uk/users/georgios.karagiannis/temp/Pattern%20recognition%20and%20machine%20learning-solutions-2.pdf>
 - <https://www.maths.dur.ac.uk/users/georgios.karagiannis/temp/Pattern%20recognition%20and%20machine%20learning-solutions-2.pdf>
-