COMP0078: Supervised Learning **2023/24**

Coursework - ⟨2⟩

Due Date: ⟨**10/11/23**⟩ *Student(s):* ⟨*23189938, 23194062*⟩

1 Rademacher Complexity of finite Spaces

We will first show an intermediate result: for any collection $X_1, \ldots, X_m$ of centered random variables (namely $\mathbb{E}X_i = 0$ for all $i = 1, \ldots, m$) taking values in $[a, b] \subset \Re$, we will show that

$$\mathbb{E} \max_{i=1,\ldots,m} X_i \leq \frac{b-a}{2}\sqrt{2\log(m)}$$

1.1 [2 marks]. Let $\bar{X} = \max_i X_i$. Show that for any $\lambda > 0$

$$\mathbb{E}\bar{X} \leq \frac{1}{\lambda}\log \mathbb{E}e^{\lambda \bar{X}}$$

Solution: Where $\lambda > 0$, we bound $\mathbb{E}(\lambda\bar{X})$ using Jensen's inequality and then divide by λ to get our solution.

$$\mathbb{E}\lambda\bar{X} = \log e^{\mathbb{E}\lambda\bar{X}} \leq \log \mathbb{E}e^{\lambda\bar{X}}$$
$$\mathbb{E}\bar{X} \leq \frac{1}{\lambda}\log \mathbb{E}e^{\lambda\bar{X}}$$

1.2 [5 marks]. Show that

$$\frac{1}{\lambda}\log \mathbb{E}e^{\lambda\bar{X}} \leq \frac{1}{\lambda}\log m + \lambda\frac{(b-a)^2}{8}$$

Hint: use Hoeffding's Lemma: for any random variable X such that $X - \mathbb{E}X \in [a, b]$ with $a, b \in \Re$, and for any $\lambda > 0$, we have

$$\mathbb{E}e^{\lambda(X-\mathbb{E}X)} \leq e^{\lambda^2(b-a)^2/8}$$

Solution: We first show that $\mathbb{E}e^{\lambda\bar{X}} \leq \sum_{n=1}^{m} \mathbb{E}e^{\lambda X_i}$,

$$\begin{aligned}\mathbb{E}e^{\lambda\bar{X}} &= \mathbb{E}\left(\max_{i=1,\ldots,m} e^{\lambda\bar{X}}\right) \\ &\leq \mathbb{E}\left(\sum_{i=1}^{m} e^{\lambda X_i}\right) \\ &= \sum_{i=1}^{m} \mathbb{E}\left(e^{\lambda X_i}\right)\end{aligned}$$

The first line is because $\lambda > 0$ and the exponential function is strictly increasing, so we can rearrange this random variable. The second inequality is due to the fact that $e^{\lambda X_i} > 0$, which means that

$\max_{i=1,\dots,m} e^{\lambda X_i} \leq \sum_{n=1}^{m} e^{\lambda X_i}$. The third line is due to linearity of expectation. Now because $\mathbb{E}X_i = 0$ and $X_i \in [a, b]$ for all i, we can use Hoeffding's Lemma on this expression.

$$\begin{aligned}\sum_{i=1}^{m} \mathbb{E}\left(e^{\lambda X_i}\right) &= \sum_{i=1}^{m} \mathbb{E}\left(e^{\lambda(X_i - \mathbb{E}X_i)}\right) \\ &\leq \sum_{i=1}^{m} e^{\lambda^2 (b-a)^2/8} \\ &= m e^{\lambda^2 (b-a)^2/8}\end{aligned}$$

Finally we log both sides, and divide by λ ($\lambda > 0$) to get our final expression.

$$\begin{aligned}\mathbb{E}e^{\lambda \bar{X}} &= m e^{\lambda^2 (b-a)^2/8} \\ \log \mathbb{E}e^{\lambda \bar{X}} &= \log m + \lambda^2 \frac{(b-a)^2}{8} \\ \frac{1}{\lambda} \log \mathbb{E}e^{\lambda \bar{X}} &= \frac{1}{\lambda} \log m + \lambda \frac{(b-a)^2}{8}\end{aligned}$$

1.3 [3 marks]. Conclude that by choosing λ appropriately,

$$\mathbb{E} \max_{i=1,\dots,m} X_i \leq \frac{b-a}{2} \sqrt{2 \log(m)}$$

Solution: Using 1.1 and 1.2, and asuming that $b - a > 0$ we find the bound

$$\mathbb{E}\bar{X} \leq \frac{1}{\lambda} \log \mathbb{E}e^{\lambda \bar{X}} \leq \frac{1}{\lambda} \log m + \lambda \frac{(b-a)^2}{8}.$$

We therefore try to find a $\lambda > 0$ such that

$$\frac{1}{\lambda} \log m + \lambda \frac{(b-a)^2}{8} = \frac{b-a}{2} \sqrt{2 \log(m)}.$$

We re-arrange into a quadratic equation

$$\lambda^2 \frac{(b-a)^2}{8} - \lambda \frac{(b-a)}{2} \sqrt{2 \log m} + \log m = 0,$$

which we hope has 1 solution, which can be found be differentiating and setting to 0

$$\begin{aligned}2\lambda \frac{(b-a)^2}{8} - \frac{(b-a)}{2} \sqrt{2 \log m} &= 0 \\ \lambda = \frac{2}{b-a} \sqrt{2 \log m}\end{aligned}$$

We substitute this into our initial equation to which gives our bound as required. We have found our appropriate λ.

$$\begin{aligned}\frac{1}{\lambda} \log m + \lambda \frac{(b-a)^2}{8} &= \frac{(b-a) \log m}{2\sqrt{2 \log m}} + \frac{(b-a)\sqrt{2 \log m}}{2 \cdot 2} \\ = \frac{b-a}{2} \left(\frac{\log m}{\sqrt{2 \log m}} + \frac{\sqrt{2 \log m}}{2} \right) &= \frac{b-a}{2} \sqrt{2 \log m}.\end{aligned}$$

We are almost ready to provide the bound for the Rademacher complexity of a finite set of hypotheses. Let S a finite set of points in $\Re^n$ with cardinality $|S| = m$. We can define the Rademacher complexity of S similarly to how we have done for the Rademacher complexity of a space of hypotheses:

$$\mathcal{R}(S) = \mathbb{E}_\sigma \max_{x \in S} \frac{1}{n} \sum_{j=1}^{n} \sigma_j x_j$$

With $\sigma_1, \ldots, \sigma_n$ Rademacher variables (independent and uniformly sampled from $\{-1, 1\}$.

1.4 [3 marks]. With $\|\cdot\|_2$ denoting the Euclidean norm, show that

$$\mathcal{R}(S) \leq \max_{x \in S} \|x\|_2 \frac{\sqrt{2\log(m)}}{n}.$$

Solution: With S, $\mathbf{x}$ and σ as defined in the question, let us define

$$X := \frac{1}{n} \sum_{j=1}^{n} \sigma_j x_j,$$

note that due to linearity of expectation and independence of the Rademacher variables

$$\mathbb{E}_\sigma X = \frac{1}{n} \sum_{j=1}^{n} x_j \cdot \mathbb{E}_\sigma \sigma_j = 0,$$

which means we can use the results from 1.1 and 1.3. Using 1.1, we know that

$$\mathbb{E}\mathcal{R}(S) \leq \frac{1}{\lambda} \log \mathbb{E} \exp \lambda \mathcal{R}(S).$$

We will now show a tighter bound than 1.2 using properties of X:

$$\begin{aligned}
\mathbb{E}_\sigma \exp(\lambda \mathcal{R}(S)) &= \mathbb{E}_\sigma \exp\left(\max_{x \in S} \frac{\lambda}{n} \sum_{j=1}^{n} \sigma_j x_j \right) \\
&= \mathbb{E}_\sigma \max_{x \in S} \exp\left(\frac{\lambda}{n} \sum_{j=1}^{n} \sigma_j x_j \right) \\
&\leq \mathbb{E}_\sigma \sum_{k=1}^{m} \exp\left(\frac{\lambda}{n} \sum_{j=1}^{n} \sigma_j x_j^{(k)} \right) \\
&= \sum_{k=1}^{m} \mathbb{E}_\sigma \exp\left(\frac{\lambda}{n} \sum_{j=1}^{n} \sigma_j x_j^{(k)} \right) \\
&= \sum_{k=1}^{m} \mathbb{E}_\sigma \prod_{j=1}^{n} \exp\left(\frac{\lambda}{n} \sigma_j x_j^{(k)} \right) \\
&= \sum_{k=1}^{m} \prod_{j=1}^{n} \mathbb{E}_\sigma \exp\left(\lambda \cdot \frac{1}{n} \sigma_j x_j^{(k)} \right).
\end{aligned}$$

In the first line we used the definition of $\mathcal{R}(S)$, in the second line we used the fact the exponential function is strictly increasing and $\lambda > 0$ which means we can move the max outside of the exponential. In the third line, because the exponential is always positive, we can bound the max by the sum, in the fourth line we have used linearity of expectation. In the fifth, because λ, $x_j^{(k)}$ and n are independent to sigma, and because the sigmas are iid, we can expand the exponential out. Finally, in the last line, because the sigmas are iid, the expectation of the product of independent random variables is the product of their expectations. Now because $\mathbb{E}_\sigma \frac{1}{n}\sigma_j x_j^{(k)} = 0$, and $\mathbb{E}_\sigma \frac{1}{n}\sigma_j x_j^{(k)} - \mathbb{E}_\sigma \frac{1}{n}\sigma_j x_j^{(k)} \in [-\frac{1}{n}|x_j^{(k)}|, \frac{1}{n}|x_j^{(k)}|]$, we can use Hoeffding's Lemma

$$\begin{aligned}\sum_{k=1}^{m}\prod_{j=1}^{n}\mathbb{E}_\sigma \exp\left(\lambda \cdot \frac{1}{n}\sigma_j x_j^{(k)}\right) &\leq \sum_{k=1}^{m}\prod_{j=1}^{n}\exp\left(\frac{\lambda^2}{8}\left(\frac{2}{n}|x_j^{(k)}|\right)^2\right) \\ &= \sum_{k=1}^{m}\exp\left(\frac{\lambda^2}{8}\left(\frac{2}{n}\|x^{(k)}\|_2\right)^2\right) \\ &\leq m\exp\left(\frac{\lambda^2}{8}\left(\frac{2}{n}\max_{x\in S}\|x\|_2\right)^2\right).\end{aligned}$$

Where in the second line we have used the definition of the $L2$ norm, and in the last line we have bounded the sum by m times the maximum. Taking the log and dividing by λ on both sides, we can then use our result from 1.3, which gives our result as required.

$$\begin{aligned}\mathcal{R}(S) &\leq \left(\frac{2}{n}\max_{x\in S}\|x\|_2\right)\frac{\sqrt{2\log m}}{2}. \\ &= \max_{x\in S}\|x\|_2\frac{\sqrt{2\log(m)}}{n}.\end{aligned}$$

1.5 [7 marks]. Let $\mathcal{H}$ be a set of hypotheses $f : \mathcal{X} \to \Re$. Assume $\mathcal{H}$ to have finite cardinality $|\mathcal{H}| < +\infty$. Let $S = (x_i)_{i=1}^n$ be a set of points in $\mathcal{X}$ an input set. Use the reasoning above to prove an upper bound for empirical Rademacher complexity $\mathcal{R}_S(\mathcal{H})$, where the cardinality of $\mathcal{H}$ appears logarithmically.

Solution: Where $|\mathcal{H}| = m$, the empirical Rademacher complexity is defined by

$$\mathcal{R}(S) = \mathbb{E}_\sigma \sup_{f\in\mathcal{H}}\frac{1}{m}\sum_{j=1}^{n}\sigma_j f(x_j).$$

But since $|\mathcal{H}|$ is finite, a maximum value exists, and we can substitute sup for max. We also define $z = (f(x_1), \ldots, f(x_n))$ and $S_f = (z_j)_{j=1}^m$. We can now use our result from 1.4.

$$\begin{aligned}&= \frac{n}{m}\mathbb{E}_\sigma \max_{f\in\mathcal{H}}\frac{1}{n}\sum_{j=1}^{r}\sigma_j f\left(x_j\right) = \frac{n}{m}\mathbb{E}_\sigma \max_{z\in S_f}\frac{1}{n}\sum_{j=1}^{r}\sigma_j z_j \\ &\leq \frac{n}{m}\max_{z\in S_f}\|z\|_2\frac{\sqrt{2\log(m)}}{n} = \max_{z\in S_f}\|z\|_2\frac{\sqrt{2\log(m)}}{m}.\end{aligned}$$

2 Bayes Decision Rule and Surrogate Approaches

In (binary) classification problems the classification or "decision" rule is a binary valued function $c : \mathcal{X} \to \mathcal{Y}$, where $\mathcal{Y} = \{1, -1\}$. The quality of a classification rule can be measured by the misclassification error

$$R(c) = \mathbb{P}_{(x,y)\sim\rho}(c(x) \neq y)$$

assuming to sample an input-output pair (x, y) according to a distribution ρ on $\mathcal{X} \times \mathcal{Y}$.

2.1 [4 marks]. Let $\mathbf{1}_{y=y'}$ be the $0-1$ loss such that $\mathbf{1}_{y=y'} = 1$ if $y \neq y'$ and 0 otherwise. Show that the misclassification error corresponds to the expected risk of the $0-1$ loss, namely

$$R(c) = \int_{\mathcal{X}\times\mathcal{Y}} \mathbf{1}_{c(x)\neq y} d\rho(x, y)$$

Solution: We show this result using simple definitions of expectation, and that $y \in \{-1, 1\}$

$$\begin{aligned} R(c) &= \mathbb{P}_{(x,y)\sim\rho}(c(x) \neq y) \\ &= 1 \cdot \mathbb{P}_{(x,y)\sim\rho}(c(x) \neq y) + 0 \cdot \mathbb{P}_{(x,y)\sim\rho}(c(x) = y) \\ &= \mathbb{E}_{(x,y)\sim\rho}\left(\mathbf{1}_{c(x)\neq y}\right) \\ &= \int_{\mathcal{X}\times\mathcal{Y}} \mathbf{1}_{c(x)\neq y} d\rho(x, y). \end{aligned}$$

(Surrogate Approaches) Since the 0-1 loss is not continuous, it is typically hard to address the learning problem directly and in practice one usually looks for a real valued function $f : \mathcal{X} \to \Re$ solving a so-called surrogate problem

$$\mathcal{E}(f) = \int_{\mathcal{X}\times\mathcal{Y}} \ell(f(x), y) d\rho(x, y)$$

where $\ell : \Re \times \Re \to \Re$ is a "suitable" convex loss function that makes the surrogate learning problem more amenable to computations. Given a function $f : \mathcal{X} \to \Re$, a classification rule $c_f : \mathcal{X} \to \{-1, 1\}$ is given in terms of a "suitable" map $d : \Re \to \{-1, 1\}$ such that $c_f(x) = d(f(x))$ for all $x \in \mathcal{X}$. Here we will look at some surrogate frameworks.

A good surrogate method satisfies the following two properties:

> **(Fisher Consistency)**. Let $f_* : \mathcal{X} \to \Re$ denote the expected risk minimizer for $\mathcal{E}(f_*) = \inf_{f:\mathcal{X}\to\Re} \mathcal{E}(f)$, we say that the surrogate framework is Fisher consistent if
>
> $$R(c_{f_*}) = \inf_{c:\mathcal{X}\to\{-1,1\}} R(c)$$

> **(Comparison Inequality)**. The surrogate framework satisfies as comparison inequality if for any $f : \mathcal{X} \to \Re$
>
> $$R(c_f) - R(c_{f_*}) \leq \sqrt{\mathcal{E}(f) - \mathcal{E}(f_*)}$$

In particular, if we have an algorithm producing estimators f_n for the surrogate problem such that $\mathcal{E}(f_n) \to \mathcal{E}(f_*)$ for $n \to +\infty$, we automatically have $R(c_{f_n}) \to R(c_{f_*})$.

2.2 [4 marks]. (Assuming to know ρ), calculate the closed-form of the minimizer f_* of $\mathcal{E}(f)$ for the:

(a) squared loss $\ell(f(x), y) = (f(x) - y)^2$,

(b) exponential loss $\ell(f(x), y) = \exp(-yf(x))$,

(c) logistic loss $\ell(f(x), y) = \log(1 + \exp(-yf(x)))$,

(d) hinge loss $\ell(f(x), y) = \max(0, 1 - yf(x))$.

(hint: recall that $\rho(x, y) = \rho(y \mid x)\rho_{\mathcal{X}}(x)$ with $\rho_{\mathcal{X}}$ the marginal distribution of ρ on $\mathcal{X}$ and $\rho(y \mid x)$ the corresponding conditional distribution. Write the expected risk as

$$\mathcal{E}(f) = \int_{\mathcal{X}\times\mathcal{Y}} \ell(f(x), y) d\rho(x, y) = \int_{\mathcal{X}} \left(\int_{\mathcal{Y}} \ell(f(x), y) d\rho(y \mid x) \right) d\rho_{\mathcal{X}}(x)$$

you can now solve the problem in the inner integral point-wise $\forall x \in \mathcal{X}$).

Solution: Using the hint, we only need to solve the inner integral point-wise $\forall x \in \mathcal{X}$, we are also in the binary classification setting which means $\mathcal{Y} = \{1, -1\}$.

(a) Given the inner integral, we find the minimiser f^* of

$$\mathcal{E}(f|x) = \int_{\mathcal{Y}} (f(x) - y)^2 \, d\rho(y \mid x),$$

by differentiating with respect to $f(x)$ and setting to 0:

$$\begin{aligned}
\frac{\partial \mathcal{E}(f|x)}{\partial f(x)} = \int_{\mathcal{Y}} 2f^*(x) - 2y \, d\rho(y \mid x) &= 0 \\
\int_{\mathcal{Y}} 2f^*(x) \, d\rho(y \mid x) &= \int_{\mathcal{Y}} 2y \, d\rho(y \mid x) \\
f^*(x) &= \int_{\mathcal{Y}} y \, d\rho(y \mid x) \\
f^*(x) &= \mathbb{E}_Y \left(Y | X = x \right). \\
f^*(x) &= \rho(y = 1|x) - \rho(y = -1|x)
\end{aligned}$$

(b) Given the inner integral, we find the minimiser f^* by differentiating and setting to 0

$$\begin{aligned}
\mathcal{E}(f|x) &= \int_{\mathcal{Y}} \exp(-yf(x)) \, d\rho(y \mid x) \\
&= e^{-f(x)}\rho(y = 1|x) + e^{f(x)}\rho(y = -1|x) \\
\frac{\partial \mathcal{E}(f|x)}{\partial f(x)} &= -e^{-f^*(x)}\rho(y = 1|x) + e^{f^*(x)}\rho(y = -1|x) = 0
\end{aligned}$$

Which we rearrange to get out solution

$$f^*(x) = \frac{1}{2} \log \left(\frac{\rho(y = 1|x)}{\rho(y = -1|x)} \right)$$

(c) Given the inner integral, we find the minimiser f^* by differentiating and setting to 0

$$\begin{aligned}\mathcal{E}(f|x) &= \int_{\mathcal{Y}} \log(1+\exp(-yf(x)))\, d\rho(y \mid x) \\ &= \log(1+e^{-f(x)})\rho(y=1|x) + \log(1+e^{f(x)})\rho(y=-1|x) \\ \frac{\partial \mathcal{E}(f|x)}{\partial f(x)} &= \frac{e^{f^*(x)}}{1+e^{f^*(x)}}\rho(y=-1|x) - \frac{e^{-f^*(x)}}{1+e^{-f^*(x)}}\rho(y=-1|x) = 0\end{aligned}$$

Which we rearrange to get out solution

$$f^*(x) = \log\left(\frac{\rho(y=1|x)}{\rho(y=-1|x)}\right)$$

(d) Given the inner integral, we find the minimiser f^* by constructing the function piecewise, and finding the minimiser

$$\begin{aligned}\mathcal{E}(f|x) &= \int_{\mathcal{Y}} \max(0, 1-yf(x))\, d\rho(y \mid x) \\ &= \max(0,1-f(x))\rho(y=1|x) + \max(0,1+f(x))\rho(y=-1|x) \\ &= \begin{cases} (1-f(x))\rho(y=1|x) & f(x) \le -1 \\ (1-f(x))\rho(y=1|x) + (1+f(x))\rho(y=-1|x) & -1 < f(x) < 1 \\ (1+f(x))\rho(y=-1|x) & 1 \le f(x) \end{cases}\end{aligned}$$

Then as $0 \le \rho(y|x) \le 1$, and the gradient is given by

$$\begin{cases} -\rho(y=1|x) & f(x) \le -1 \\ \rho(y=-1|x) - \rho(y=1|x) & -1 < f(x) < 1 \\ \rho(y=-1|x) & 1 \le f(x) \end{cases}$$

the minimum must be either at $f(x) = -1$ and/or $f(x) = 1$ which means

$$\begin{aligned}f^*(x) &= \arg\min_{f(x)\in\{-1,1\}} \mathcal{E}(f(x)|x) \\ &= \begin{cases} -1 & \rho(y=1|x) \le \rho(y=-1|x) \\ 1 & \text{else} \end{cases} \\ &= 1 - 2\cdot \mathbf{1}_{\rho(y=-1|x)>0.5}\end{aligned}$$

2.3 [4 marks]. The minimizer c_* of $R(c)$ over all possible decision rules $c : \mathcal{X} \to \{-1, 1\}$ is called Bayes decision rule. Write explicitly the Bayes decision rule (again Assuming ρ known a priori).

Solution: By rewriting the misclassification error the same as above, we can minimise $R(c|x)$

$$\begin{aligned}R(c|x) &= \int_{\mathcal{Y}} \mathbf{1}_{c(x)\neq y}\, d\rho(y|x) \\ &= \begin{cases} \rho(y=1|x) & c(x) = -1 \\ \rho(y=-1|x) & c(x) = 1 \end{cases}\end{aligned}$$

Which, for all $x \in \mathcal{X}$, is trivially minimised by

$$c^*(x) = \begin{cases} -1 & \rho(y=1|x) \leq \rho(y=1|x) \\ 1 & \text{else} \end{cases}$$
$$= 1 - 2 \cdot \mathbf{1}_{\rho(y=-1|x)>0.5}$$

2.4 [4 marks]. Are the surrogate frameworks in problem (2.2) Fisher consistent? Namely, can you find a map $d : \Re \to \{-1, 1\}$ such that $R(c_*(x)) = R(d(f_*(x)))$ where f_* is the corresponding minimiser of the surrogate risk $\mathcal{E}$? If it is the case, write d explicitly.

Solution: If we define

$$d := \text{sign}(x) = \begin{cases} -1 & x \leq 0 \\ 1 & x > 0 \end{cases}$$

Then we find that all of the minimisers are Fischer consistent. Because $\rho \in [0, 1]$, all of the minimisers will be positive if $\rho(y = 1|x) > 0.5$, and negative otherwise, meaning $d(f^*) = c^*$ for all of our loss functions. Note for d), we do not have to do anything!.

2.5 [24 marks]. Let $f_* : \mathcal{X} \to \mathbb{R}$ be the minimizer of the expected risk for the surrogate least squares classification problem obtained in problem (2.2). Let sign : $\mathbb{R} \to \{-1, 1\}$ denote the "sign" function

$$\text{sign}(x) = \begin{cases} +1, & x \geq 0 \\ -1, & x < 0 \end{cases}.$$

Prove the following comparison inequality

$$0 \leq R(\text{sign}(f)) - R(\text{sign}(f_*)) \leq \mathcal{E}(f) - \mathcal{E}(f_*),$$

by showing the following intermediate steps:

2.5.1 $|R(\text{sign}(f)) - R(\text{sign}(f_*))| = \int_{\mathcal{X}_f} |f_*(x)| d\rho_{\mathcal{X}}(x)$, where $\mathcal{X}_f = \{x \in \mathcal{X} \mid \text{sign}(f(x)) \neq \text{sign}(f_*(x))\}$.

Solution: Here, I will write $s(x) := \text{sign}(x)$ for abbreviation.

$$\begin{aligned} |R(s(f)) - R(s(f_*))| &= |\mathbb{P}_\rho(s(f(X)) \neq Y) - \mathbb{P}_\rho(s(f_*(X)) \neq Y)| \\ &= \left| \int_{\mathcal{X}\times\mathcal{Y}} \rho(x, y) \left(\mathbb{1}\{s(f(x)) \neq y\} - \mathbb{1}\{s(f_*(x)) \neq y\}\right) dxdy \right| \\ &= \left| \int_{\mathcal{X}_f} dx \int_{\mathcal{Y}} \rho(x, y) \left(\mathbb{1}\{s(f(x)) \neq y\} - \mathbb{1}\{s(f_*(x)) \neq y\}\right) dy \right| \end{aligned}$$

since the integrand is zero if $f_*(x)$ and $f(x)$ have the same sign

$$\begin{aligned} &= \Big| \int_{\mathcal{X}_f} \rho(x, 1) \left(\mathbb{1}\{s(f(x)) \neq 1\} - \mathbb{1}\{s(f_*(x)) \neq 1\}\right) \\ &\qquad + \rho(x, -1) \left(\mathbb{1}\{s(f(x)) \neq -1\} - \mathbb{1}\{s(f_*(x)) \neq -1\}\right) dx \Big| \\ &= \Big| \int_{\mathcal{X}_f} \rho(x, 1) \left(\mathbb{1}\{s(f_*(x)) = 1\} - \mathbb{1}\{s(f_*(x)) \neq 1\}\right) \\ &\qquad + \rho(x, -1) \left(\mathbb{1}\{s(f_*(x)) = -1\} - \mathbb{1}\{s(f_*(x)) \neq -1\}\right) dx \Big| \end{aligned}$$

since $f_*(x)$ and $f(x)$ have different signs in $\mathcal{X}_f$

$$\begin{aligned}
&= \left| \int_{\mathcal{X}_f} \mathbb{1}\{s(f_*(x)) = 1\}(\rho(x,1) - \rho(x,-1)) \right. \\
&\qquad \left. + \mathbb{1}\{s(f_*(x)) = -1\}(\rho(x,-1) - \rho(x,-1))dx \right| \\
&= \left| \int_{\mathcal{X}_f} \mathbb{1}\{f_*(x) \geq 0\} f_*(x)\rho(x) - \mathbb{1}\{f_*(x) < 0\} f_*(x)\rho(x)dx \right|
\end{aligned}$$

since $f_*(x) = \mathbb{E}[Y|X = x] = \rho(Y = 1|x) - \rho(Y = -1|x)$

$$\begin{aligned}
&= \left| \int_{\mathcal{X}_f} |f_*(x)|\rho(x)dx \right| \\
&= \int_{\mathcal{X}_f} |f_*(x)| d\rho_{\mathcal{X}}(x)
\end{aligned}$$

2.5.2 $\int_{\mathcal{X}_f} |f_*(x)| d\rho_{\mathcal{X}}(x) \leq \int_{\mathcal{X}_f} |f_*(x) - f(x)| d\rho_{\mathcal{X}}(x) \leq \sqrt{\mathbb{E}[|f(x) - f_*(x)|^2]}$.

where $\mathbb{E}$ denotes the expectation with respect to $\rho_{\mathcal{X}}$.

Solution:

For the first inequality:

In $\mathcal{X}_f$:
If $f_*(x) \geq 0$, then $f(x) < 0$, so $f_*(x) - f(x) > f_*(x) \geq 0$, and hence $|f_*(x) - f(x)| > |f_*(x)|$.
If $f_*(x) < 0$, then $f(x) \geq 0$, so $f_*(x) - f(x) \leq f_*(x) < 0$, and hence $|f_*(x) - f(x)| > |f_*(x)|$.
Since $|f_*(x) - f(x)| > |f_*(x)| \forall x \in \mathcal{X}_f$, we must have:

$$\int_{\mathcal{X}_f} |f_*(x)| d\rho_{\mathcal{X}}(x) \leq \int_{\mathcal{X}_f} |f_*(x) - f(x)| d\rho_{\mathcal{X}}(x) \tag{1.1}$$

For the second inequality:

$$\begin{aligned}
\int_{\mathcal{X}_f} |f_*(x) - f(x)| d\rho_{\mathcal{X}}(x) &\leq \int_{\mathcal{X}} |f_*(x) - f(x)| d\rho_{\mathcal{X}}(x) \\
&= \mathbb{E}_{\rho_{\mathcal{X}}}[|f(x) - f_*(x)|] \\
&\leq \sqrt{\mathbb{E}[|f(X) - f_*(X)|^2]}
\end{aligned}$$

using the fact that $\sqrt{\mathbb{E}[X^2]} \geq \mathbb{E}[X]$ for $\mathbb{E}[X] \geq 0$ by Jensen's inequality.

2.5.3 $\mathcal{E}(f) - \mathcal{E}(f_*) = \mathbb{E}[|f(X) - f_*(X)|^2]$.

Solution:

$$
\begin{aligned}
\mathcal{E}(f) - \mathcal{E}(f_*) &= \mathbb{E}[(f(X) - Y)^2] - \mathbb{E}[(f_*(X) - Y)^2] \\
&= \int_{\mathcal{X}} \rho_{\mathcal{X}}(x)dx \left(\int_{\mathcal{Y}} \left((f(x) - y)^2 - (f_*(x) - y)^2 \right) \rho_{\mathcal{Y}}(y|x)dy \right) \\
&= \int_{\mathcal{X}_f} \left(\rho(1|x) \left((f(x) - 1)^2 - (f_*(x) - 1)^2 \right) + \rho(-1|x) \left((f(x) + 1)^2 - (f_*(x) + 1)^2 \right) \right) \rho_{\mathcal{X}}(x)dx \\
&= \int_{\mathcal{X}_f} (\rho(1|x)(f(x) + f_*(x) - 2)(f(x) - f_*(x)) \\
&\quad + \rho(-1|x)(f(x) + f_*(x) + 2)(f(x) - f_*(x))\rho_{\mathcal{X}}(x)dx \qquad 2005/0 \\
&= \int_{\mathcal{X}_f} (f(x) - f_*(x)) \left(2(\rho(-1|x) - \rho(1|x)) + (f(x) + f_*(x))(\rho(-1|x) + \rho(1|x)) \right) \rho_{\mathcal{X}}(x)dx \\
&= \int_{\mathcal{X}_f} (f(x) - f_*(x)) \left(-2f_*(x) + f(x) + f_*(x) \right) \rho_{\mathcal{X}}(x)dx \\
&= \int_{\mathcal{X}_f} (f(x) - f_*(x))^2 \rho_{\mathcal{X}}(x)dx \\
&= \mathbb{E}[|f(X) - f_*(X)|^2]
\end{aligned}
$$

3 Kernel Perceptron

The kernel perceptron is an extension of the basic perceptron algorithm which allows the learning of a non-linear decision boundary without giving specific data transformations. My implementation was a one-vs-all method, which forms a classifier for each label. The classifier is required to determine whether a point belongs to that label. Mathematically, for a point $\mathbf{x}_t$, we take $\text{sign}(\sum_i \alpha_i K(\mathbf{x}_i, \mathbf{x}_t))$. A positive sign indicates that $\mathbf{x}_t$ belongs to the classifier's label, and vice versa.

In code, this was represented in two data structures: the alpha matrix and the kernel matrix. Each row of the alpha matrix contains the alpha vector for a classifier, which facilitates efficient computation. This pairs with the kernel matrices. For the dot product and Gaussian kernels, we store the kernel values between each point in the dataset in matrices, since they never change. Then the evaluation of the sum, which just involves taking matrix products between the alpha and kernel matrices, just involves retrieving kernel values.

The dual online learning scenario involves testing the classifier against each point one at a time. If the classifier makes a mistake at a point $\mathbf{x}_t$, then its alpha vector gets updated at the point α_t, depending on the way in which it was wrong.

3.1 Basic results

The first experiment involved measuring the train and test error of an online kernel perceptron using a polynomial dot product kernel. We vary the power on the dot product. The number of epochs in this experiment, and in all following experiments, is 3, based on empirical evidence: here, both the training and testing error failed to decrease; the classifier demonstrated convergence.

Power	Train Error (mean ± s.d.)	Test Error (mean ± s.d.)
1	0.07043 ± 0.00213	0.09890 ± 0.01192
2	0.01579 ± 0.00085	0.04269 ± 0.00780
3	0.00608 ± 0.00071	0.03457 ± 0.00483
4	0.00362 ± 0.00055	0.03207 ± 0.00389
5	0.00232 ± 0.00062	0.02844 ± 0.00342
6	0.00204 ± 0.00051	0.03038 ± 0.00351
7	0.00168 ± 0.00054	0.02874 ± 0.00445

The table shows the means and standard deviations of the train and test error, calculated over 20 runs. We see a steady decrease in training error and testing error as the exponent increased. A higher degree polynomial allows for a more flexible decision boundary due to considering higher order interactions.

3.2 Cross-validation

In this case, we performed 5 fold cross-validation across a training set to obtain the best value of d. Then we used that value of d to train on the whole dataset. This was averaged over 20 runs. The following are the means and standard deviations for the test error and the best value of d:

Test error: 0.028924 ± 0.0043796

$d^* : 6.1 \pm 0.8306$

Confusion matrix; (a, b) is the rate of misclassifying a as b during testing.

In units of 10^{-3}, to 2 d.p:

	0	1	2	3	4
0	0	0.77 ± 1.34	2.99 ± 2.90	1.61 ± 2.15	1.14 ± 1.56
1	0	0	0.80 ± 1.60	0.20 ± 0.89	3.35 ± 4.15
2	3.39 ± 4.23	2.27 ± 3.98	0	5.16 ± 5.07	7.08 ± 5.88
3	3.01 ± 4.47	1.23 ± 3.13	6.05 ± 4.74	0	1.83 ± 2.80
4	0.57 ± 1.72	3.90 ± 4.40	5.04 ± 5.27	0.85 ± 2.04	0
5	7.53 ± 7.01	1.06 ± 2.53	2.42 ± 3.30	11.86 ± 9.08	6.79 ± 5.14
6	4.87 ± 7.17	2.70 ± 3.53	2.67 ± 3.52	0	4.22 ± 4.71
7	0.95 ± 2.27	2.61 ± 6.11	3.93 ± 6.95	1.29 ± 2.60	7.18 ± 7.59
8	10.11 ± 9.66	5.62 ± 7.15	7.22 ± 6.73	11.76 ± 7.76	4.98 ± 5.55
9	1.47 ± 3.15	0.28 ± 1.22	1.79 ± 3.39	0.58 ± 1.74	10.18 ± 9.32

	5	6	7	8	9
0	2.01 ± 2.95	2.24 ± 2.71	0.65 ± 1.30	0.49 ± 1.18	0.79 ± 1.38
1	0.20 ± 0.86	1.98 ± 2.68	1.23 ± 2.27	1.79 ± 2.35	0.42 ± 1.82
2	1.63 ± 3.50	2.13 ± 3.57	4.70 ± 5.28	4.01 ± 5.02	0.23 ± 1.02
3	16.56 ± 8.82	1.20 ± 2.42	3.37 ± 4.95	8.50 ± 5.87	1.24 ± 2.48
4	2.10 ± 2.86	6.06 ± 6.38	2.63 ± 4.31	0.68 ± 2.04	6.78 ± 6.56
5	0	6.91 ± 5.83	1.37 ± 3.44	5.17 ± 5.34	3.06 ± 3.97
6	3.29 ± 4.78	0	0	3.86 ± 4.70	0.06 ± 0.18
7	1.90 ± 2.91	0	0	3.39 ± 3.60	13.24 ± 11.68
8	13.24 ± 10.68	3.96 ± 6.93	8.11 ± 8.77	0	3.80 ± 5.54
9	0.59 ± 4.37	0.33 ± 1.15	18.35 ± 9.27	4.25 ± 3.99	0

3.3 Hardest images to predict

To fairly measure which images are the hardest to predict, we must take care to test each image the same number of times. One way to achieve this is to use the cross-validation method of dataset splitting. To test every image in the set once, we simply run a k-fold classification test, as in question 2, and keep track of which image was predicted incorrectly. The two parameters left to choose are the values of d, and the number of runs to complete.

After running the experiment to obtain the best values of d, $5, 6, 7$ were the most commonly returned values. Hence we do 4 runs for each of these, using 3 epochs.

3.4 Gaussian kernel

To select a sensible range of values for c to cross-validate with, we first test a range of possible values. For each candidate value, we conduct 5 random 80-20 train-test splits of the data, then train the perceptron. The mean errors are seen in the table below.

Candidate	Test Error
-2.000	0.84140
-1.000	0.96828
-0.500	0.96183
-0.100	0.97097
0.001	0.07849
0.005	0.03978
0.010	0.03871
0.025	0.03280
0.050	0.05108
0.075	0.04677
0.100	0.05430
0.150	0.05376
0.200	0.06129
0.300	0.07849
0.500	0.05753
1.000	0.07366
3.000	0.08538
10.000	0.19516

We minimise our test error at $c = 0.025$, and judging by the surrounding values, we pick the range of values for c: $\{0.005, 0.010, 0.015, 0.020, 0.025, 0.030, 0.035\}$. The (unseen) hyper-parameter we pick here is the number of candidates we cross-validate over. Since increasing the number of candidates (specifically adding other candidates to the options) is guaranteed to do at least as well as the current number in terms of best train and test error, it seems sensible to adopt the same number as we had in previous settings for the sake of comparison.

For Q1, the results are:

c	Train Error (mean ± s.d.)	Test Error (mean ± s.d.)
0.005	0.07565 ± 0.00199	0.06005 ± 0.01064
0.010	0.06006 ± 0.00156	0.04363 ± 0.00537
0.015	0.05429 ± 0.00120	0.03820 ± 0.00472
0.020	0.05391 ± 0.00130	0.03798 ± 0.00275
0.025	0.05541 ± 0.00170	0.03798 ± 0.00481
0.030	0.05770 ± 0.00166	0.04148 ± 0.00470
0.035	0.05985 ± 0.00137	0.04234 ± 0.00452

For Q2, the results are:

Test error: