

## 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.

**Solution.** Let  $u, v \in \mathbb{R}^d$  and  $\alpha \in [0, 1]$ . It is

$$\begin{aligned}
 f(\alpha u + (1 - \alpha)v) &= g(\langle \alpha u + (1 - \alpha)v, x \rangle + y) \\
 &= g(\langle \alpha u, x \rangle + \langle (1 - \alpha)v, x \rangle + y) \\
 &= g(\alpha \langle u, x \rangle + y + (1 - \alpha) \langle v, x \rangle + y) & y = \alpha y + (1 - \alpha)y \\
 &\leq \alpha g(\langle u, x \rangle + y) + (1 - \alpha) g(\langle v, x \rangle + y) & (g \text{ is convex}) \\
 &= \alpha f(u) + (1 - \alpha) f(v)
 \end{aligned}$$

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

**Solution.**

$$\begin{aligned}
 |f(w_1) - f(w_2)| &= |g_1(g_2(w_1)) - g_1(g_2(w_2))| \\
 &\leq \rho_1 |g_2(w_1) - g_2(w_2)| \\
 &\leq \rho_1 \rho_2 |w_1 - w_2|
 \end{aligned}$$

**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  $\beta$ -smooth function. Then show that  $f$  is a  $(\beta \|x\|^2)$ -smooth.

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

$$f(v) = g(\langle w, x \rangle + y)$$

$$\leq g(\langle w, x \rangle + y) + g'(\langle w, x \rangle + y) \langle v - w, x \rangle + \frac{\beta}{2} (\langle v - w, x \rangle)^2 \quad (g \text{ is smooth})$$

$$\leq g(\langle w, x \rangle + y) + g'(\langle w, x \rangle + y) \langle v - w, x \rangle + \frac{\beta}{2} (\|v - w\| \|x\|)^2 \quad (\text{Cauchy-Schwarz inequality})$$

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

**Exercise 4.** (★) Show that  $f : S \rightarrow \mathbb{R}$  is  $\rho$ -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  $\langle y, x \rangle \leq \|y\| \|x\|$

**Solution.**  $\Rightarrow$  Let  $f : S \rightarrow \mathbb{R}$  be  $\rho$ -Lipschitz over convex set  $S$ ,  $w \in S$  and  $v \in \partial f(w)$ .

- Since  $S$  is open we get that there exist  $\epsilon > 0$  such as  $u := w + \epsilon \frac{v}{\|v\|}$  where  $u \in S$ . So  $\langle u - w, v \rangle = \epsilon \|v\|$  and  $\|u - w\| = \epsilon$ .
- From the subgradient definition we get

$$f(u) - f(w) \geq \langle u - w, v \rangle = \epsilon \|v\|$$

- From the Lipschitzness of  $f(\cdot)$  we get

$$f(u) - f(w) \leq \rho \|u - w\| = \rho \epsilon$$

Therefore  $\|v\| \leq \rho$ .

$\Leftarrow$  It is for all  $w \in S$  and  $v \in \partial f(w)$  it is  $\|v\| \leq \rho$ .

- For any  $u \in S$ , it is

$$\begin{aligned} f(w) - f(u) &\leq \langle v, w - u \rangle && (\text{because } v \in \partial f(w)) \\ (1) \quad &\leq \|v\| \|w - u\| && \text{by Cauchy-Schwarz inequality} \\ &\leq \rho \|w - u\| && \text{because } \|v\| \leq \rho \end{aligned}$$

- Similarly it results  $u, w \in S$

$$f(w) - f(u) \leq \langle v, u - w \rangle \|v\| \leq \|v\| \|u - w\| \leq \rho \|u - w\|$$

from (1) because  $w, u$  can be swapped in (1) as they both are any values in  $S$ .

**Exercise 5.** (★) Let  $g_1(w), \dots, g_r(w)$  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  $w$ .

**Solution.** Since  $g_k$  is convex, for all  $u$

$$g_k(u) \geq g_k(w) + \langle u - w, \nabla g_k(w) \rangle$$

However  $f(u) = \max_{\forall j} (g_j(u)) \geq g_k(u)$  for any  $j$ , and  $f(w) = g_k(w)$  at  $w$ . Then

$$\begin{aligned} f(u) &\geq g_k(u) \\ &\geq g_k(w) + \langle u - w, \nabla g_k(w) \rangle \\ &= f(w) + \langle u - w, \nabla g_k(w) \rangle \end{aligned}$$

Then by the definition of the sub-gradient  $\nabla g_k(w) \in \partial f(w)$

**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.

**Solution.** According to the definitions given in the lecture:

- Convex-Lipschitz-Bounded Learning Problem  $(\mathcal{H}, \mathcal{Z}, \ell)$  with parameters  $\rho$ , and  $B$ , is called the learning problem whose the hypothesis class  $\mathcal{H}$  is a convex set, for all  $w \in \mathcal{H}$  it is  $\|w\| \leq B$ , and the loss function  $\ell(\cdot, z)$  is convex and  $\rho$ -Lipschitz function for all  $z \in \mathcal{Z}$ .

I have:

**Convexity:** The function  $g : \mathbb{R} \rightarrow \mathbb{R}$ , defined by  $g(a) = a^2$  is convex. Eg.  $\frac{d^2}{da^2} g(a) = 2 \geq 0$  is non-negative. The convexity of  $\ell(w, z = (x, y))$  for all  $z$  follows as a composition of  $g$  with a linear function.

**Lipschitzness:** The function  $g(a) = a^2$  is 1-Lipschitz since It is

$$|g(a_2) - g(a_1)| = |a_2^2 - a_1^2| = |(a_2 + a_1)(a_2 - a_1)| \leq 2\rho(a_2 - a_1) = 2\rho|a_2 - a_1|$$

Hence because  $|x| \leq \rho$ ,  $g(a)$  is  $2\rho^2$ -Lipschitz as a composition.

**Boundness:** The norm of each hypothesis  $w$  is bounded by  $\rho$  according to the assumptions.

Therefore,

- (1) the learning problem under consideration is a Convex-Lipschitz-Bounded learning problem.
- (2) the parameter of Lipschitzness is  $2\rho^2$ .

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

$$(2) \quad h_w(x) = \text{sign}(w^\top x)$$

$$(3) \quad = \text{sign}\left(\sum_{j=1}^d w_j x_j\right)$$

Let the hypothesis class of prediction rules be

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

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

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

$$(4) \quad \ell(w, z = (x, y)) = \max(0, 1 - yw^\top x) + \lambda \|w\|_2^2$$

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  $\pm 1$  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 (4) is convex.

**Hint::** You may use Example 13 from Handout 1.

- (2) Show that the loss  $\ell(w, z)$  for  $\lambda = 0$  (4) 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  $1 - yw_2^\top x > 0$  or  $< 0$  and  $1 - yw_1^\top x > 0$  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

$$(5) \quad w^* = \arg \min_{\forall w: h_w \in \mathcal{H}} (\mathbb{E}_{z \sim g} (\ell(w, z = (x, y))))$$

The formulas in your algorithm have to be tailored to 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  $m$ ,  $\eta_t$  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  $w_{\text{adaGrad}}^*$  (any type) of the algorithm as the solution to (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)

```

**Solution.**

- (1)  $f_1(x) = 0$  is convex,  $f_2(x) = 1 - x$  is convex, hence from the example in Handout 1,  $f(x) = \max(f_1(x), f_2(x))$  is convex as well. Regarding the loss function, we just have  $f_2(w) = 1 - yx^\top w$  which is convex as a composition due to linearity.
- (2) Given a fixed example  $(x, y) \in \{x \in \mathbb{R}^d : \|x'\|_2 \leq R\} \times \{-1, 1\}$ .

Assume  $w_1, w_2 \in \mathbb{R}^d$ . Let  $\ell_i = \max\{0, 1 - yx^\top w_i\}$ , for  $i = 1, 2$ . It suffices to show that  $|\ell_1 - \ell_2|_2 \leq R|w_1 - w_2|_2$ . I take cases

**Case-1:** Assume  $yx^\top w_1 \geq 1$  and  $yx^\top w_2 \geq 1$  then  $|\ell_1 - \ell_2|_2 = 0 \leq R|w_1 - w_2|_2$

**Case-2:** Assume that at least one of  $yx^\top w_1 < 1$  or  $yx^\top w_2 < 1$  but not both is true.

Assume without loss of generality that  $1 - yx^\top w_1 < 1 - yx^\top w_2$ . Then

$$\begin{aligned}
|\ell_1 - \ell_2|_2 &= \ell_1 - \ell_2 \\
&= 1 - yx^\top w_1 - \max(0, 1 - yx^\top w_2) \\
&\leq 1 - yx^\top w_1 - (1 - yx^\top w_2) \\
&= yx^\top (w_2 - w_1) \\
&\leq y \left\| x^\top \right\|_2 \|w_1 - w_2\|_2 \quad \text{because } a^\top b \leq \|a\| \|b\|
\end{aligned}$$

(3) It is

$$f(x) = \max(0, 1 - x) = \begin{cases} 0 & x > 1 \\ 0 & x = 1 \\ 1 - x & x < 1 \end{cases}$$

- For  $x > 1$ ,  $f$  is differentiable so  $\partial f(x) = \{f'(x)\} = \{0\}$ .
- For  $x < 1$ ,  $f$  is differentiable so  $\partial f(x) = \{f'(x)\} = \{-1\}$ .
- For  $x = 1$ ,  $f$  is not differentiable. By definition I have that  $v$  is subgradient of  $f(x)$  at  $x = 0 \in S$  if

$$\forall u \in \mathbb{R}, \quad f(u) \geq f(x) + \langle u - x, v \rangle$$

So, for  $u \geq 1$ , it is  $0 \geq (u - 1)v \implies v \leq 0$ , and for  $u < 1$  it is  $(1 - u) \geq (u - 1)v \implies v \geq -1$ . Hence the common space is  $v \in [0, 1]$  So  $\partial f(x) = [0, 1]$ . Hence,

$$\partial f(x) = \begin{cases} 0, & x > 1 \\ [-1, 0], & x = 1 \\ -1, & x < 1 \end{cases}$$

Now regarding the loss  $\partial_w \ell(w, z = (x, y))$

- for  $yw^\top x > 1$  it is differentiable so  $\nabla_w \ell(w, z = (x, y)) = \nabla_w (0 + \lambda \sum_{j=1}^d w_j^2) = 2\lambda w$ ;  
as

$$\frac{d}{dw_j} \sum_{j'=1}^d w_{j'}^2 = 2\lambda w_j$$

- for  $yw^\top x < 1$  it is differentiable so  $\nabla_w \ell(w, z = (x, y)) = \nabla_w (1 - yw^\top x + \lambda \sum_{j=1}^d w_j^2) = yx + 2\lambda w$  as

$$\frac{d}{dw_j} (1 - yw^\top x) = \frac{d}{dw_j} \left( 1 - y \sum_{j'=1}^d w_{j'} x_{j'} \right) = -yx_j$$

- for  $yw^\top x = 1$ ,  $v = 0$  satisfies the definition of the sub-gradient

$$\begin{aligned} \forall u, \quad f(u) &\geq \cancel{f(w)}^0 + \langle u - w, v \rangle \\ \max(0, 1 - yu^\top x) &\geq 0 + (u - w)^\top 0 \end{aligned}$$

So

$$\begin{aligned} \partial \ell(w, z = (x, y)) &= \partial \left( \max(0, 1 - yw^\top x) + \lambda \|w\|_2^2 \right) \\ &= \partial \left( \max(0, 1 - yw^\top x) \right) + \partial \left( \lambda \|w\|_2^2 \right) \\ &= \partial \left( \max(0, 1 - yw^\top x) \right) + \nabla \left( \lambda \|w\|_2^2 \right) \\ &= 0 + 2\lambda w \end{aligned}$$

but  $\partial \left( \lambda \|w\|_2^2 \right) = \left\{ \nabla \left( \lambda \|w\|_2^2 \right) \right\}$  because  $\lambda \|w\|_2^2$  is differentiable. Hence

$$\partial \ell(w, z = (x, y)) = 0 + 2\lambda w$$

Hence

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

(4)

**Algorithm.** For  $t = 1, 2, 3, \dots$  iterate:

- Get a random sub-sample  $\left\{ \tilde{z}_i^{(t)} = (\tilde{x}_i^{(t)}, \tilde{y}_i^{(t)}) ; i = 1, \dots, m \right\}$  of size  $m$  with or without replacement from the complete data-set  $\mathcal{S}_n$ .
- For  $j = 1, \dots, d$  (index  $j$  indicates the dimension of  $w$ ) compute

$$w_j^{(t+1)} = w_j^{(t)} - \eta_t \frac{1}{\sqrt{[G_t]_{j,j} + \epsilon}} \bar{v}_{t,j}$$

$[G_t]_{j,j} = [G_{t-1}]_{j,j} + (\bar{v}_{t,j})^2$  where  $\bar{v}_t = \frac{1}{m} \sum_{i=1}^m \tilde{v}_{t,i}$  and

$$\tilde{v}_{t,i} = \begin{cases} 2\lambda w^{(t)}, & \tilde{y}_i^{(t)} (w^{(t)})^\top \tilde{x}_i^{(t)} > 1 \\ 2\lambda w^{(t)}, & \tilde{y}_i^{(t)} (w^{(t)})^\top \tilde{x}_i^{(t)} = 1 \\ -\frac{1}{m} \tilde{y}_i^{(t)} \tilde{x}_i^{(t)} + 2\lambda w^{(t)}, & \tilde{y}_i^{(t)} (w^{(t)})^\top \tilde{x}_i^{(t)} < 1 \end{cases}$$

where index  $i$  indicates the sub-sample, and  $\epsilon > 0$  small.

- Terminate if a termination criterion is satisfied

(5)

- The R code can be found in the link [https://raw.githubusercontent.com/georgios-stats/Machine\\_Learning\\_and\\_Neural\\_Networks\\_III\\_Epiphany\\_2023/main/Exercises/supplementary/q6\\_adagrad.R](https://raw.githubusercontent.com/georgios-stats/Machine_Learning_and_Neural_Networks_III_Epiphany_2023/main/Exercises/supplementary/q6_adagrad.R)
- The figures are presented below



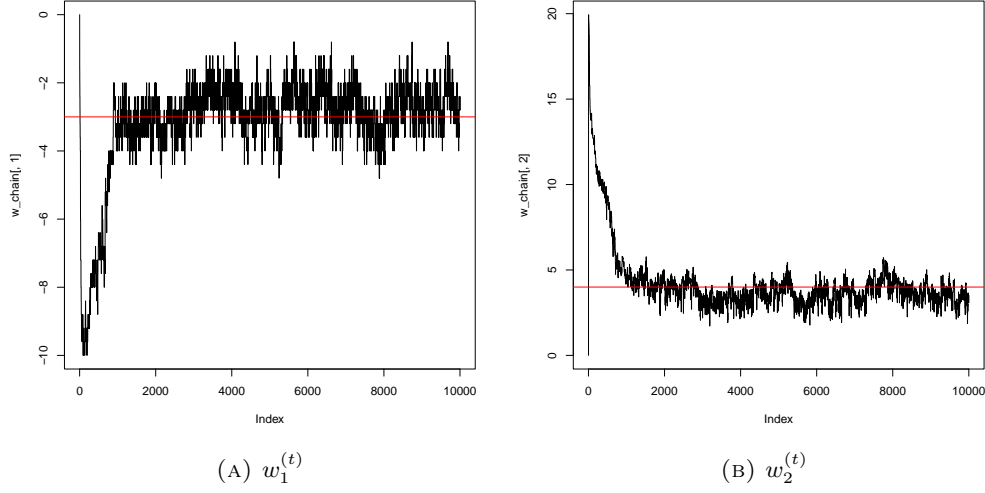


FIGURE 1. trace plots

- (c) I found  $w = (-2.674615, 3.205785)$
- (d) It belongs to  $-1$

**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

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

**Solution.** It is

$$\begin{aligned}
\mathbb{E}_{\mathcal{J}^{(t)} \sim \text{SRS}} \left( \frac{n}{m} \sum_{j \in \mathcal{J}^{(t)}} \nabla_w \log \left( f \left( z_j | w^{(t)} \right) \right) \right) &= \frac{n}{m} \sum_{j \in \mathcal{J}^{(t)}} \mathbb{E}_{\mathcal{J}^{(t)} \sim \text{SRS}} \left( \nabla_w \log \left( f \left( z_j | w^{(t)} \right) \right) \right) \\
&= \frac{n}{m} \sum_{j \in \mathcal{J}^{(t)}} \mathbb{E}_{\mathcal{J}^{(t)} \sim \text{SRS}} \left( \nabla_w \log \left( f \left( z_j | w^{(t)} \right) \right) \right) \\
&= \frac{n}{m} \sum_{j \in \mathcal{J}^{(t)}} \frac{1}{n} \sum_{i=1}^n \nabla_w \log \left( f \left( z_i | w^{(t)} \right) \right) \\
&= \sum_{i=1}^n \nabla_w \log \left( f \left( z_i | w^{(t)} \right) \right)
\end{aligned}$$

It is  $\mathbb{E}_{\mathcal{J}^{(t)} \sim \text{SRS}} \left( \nabla_w \log \left( f \left( z_j | w^{(t)} \right) \right) \right) = \frac{1}{n} \sum_{i=1}^n \nabla_w \log \left( f \left( z_i | w^{(t)} \right) \right)$  because the expectation is under the probability I get randomly an integer and for the  $j$ th on the probability is  $1/n$  due to the random scheme. Also  $|\mathcal{J}^{(t)}| = m$ .

---

## 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  $\phi(t) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}t^2\right)$  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  $\nabla_w \ell(w, (x, y))$ .

**Solution.**

- (1)

**Forward pass:**

**Set:**  $o_{0,i} = x_i$  for  $i = 1, \dots, d$

**Compute:**

**at**  $t = 1$ : **for**  $j = 1, \dots, c$

**comp:**  $\alpha_{1,j} = \sum_{i=1}^d w_{1,i,j} x_i$

**comp:**  $o_{1,j} = \exp(-\alpha_{1,j}^2)$

**at**  $t = 2$ :

**comp:**  $\alpha_{2,1} = \sum_{j=1}^c w_{2,1,j} o_{1,j}$

**comp:**  $o_{2,1} = \alpha_{2,1} \Phi(\alpha_{2,1})$

**get:**  $h_1 = o_{2,1}$

(2) It is

$$\frac{d}{da} \sigma_1(a) = -2a \exp(-a^2)$$

and

$$\frac{d}{da} \sigma_2(a) = \Phi(a) + a\phi(a)$$

and

$$\frac{d\ell_2}{d\alpha_{2,1}} = \frac{d\ell_2}{do_{2,1}} \frac{do_{2,1}}{d\alpha_{2,1}} = (o_{2,1} - y_1) (\Phi(\alpha_{2,1}) + \alpha_{2,1}\phi(\alpha_{2,1}))$$

**Backward pass:**

**at**  $t = 2$ :

**comp:**  $\tilde{\delta}_{2,1} = \frac{d}{d\alpha_{2,1}} \ell_T = (o_{2,1} - y_1) (\Phi(\alpha_{2,1}) + \alpha_{2,1}\phi(\alpha_{2,1}))$

**at**  $t = 1$ : **for**  $j = 1, \dots, c$

**comp:**

$$\begin{aligned} \tilde{\delta}_{1,j} &= \left. \frac{d}{d\xi} \sigma_1(\xi) \right|_{\xi=\alpha_{1,j}} \sum_{k=1}^q w_{2,k,j} \tilde{\delta}_{2,k} \\ &= -2\alpha_{1,j} \exp(-\alpha_{1,j}^2) \sum_{k=1}^q w_{2,k,j} \tilde{\delta}_{2,k} \end{aligned}$$

**Output:**

$$\frac{d}{dw_{1,j,i}} \ell = \tilde{\delta}_{1,j} x_i \text{ and } \frac{d}{dw_{2,k,j}} \ell = \tilde{\delta}_{2,k} o_{1,j}$$

**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-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>

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 values 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

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

for  $k = 1, \dots, q$ . 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  $\nabla_w \ell(w, (x, y))$ .

**Solution.**

- (1) Forward pass

**Set:**  $o_{0,i} = x_i$  for  $i = 1, \dots, d$

**Compute:**

**at**  $t = 1$ : for  $j = 1, \dots, c$

**comp:**  $\alpha_{1,j} = \sum_{i=1}^d w_{1,i,j} x_i$

**comp:**  $o_{1,j} = \arctan(\alpha_{1,j})$

**at**  $t = 2$ : for  $k = 1, \dots, q$

**comp:**  $\alpha_{2,k} = \sum_{j=1}^c w_{2,k,j} o_{1,j}$

**comp:**  $o_{2,k} = \frac{\exp(\alpha_{2,k})}{\sum_{k'=1}^q \exp(\alpha_{2,k'})}$

**get:**  $h_k = o_{2,k}$

- (2) It is

$$\begin{aligned} \frac{d}{da_k} \sigma_2(a_j) &= \frac{d}{da_k} \frac{\exp(a_j)}{\sum_{j'} \exp(a_{j'})} = \begin{cases} \sigma_2(a_j) (1 - \sigma_2(a_j)) & j = k \\ -\sigma_2(a_j) \sigma_2(a_k) & j \neq k \end{cases} \\ &= \sigma_2(a_j) (1(j = k) - \sigma_2(a_k)) \end{aligned}$$

- (3) It is

$$\frac{d}{da} \sigma_1(a) = \frac{1}{1 + a^2}$$

and

$$\begin{aligned} \frac{d}{da_k} \sigma_2(a_k) &= \sigma_2(a_j) (1(j = k) - \sigma_2(a_k)) \\ &= o_j (1(j = k) - o_k) \end{aligned}$$

and

$$\frac{d\ell_2}{do_{2,j}} = -y_j \frac{1}{o_{2,j}}$$

and

$$\begin{aligned}
\frac{d\ell_2}{da_{2,k}} &= \sum_{j=1}^q \frac{d\ell_2}{do_{2,j}} \frac{do_{2,j}}{do_{2,k}} \\
&= \sum_{j=1}^q \left( -y_j \frac{1}{o_{2,j}} o_{2,j} (1 (j = k) - o_{2,k}) \right) \\
&= \sum_{j=1}^q (-y_j (1 (j = k) - o_{2,k})) \\
&= o_{2,k} - y_k
\end{aligned}$$

**Backward pass:**

**at**  $t = 2$ : **for**  $k = 1, \dots, q$   
**comp:**  $\tilde{\delta}_{2,k} = \frac{d}{da_{2,k}} \ell_T = o_{2,k} - y_k$   
**at**  $t = 1$ : **for**  $j = 1, \dots, c$   
**comp:**

$$\begin{aligned}
\tilde{\delta}_{1,j} &= \left. \frac{d}{d\xi} \sigma_1(\xi) \right|_{\xi=\alpha_{1,j}} \sum_{k=1}^q w_{2,k,j} \tilde{\delta}_{2,k} \\
&= \left( \frac{1}{1 + \alpha_{1,j}^2} \right) \sum_{k=1}^q w_{2,k,j} \tilde{\delta}_{2,k}
\end{aligned}$$

**Output:**

$$\frac{d}{dw_{1,j,i}} \ell = \tilde{\delta}_{1,j} x_i \text{ and } \frac{d}{dw_{2,k,j}} \ell = \tilde{\delta}_{2,k} o_{1,j}$$


---

### 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:**

$$(6) \quad (w^*, b^*, \xi^*) = \arg \min_{(w, b, \xi)} \left( \frac{1}{2} \|w\|_2^2 + C \sum_{i=1}^m \xi_i \right)$$

$$(7) \quad \text{subject to: } y_i (\langle w, x_i \rangle + b) \geq 1 - \xi_i, \quad \forall i = 1, \dots, m$$

$$(8) \quad \xi_i \geq 0, \quad \forall i = 1, \dots, m$$

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 (7), and  $\{\beta_i\}$  are the Lagrange coefficients wrt (8). 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  $\alpha^*$  and  $b^*$ .

### Solution.

- (1) It is

$$(9) \quad L(w, b, \xi, \alpha, \beta) = \frac{1}{2} \|w\|_2^2 + \sum_{i=1}^m C \xi_i + \sum_{i=1}^m \alpha_i (1 - y_i (\langle w, x_i \rangle + b) - \xi_i) - \sum_{i=1}^m \beta_i \xi_i$$

- (2) Let  $\alpha, \beta$  be fixed. We minimize (9) wrt  $w, b$  and we get

$$(10) \quad 0 = \frac{\partial L}{\partial w}(w, b, \xi, \alpha, \beta) \implies w = \sum_{i=1}^m \alpha_i y_i x_i$$

$$0 = \frac{\partial L}{\partial b}(w, b, \xi, \alpha, \beta) \implies 0 = \sum_{i=1}^m \alpha_i y_i$$

$$(11) \quad 0 = \frac{\partial L}{\partial \xi_i}(w, b, \xi, \alpha, \beta) \implies \alpha_i = C - \beta_i$$

and we substitute (10)-(11) in (9) and we get

$$\tilde{L}(\alpha) = \sum_{i=1}^m \alpha_i - \frac{1}{2} \sum_{i=1}^m \sum_{j=1}^m \alpha_i \alpha_j y_i y_j \langle x_j, x_i \rangle$$

(3) The Karush–Kuhn–Tucker (KKT) conditions applied to the above problem are

$$0 = \nabla \frac{1}{2} \|w\|_2^2 + \nabla \sum_{i=1}^m C \xi_i + \nabla \sum_{i=1}^m \alpha_i (1 - y_i (\langle w, x_i \rangle + b) - \xi_i) - \nabla \sum_{i=1}^m \beta_i \xi_i \quad \text{Stationarity}$$

$$1 - y_i (\langle w, x_i \rangle + b) - \xi_i \leq 0, \quad \forall i = 1, \dots, m \quad \text{Primal feasibility}$$

$$\xi_i \geq 0$$

(12)

$$\alpha_i \geq 0 \quad \forall i = 1, \dots, m \quad \text{Dual feasibility}$$

(13)

$$\beta_i \geq 0 \quad \forall i = 1, \dots, m$$

(14)

$$\alpha_i (1 - y_i (\langle w, x_i \rangle + b) - \xi_i) = 0, \quad \forall i = 1, \dots, m \quad \text{Complementary slackness}$$

(15)

$$\beta_i \xi_i = 0, \quad \forall i = 1, \dots, m$$

(4) It is

$$(16) \quad \alpha^* = \arg \max_{\alpha \in \mathbb{R}^m: \alpha \geq 0} \left( \sum_{i=1}^m \alpha_i - \frac{1}{2} \sum_{i=1}^m \sum_{j=1}^m \alpha_i \alpha_j y_i y_j \langle x_j, x_i \rangle \right)$$

$$\text{subject to } 0 = \sum_{i=1}^m \alpha_i y_i$$

(17)

$$\alpha_i \in [0, C] \quad \forall i = 1, \dots, m$$

constrain (17) results from (11), (13), and (12).

(5)

- By (10), if  $\alpha_i = 0$  then  $x_i$  does not contribute to the computation of the weights.
- By (10), if  $\alpha_i \neq 0$ , then  $x_i$  is a support vector and contributes.
- If  $\alpha_i \in (0, C)$  (where  $\alpha_i \neq C$ ) then (11) implies that  $\beta_i > 0$ . By (15) if  $\beta_i > 0$  then  $\xi_i = 0$ . Hence, given these, from (14), it is  $1 = y_i (\langle w, x_i \rangle + b)$  i.e.  $x_i$  lies on the boundary.
- If  $\alpha_i = C$ , then  $x_i$  lies inside the boundary.

(6) From (14), it is either  $\alpha_i = 0$  or  $(1 - y_i (\langle w, x_i \rangle + b) - \xi_i) = 0$ . Let  $\mathcal{S} = \{i : y_i (\langle w, x_i \rangle + b) = 1 - \xi_i\}$ . From (10), it is

$$(18) \quad w^* = \sum_{i \in \mathcal{S}} \alpha_i^* y_i x_i$$

Using (14) and summing up indexes in  $\mathcal{M} = \{i : \alpha_i \in (0, C)\}$  for which  $\xi_i = 0$  it is

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



(7) The formula is

$$\begin{aligned} h_{w,b}(x) &= \text{sign}(\langle w^*, x \rangle + b^*) \\ (19) \quad &= \text{sign}\left(\sum_{i=1}^m \alpha_i^* y_i \langle x_i, x \rangle + b^*\right) \end{aligned}$$


---

**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  $e^{ix} = \cos(x) + i \sin(x)$

**Solution.** It is

$$\begin{aligned} K(x, y) &= \frac{\sin\left(2\pi\left(N + \frac{1}{2}\right)(x - y)\right)}{\sin(\pi(x - y))} = \frac{-2i \sin\left(2\pi\left(N + \frac{1}{2}\right)(x - y)\right)}{-2i \sin(\pi(x - y))} \\ &= \frac{e^{-2\pi\left(N + \frac{1}{2}\right)i(x-y)} - e^{2\pi\left(N + \frac{1}{2}\right)i(x-y)}}{e^{-\pi i(x-y)} - e^{\pi i(x-y)}} \\ &= \frac{e^{\pi i(x-y)} e^{-2\pi\left(N + \frac{1}{2}\right)i(x-y)} - e^{2\pi\left(N + \frac{1}{2}\right)i(x-y)}}{e^{\pi i(x-y)} e^{-\pi i(x-y)} - e^{\pi i(x-y)} e^{\pi i(x-y)}} \\ &= e^{-2\pi i N(x-y)} \frac{1 - (e^{2\pi i(x-y)})^{2N+1}}{1 - e^{2\pi i(x-y)}} \\ &= e^{-2\pi i N(x-y)} \sum_{n=0}^{2N} \left(e^{2\pi i(x-y)}\right)^n = \sum_{n=-N}^N e^{2\pi i n(x-y)} = \sum_{n=-N}^N e^{2\pi i n x} e^{2\pi i n y} \\ &= \langle \psi(x), \psi(y) \rangle \end{aligned}$$

with  $\psi(x) = (e^{-2\pi i N x}, e^{-2\pi i(N-1)x}, \dots, 1, \dots, e^{2\pi i(N-1)x}, e^{2\pi i N x})^\top$ . Based on the theorem in the Handout, the Kernel can be expressed as an inner product of a vector of bases, hence it is a valid kernel.

---

**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-2009-09-08.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>
- 

#### 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>
-