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

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

Exercise 4. ($\star$)Show that $f : S \to \mathbb{R}$ 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. ($\star$)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 $\alpha \to 0$.

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

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 $w \in \mathbb{R}^d$,

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

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

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 $w \in \mathcal{H}$. $R_g\left(\cdot\right)$ 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. ($\star$) 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

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

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

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

(3) (continue) it is

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

Exercise 13. ($\star$) 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 17 from "Handout 4: 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 4 in "Handout 4: 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 3 in "Handout 5: Stochastic gradient descent" holds.

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

Exercise 14. ($\star$) [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

$$h_w(x) = \text{sign}\left(w^\top x \right) \tag{1}$$

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

Let the hypothesis class is

$$\mathcal{H} = \left\{ x \to w^\top x : \forall w \in \mathbb{R}^d \right\} \tag{3}$$

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

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

[1]We use standard notation

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

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

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

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

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 (4) is convex.

Hint:: You may use Note 11 from Lecture notes 2: Elements of convex learning problems.

(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 $\mathcal{X} := \left\{x \in \mathbb{R}^d : \left\|x\right\|_2 \leq L\right\}$.

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 $1 - yw_2^\top x \; > \text{or} < 0$ 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 $f : \mathbb{R} \to \mathbb{R}_+$ 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

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

The formulas in your algorithm shoud be implemented for the above learning problem and tailored to 1, 3, and 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 $\left\{w^{(t)}\right\}$ against the iteration t.

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

Exercise 15. ($\star$) 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

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

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

Exercise 16. ($\star\star$) Consider a training data set $\mathcal{S} = \{z_i = (x_i, y_i)\}_{i=1}^m$. 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:

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

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

$$\xi_i \geq 0, \ \forall i = 1, ..., m \tag{8}$$

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 $\mathcal{S}$.

(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 can lie 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 in the 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 17. ($\star\star\star$) *[This is the Relevance Vector Machine. The Exercise is taken from "Exercise Sheet: Bayesian Statistics" of the module "Bayesian Statistics III/IV (MATH3361/4071)" taught in "Michaelmas term 2021". The supplementary material in the box was mainly provided for the*

students who had not been introduced to the SVM ideas or the Kernel trick -so it can be skipped. Also, the supplementary material in the box is presented with a statistical (geostatistical modeling) motivation. The exercise requires basic knowledge of Bayesian statistical inference and in particular the use of Bayes theorem for the computation of the posterior as well as basis probability density calculus. However, the exercise is a useful example of extending the SVM ideas to the Bayesian learning setting.]

Regarding the statistical model: Long story, short (supplementary material)

Consider that we are interested in recovering the mapping

$$x \overset{\eta}{\longmapsto} \eta(x)$$

in the sense that $y \in \mathbb{R}$ is the response (output quantity) that depends on $x = (x_1, ..., x_d) \in \mathcal{X} \subseteq \mathbb{R}^d$ which is the independent variable (input quantity) in a procedure; E.g.:,

- y: precipitation in log scale
- $x = (\text{longitude}, \text{latitude})$: geographical coordinates.

Consider a set of observed data $\{(y_i, x_i)\}_{i=1}^n$, which may be contaminated by additive noise of unknown variance; i.e.

$$y_i = \eta(x_i) + \epsilon_i,$$

where $\epsilon_i \overset{\text{IID}}{\sim} \mathrm{N}\left(0, \sigma^2\right)$ and $\sigma^2 > 0$ is unknown. We wish to recover $\eta(x)$ by using the Tikhonov regularization on the functional space $\mathcal{H}$ such that

$$\eta = \arg\min_{\forall \tilde{\eta} \in \mathcal{H}} \left\{ \sum_{i=1}^{n} L\left(y_i - \tilde{\eta}\left(x_i\right)\right) + \lambda \left\|\tilde{\eta}\right\|_{\mathcal{H}}^2 \right\} \tag{9}$$

By assuming that $\mathcal{H}$ is a Reproducing Kernel Hilbert Space (RKHS), the solution to the above Ridge regularizes loss minimization problem is such that

$$\eta\left(x\right) = \beta_0 + \sum_{j=1}^{n} k(x, x_j)\beta_j = k(x)^\top \beta$$

where $k(x) = \left(1, k\left(x, x_1\right), ..., k\left(x, x_n\right)\right)^\top$, $k\left(x, x_j\right)$ is the reproducing kernel (such as $k_\phi\left(x, x_j\right) = \exp\left(-\phi \left\|x - x_j\right\|^2\right)$ for some known parameter $\phi > 0$), and $\beta \in \mathbb{R}^{n+1}$ is an unknown vector.

Consider the following Bayesian model[2]

$$\begin{cases} y|\beta, \sigma^2 & \sim \mathrm{N}\left(K\beta, I\sigma^2\right) \\ \beta|\lambda & \sim \mathrm{N}\left(0, D^{-1}\right), \quad D = (\lambda_0, \lambda_1, ..., \lambda_n) \\ \lambda_i & \overset{\text{iid}}{\sim} \mathrm{d}\Pi\left(\lambda_i\right) \propto \lambda_i^{a-1} \exp\left(-b\lambda_i\right) \mathrm{d}\lambda_i, \quad \forall i = 1, ..., n \\ \sigma^2 & \sim \mathrm{d}\Pi\left(\sigma^2\right) \propto \left(\sigma^2\right)^{c-1} \exp\left(-\frac{1}{\sigma^2} d\right) \mathrm{d}\sigma^2 \\ \beta, \sigma^2 & \text{a priori independent} \end{cases}$$

[2]Dixit, A., & Roy, V. (2021). Posterior impropriety of some sparse Bayesian learning models. Statistics & Probability Letters, 171, 109039.

where K is a known matrix with size $n \times (n+1)$ such that

$$K = \begin{bmatrix} 1 & k(x_1, x_1) & \cdots & k(x_1, x_n) \\ \vdots & \vdots & \ddots & \vdots \\ 1 & k(x_n, x_1) & \cdots & k(x_n, x_n) \end{bmatrix}.$$

The quantities $a > 0$, $b > 0$, $c > 0$, $d > 0$, and $\phi > 0$ are considered as fixed.

(1) When $b = 0$, show that a necessary condition for a valid posterior inference is $a \in (-1/2, 0)$ for any choice of prior for τ (i.e. any choice of (c, d)).

(2) Let $P = K\left(K^\top K\right)^{-1} K^\top$. Show that (2a) and (2b) are sufficient conditions for the Bayesian model to lead to a valid posterior inference
   (a) if $a > 0$ and $b > 0$, or
   (b) if $y^\top (I - P)\, y + 2d > 0$ and $c > -\frac{n}{2}$

(3) Does the the improper Uniform prior on the joint $\log(\lambda_i)$ and $\log\left(\sigma^2\right)$, i.e. $\pi\left(\log(\lambda_i), \log\left(\sigma^2\right)\right) \propto 1$, lead to a valid inference?

(4) Does the Jeffreys' prior $\pi(\lambda_i) \propto 1/\lambda_i$ lead to a valid inference?

Hint-1::

$$(y - K\beta)^\top (y - K\beta) + (\beta - \mu)^\top V^{-1} (\beta - \mu) = (\beta - \mu^*)^\top (V^*)^{-1} (\beta - \mu^*) + S^*;$$

$$S^* = \mu^\top V^{-1} \mu - (\mu^*)^\top (V^*)^{-1} (\mu^*) + y^\top y; \qquad V^* = \left(V^{-1} + K^\top K\right)^{-1}; \qquad \mu^* = V^* \left(V^{-1}\mu + K^\top y\right)$$

Hint-2:: Sherman-Morrison-Woodbury formula:

$$(A + UCV)^{-1} = A^{-1} - A^{-1} U \left(C^{-1} + V A^{-1} U\right)^{-1} V A^{-1}$$

Hint-3::

$$-\frac{y^\top y}{2\sigma^2} \leq -\frac{y^\top \left(I\sigma^2 + K D^{-1} K^\top\right)^{-1} y}{2} \leq -\frac{1}{2\sigma^2} y^\top (I - P)\, y$$

where $P = K\left(K^\top K\right)^{-1} K$.

Hint-4:: It is given that $\int_{(0,\infty)} \frac{t^{-(a+1)}}{(\xi+t)^{1/2}} \mathrm{d}t < \infty$ if and only if $a \in (-1/2, 0)$.

Exercise 18. ($\star$) 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-and-Machine-Learning-2006.pdf

The solutions are available from

- https://blackboard.durham.ac.uk/ultra/courses/_44662_1/outline/create/document?id=_1396738_1

Part 4. The kernel trick

Exercise 19. ($\star\star$) Show that the function K defined over $\mathbb{R} \times \mathbb{R}$ with expression

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

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 20. ($\star\star$) (Kernel ridge regression) Consider the standard linear regression problem with learning rule $h_\theta(x) = \theta^\top x$. Consider a training data set $\mathcal{S} = \{z_i = (x_i, y_i)\}_{i=1}^m$. The ridge regression cost function is then

$$\hat{R}_{\mathcal{S}}(\theta) = \frac{1}{2} \sum_{i=1}^{m} \left(\theta^\top x_i - y_i\right)^2 + \frac{\lambda}{2} \|\theta\|_2^2$$

(1) Use a vector notation to find a closed-form expression for the value of θ which minimizes the ridge regression cost function.

(2) Suppose that we want to use kernels to implicitly represent our feature vectors in a high-dimensional (possibly infinite dimensional) space. Using a feature mapping ψ, the ridge regression cost function becomes

$$\hat{R}_{\mathcal{S}}^{\psi}(\theta) = \frac{1}{2} \sum_{i=1}^{m} \left(\theta^\top \psi(x_i) - y_i\right)^2 + \frac{\lambda}{2} \|\theta\|_2^2$$

Making a prediction on a new input x_{new} would now be done by computing $\theta^\top \psi(x_{\text{new}})$. Show how we can use the "kernel trick" to obtain a closed form for the prediction on the new input x_{new} without ever explicitly computing $\psi(x_{\text{new}})$. You may assume that the parameter vector θ can be expressed as a linear combination of the input feature vectors; i.e., $\theta = \sum_{i=1}^m \alpha_i \psi(x_i)$ for some set of parameters $\{\alpha_i\}$.

Hint:: You may need to use the identity

$$(\lambda I + BA)^{-1} B = B(\lambda I + AB)^{-1}.$$

Exercise 21. ($\star\star$) (On the SVM with Gaussian kernel) Consider the task of training a support vector machine using the Gaussian kernel

$$k(a, b) = \exp\left(-\frac{1}{\phi^2} \|a - b\|_2^2\right)$$

We will show that as long as there are no two identical points in the training set, we can always find a value for the scale parameter $\phi > 0$ such that the SVM achieves zero training error. Consider a training data set $\mathcal{S} = \{z_i = (x_i, y_i)\}_{i=1}^m$.

(1) Recall from class that the decision function learned by the support vector machine can be written as

$$h_w(x) = \sum_{i=1}^{m} \alpha_i y_i k(x_i, x) + b$$

Assume that the training data $\mathcal{S} = \{z_i = (x_i, y_i)\}_{i=1}^m$ consists of points which are separated by at least a distance of ϵ that is $\|x_i - x_j\|_2 \geq \epsilon$ for any $i \neq j$. Find values for the set of parameters $\{\alpha_i\}$ and b and Gaussian kernel scale parameter ϕ such that x_i is correctly classified, for all $i = 1, ..., m$.

Hint:: Let $\alpha_i = 1$ for all i and $b = 0$. Now notice that for $y \in \{-1, +1\}$ the prediction on x_i will be correct if $|f(x_i) - y_i| < 1$, so find a value of ϕ that satisfies this inequality for all i.

(2) (Requires additional reading) Suppose we run a SVM with slack variables using the parameter $\phi > 0$ you found in the previous part. Consider that you are using the "Sequential minimal optimization" to solve the quadratic programming (QP) problem that arises during the training of support-vector machines (SVM) –for details see:

`https://en.wikipedia.org/wiki/Sequential_minimal_optimization`

Will the resulting classifier necessarily obtain zero training error? Why or why not? A short explanation (without proof) will suffice.

Exercise 22. (⋆⋆) (Kernel Principal Component Analysis) We will kernelize the classical PCA.

Classical Principal Component Analysis (PCA)

If x is a random vector with mean μ and covariance matrix Σ then the principal component transformation is the transformation

$$x \mapsto y = \Gamma^\top (x - \mu),$$

where Γ is orthogonal, $\Gamma^\top \Sigma \Gamma = \Lambda$ is diagonal $\Lambda = \text{diag}(\lambda_1, ..., \lambda_p)$, $\lambda_1 \geq ... \geq \lambda_j \geq \lambda_{j+1} \geq ... \geq \lambda_p = 0$. The i-th principal component of x may be defined as the i-th element of the vector y, i.e.

$$y_i = \gamma_{(i)}^\top (x - \mu)$$

where $\gamma_{(i)}$ is the i-th column of Γ and it is called the i-th vector of principal component loadings. In other words, Γ and Λ contain the eigenvectors and eigenvalues from the eigen-decomposition of Σ.

Consider the classical PCA. Consider a given dataset in the form of a matrix X such as $[X]_{i,j} = x_{i,j}$ for $i = 1, ..., m$ and $j = 1, ..., p$. Let $x_i = (x_{i,1}, ..., x_{i,p})^\top$ for $i = 1, ..., m$. Assume the data are centered around zero as $\sum_{i=1}^m x_i = 0$. Hence, $\mu = \text{E}(x) = 0$. The sample covariance matrix is

$$\Sigma = \frac{1}{m} \sum_{i=1}^{m} x_i x_i^\top$$

the ith PC is $y_i = \gamma_{(i)} x$ where $\gamma_{(i)}$ is the i-th column of Γ where $\Gamma^\top \Sigma \Gamma = \Lambda$ is diagonal $\Lambda = \text{diag}(\lambda_1, ..., \lambda_p)$, $\lambda_1 \geq ... \geq \lambda_j \geq \lambda_{j+1} \geq ... \geq \lambda_p = 0$.

Suppose that we want to use kernels to implicitly represent our feature vectors in a high-dimensional (possibly infinite dimensional) space. Using a feature mapping $x \in \mathbb{R}^p \to \psi(x) \in \mathbb{H}$, the sample covariance matrix of the images, after mapping into $\mathbb{H}$ and assuming centered data, is given by

$$C = \frac{1}{m} \sum_{i=1}^{m} \psi(x_i) (\psi(x_i))^\top$$

Keep on assuming that the transformation keeps our data centered around zero; $\mathrm{E}(\psi(x)) = 0$. Show how we can use the "kernel trick" to obtain a closed form for the principal component loadings, and the principal components without ever explicitly computing $\psi(\cdot)$.

Hint:: You may follow the steps:

(1) Show that for $n = 1, ..., m$ it is

$$\frac{1}{m} \sum_{j} \alpha_j \sum_{i=1}^{m} \left[(\psi(x_n))^\top \psi(x_i) \right] \left[(\psi(x_i))^\top \psi(x_j) \right] = \lambda \sum_{j=1}^{m} \alpha_j \left[(\psi(x_n))^\top \psi(x_j) \right]$$

(2) Show that this implies $K\alpha_k = m\lambda_k \alpha_k$ for some function $k(a, b) = (\psi(a))^\top \psi(b)$ and certain vectors α_k.

(3) Normalise α_k's properly

(4) Compute the principal components y_k

Exercise 23. ($\star$)Show that the hinge loss function (5) in the Lecture notes 10 : Multi-class classification, namely

$$\ell_{\text{hinge}}\left(h_w,(x,y)\right)=\max_{\xi\in\mathcal{Y}}\left(\ell_{0-1}\left(\xi,(x,y)\right)+\langle w,\Psi\left(x,\xi\right)-\Psi\left(x,y\right)\rangle\right)$$

is $\rho-$Lipschitz w.r.t. w with $\rho=\max_{\xi\in\mathcal{Y}}\left(\Psi\left(x,\xi\right)-\Psi\left(x,y\right)\right)$.

Hint: You may use as given that

$$\left|\max_{1\leq i\leq N}\left(a_i\right)-\max_{1\leq j\leq N}\left(b_j\right)\right|\leq\max_{1\leq i\leq N}\left|a_i-b_i\right|$$

Exercise 24. ($\star$)Consider the regression problem, with a predictive rule $h_w : \mathbb{R}^d \to \mathbb{R}$, as a classification probability , that receives values $x \in \mathbb{R}^d$ returns vales 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 $\Phi(x) = \int_{-\infty}^{x} \phi(t)\,\mathrm{d}t$ is considered as known function, and $\phi(t) = \frac{1}{\sqrt{2\pi}}\exp\left(-\frac{1}{2}t^2\right)$ and

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

Consider a loss

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

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 $\nabla_w \ell(w, (x, y))$.

The following is given as a homework (Formative assessment 4)

Exercise 25. ($\star$)Consider the multi-class classification problem, with a predictive rule $h_w : \mathbb{R}^d \to \mathcal{P}$, 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 $h_w = (h_{w,1}, ..., h_{w,q})^\top$, 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 $k = 1, ..., q$, and let the associated activation functions be

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

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

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

at w and example $z = (x, y)$, where $x \in \mathbb{R}^d$ is the input vector (features), and $y = (y_1, ..., y_q)$ 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

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

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

Exercise 26. ($\star$) 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

  The solutions are available from
  - https://blackboard.durham.ac.uk/ultra/courses/_44662_1/outline/create/document?id=_1396738_

Exercise 27. ($\star$) Students are encouraged to practice on the Exercises chapter 4, 5, and 8 from the textbook

- Bishop, C. M., & Bishop, H. (2023). Deep learning: Foundations and concepts. Springer Nature.
  available from
  - https://www.bishopbook.com/

  The solutions are available from
  - https://www.bishopbook.com/Concepts_in_Deep_Learning_Solutions_v1.0.pdf

---

Part 7. Gaussian process regression

Exercise 28. ($\star$) 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

  The solutions are available from
  - https://blackboard.durham.ac.uk/ultra/courses/_44662_1/outline/create/document?id=_1396738_

Part 8. **Revision**

The following has been used as Question 2 (Section A) in the written examination 2021 May

Exercise 29. ($\star$)Consider a prediction rule $h : \mathbb{R}^d \to \mathbb{R}_+^q$ with $h(x) = (h_1(x), ..., h_q(x))^\top$ which receives inputs $x = (x_1, ..., x_d)^\top \in \mathbb{R}^d$ and which is modeled as a feedforward neural network (NN) with equation

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

for $k = 1, ..., q$. We consider activation functions $\sigma_1(a) = \frac{1}{1+\exp(-a)}$ and $\sigma_2(a) = \log(1 + \exp(a))$. Parameters $c \in \mathbb{N}_+$, and $d \in \mathbb{N}_+$ are considered as known, while the weights $\{w_{\cdot,\cdot,\cdot}\}$ of the NN are unknown. To learn the unknown weights $\{w_{\cdot,\cdot,\cdot}\}$, we specify the loss function

$$\ell(w, z = (x, y)) = \frac{1}{2}\|h(x) - y\|_2^2 = \frac{1}{2}\sum_{k=1}^{q}(h_k(x) - y_k)^2$$

where $z = (x, y)$ denotes an example, $x \in \mathbb{R}^d$ is the input vector (features), and $y = (y_1, ..., y_q)^\top \in \mathbb{R}^q$ is the output vector (targets).

(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 gradient

$$\nabla_w \ell(w, (x, y)) = \left(\left(\frac{\partial}{\partial w_{1,j,i}}\ell(w, (x, y))\right)_{j=1,i=1}^{c,d}, \left(\frac{\partial}{\partial w_{2,k,j}}\ell(w, (x, y))\right)_{k=1,j=1}^{q,c}\right)$$

of the loss function $\ell(w, z)$ with respect to w for any example $z = (x, y)$. Clearly state the steps of the procedure as well as state the quantities

$$\frac{\partial}{\partial w_{1,j,i}}\ell(w, (x, y)), \text{ and } \frac{\partial}{\partial w_{2,k,j}}\ell(w, (x, y))$$

for all $k = 1, ..., q$, $j = 1, ..., c$, and $i = 1, ..., d$.

The following has been used as Question 4 (Section B) in the written examination 2021 May

Exercise 30. ($\star$)Consider the binary classification learning problem: Let the set of targets be $\mathcal{Y} = \{-1, +1\}$, let the set of inputs be $\mathcal{X} = \left\{x \in \mathbb{R}^d : \|x\|_2 \leq B\right\}$ for some scalar $B > 0$, let the prediction rule be $h_w(x) = x^\top w$, and let the loss function ℓ be

$$\ell(w, z = (x, y)) = \log\left(1 + \exp\left(-yx^\top w\right)\right),$$

for $x \in \mathcal{X}$, $y \in \mathcal{Y}$, and $w \in \mathcal{W}$ where $\mathcal{W} = \left\{w \in \mathbb{R}^d : \|w\|_2 \leq B\right\}$.

(1) Show that the resulting learning problem is convex-Lipschitz-bounded. Specify the parameter of Lipschitzness.

(2) Show that the above loss $\ell(w, z = (x, y))$ is $B^2/4$ smooth.

Hint:: You may use the mean value theorem which states that (under pre-assumed conditions), $f(b) - f(a) = \frac{\mathrm{d}}{\mathrm{d}x} f(x)\big|_{x=c} (b - a)$ for $a \leq c \leq b$.

(3) Consider a risk function $R_g(w) = \mathrm{E}_{z\sim g}(\ell(w, z = (x, y)))$ where g denotes the unknown data generating process. Assume there is a set of available examples $\mathcal{D} = \{z_i = (x_i, y_i); i = 1, ..., n\}$. Now assume that $w \in \mathbb{R}^d$. To learn w, we aim to compute $w^* \in \mathbb{R}^d$ such that

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

where $f(w) = R_g(w) + \frac{\lambda}{2} \|w\|_2^2$.

(a) Show that the stochastic gradient descent algorithm with batch size one and with learning rate

$$\eta_t = \frac{1}{\lambda t}$$

at iteration $t \in \mathbb{N}_+$ which is used to address the learning problem under consideration has a recursion that can be written in the form

$$w^{(t+1)} = -\frac{1}{\lambda t} \sum_{j=1}^{t} v_j$$

where $\{v_j\}$ is the gradient of the loss function at certain values of w and example. Show your working.

(b) Compute the exact formula of v_j as a function of λ, t, $\mathcal{D}$, and $\left\{w^{(t)}\right\}$. Show your working.
