COMP0078 Supervised Learning Coursework 1

PART II

2.1 Generating the data

6.

Here is the plot for a hypothesis $h_{S,v}$ visualized with $|S| = 100$ and $v = 3$.

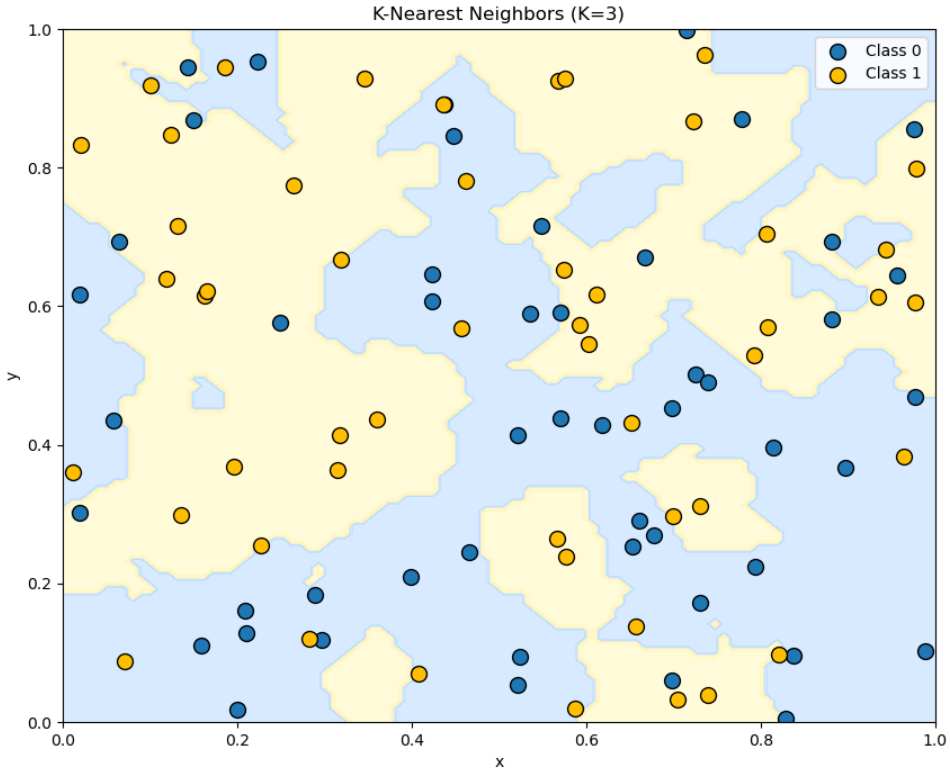

Figure 1: A hypothesis $h_{S,v}$ visualized with $|S| = 100$ and $v = 3$.

2.2 Generating the data

7.

In Protocol A, we sampled the data from 3-NN model and we predicted it with various k values. The error is high for $k = 1$ and $k = 2$ because the model overfits the data, capturing noise rather than the underlying structure. At $k = 3$, the error decreases significantly, as this k-value matches the sampling configuration, resulting in a noticeable improvement.

As k increases, the error continues to decrease, reaching a minimum at $k = 9$ in this case, where the model achieves the optimal balance between bias and variance. However, as k grows further, the error rises again because the inclusion of too many neighbors that leads to underfitting of data.

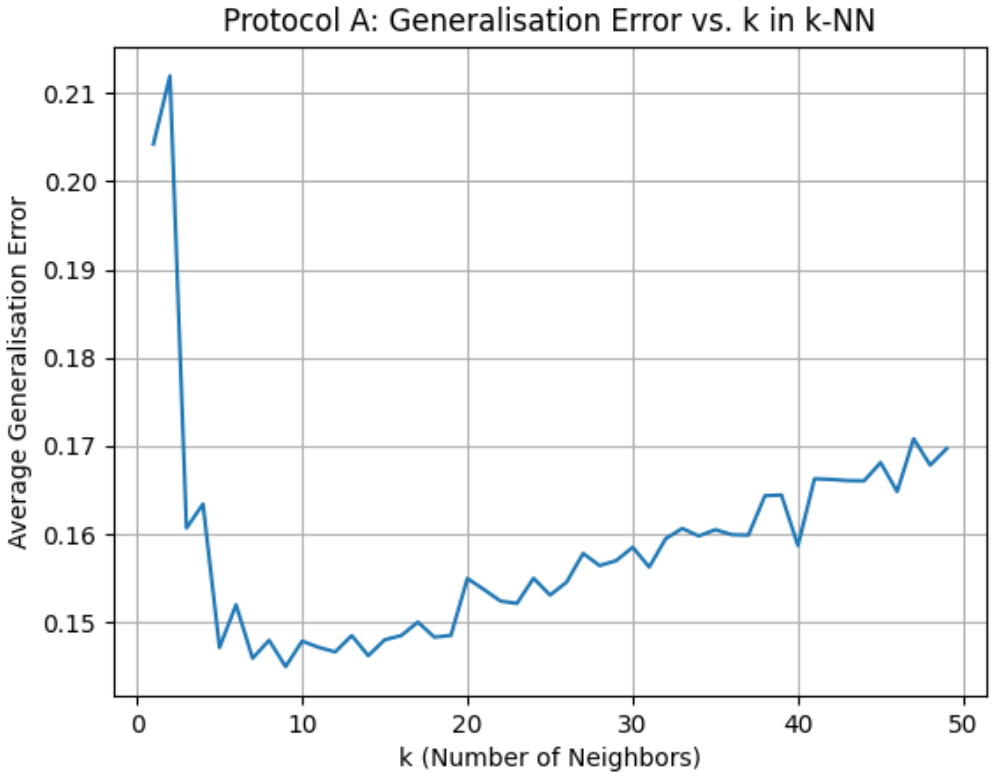

Figure 2: Protocol A

2.3 Generating the data

8.

In Protocol B, we explore the relationship between training size and the optimal k-value in the k-NN algorithm. For smaller training sizes, the optimal k-value is approximately 3, as fewer neighbors are sufficient to capture the local patterns in the limited data. As the training size increases, the optimal k-value grows sharply because more neighbors are needed to accurately represent the more complex local distributions in the larger dataset. However, this rate of increase slows as the training size continues to grow, eventually approaching a steady state. This plateau occurs because, once k becomes large enough to capture the local distribution effectively, additional neighbors do not contribute to better performance and can even lead to underfitting, where the model becomes too generalized and fails to capture finer details in the data.

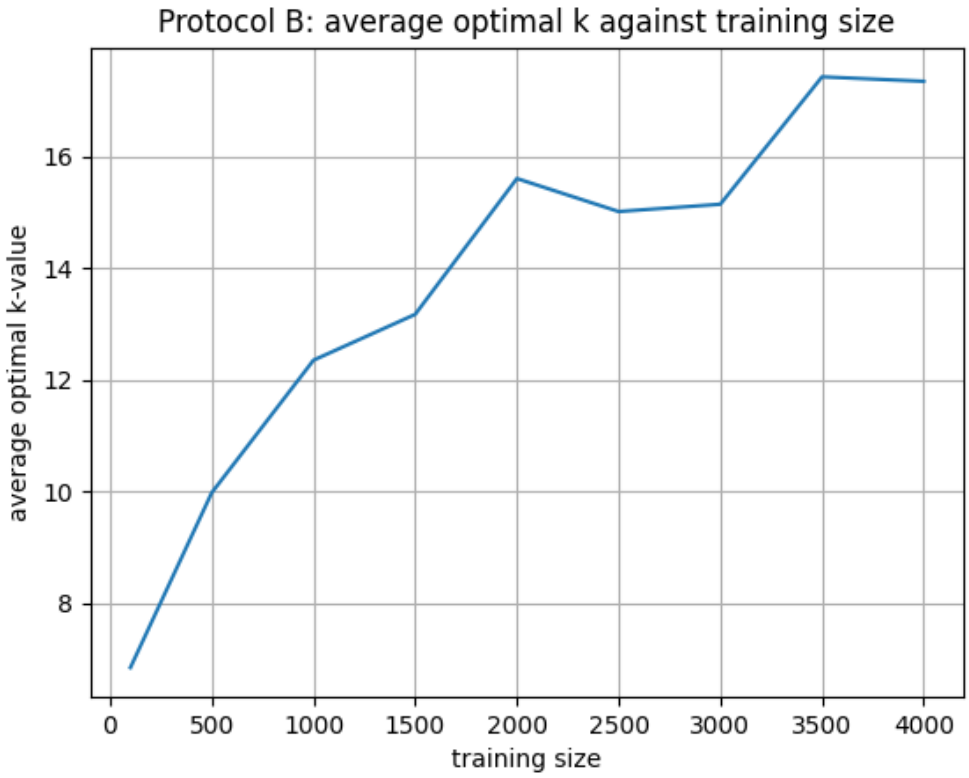

Figure 3: Protocol B

PART III

3.1 Questions

9.a

To show the kernel function is positive semidefinite, we can start from its definition. A kernel K is called a is positive semidefinite kernel on $\mathcal{X}$ if

$$\sum_{i=1}^{n}\sum_{j=1}^{n} a_i a_j K\left(x_i, x_j\right) \geq 0 \tag{1}$$

holds for all $x_1, \ldots, x_n \in \mathcal{X}, n \in \mathbb{N}, a_1, \ldots, a_n \in \mathbb{R}$. In our case, we have:

$$\sum_{i=1}^{n}\sum_{j=1}^{n} a_i a_j K\left(x_i, x_j\right) = \sum_{i=1}^{n}\sum_{j=1}^{n} a_i a_j \left(c + x_i^T x_j\right), \tag{2}$$

which means it need to satisfy:

$$\begin{aligned} & \sum_{i=1}^{n}\sum_{j=1}^{n} a_i a_j c + \sum_{i=1}^{n}\sum_{j=1}^{n} a_i a_j \left(x_i^T x_j\right) \geq 0, \\ \Rightarrow \quad & \|\boldsymbol{a}\|^2 c + \| \sum_{i=1}^{n} a_i x_i \|^2 \geq 0, \\ \Rightarrow \quad & c \geq 0. \end{aligned} \tag{3}$$

In that case, we need c be greater than or equal to zero.

9.b

Suppose we use K_c as a kernel function with linear regression, the constant c can be regarded as a bias. Then, we can consider the influence of this bias. When c is larger, it increases the overall similarity between any two points, which leads to a solution that is less sensitive to individual differences among points. If $c = 0$, the solution is based purely on the inner product between $\boldsymbol{x}$ and $\mathbf{z}$, meaning there is no additional offset or baseline similarity introduced. In summary, c acts as a regularization term that controls the baseline similarity in the data.

10

To simulate a 1-Nearest Neighbor (1-NN) classifier using a Gaussian kernel, we start by examining the Gaussian kernel function $K_\beta(\boldsymbol{x}, \mathbf{t}) = \exp\left(-\beta\|\boldsymbol{x} - \mathbf{t}\|^2\right)$ applied to our dataset $(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_m, y_m)$. Given this kernel, we construct the kernel matrix $K_\beta(\boldsymbol{x}, \boldsymbol{x})$ for the training points:

$$K_\beta(\boldsymbol{x}, \boldsymbol{x}) = \begin{bmatrix} 1 & K_\beta(\boldsymbol{x}_1, \boldsymbol{x}_2) & \cdots & K_\beta(\boldsymbol{x}_1, \boldsymbol{x}_m) \\ K_\beta(\boldsymbol{x}_2, \boldsymbol{x}_1) & 1 & \cdots & K_\beta(\boldsymbol{x}_2, \boldsymbol{x}_m) \\ \vdots & \vdots & \ddots & \vdots \\ K_\beta(\boldsymbol{x}_m, \boldsymbol{x}_1) & K_\beta(\boldsymbol{x}_m, \boldsymbol{x}_2) & \cdots & 1 \end{bmatrix},$$

where each diagonal element is 1, indicating $K_\beta(\boldsymbol{x}_i, \boldsymbol{x}_i) = 1$. We can express the weight vector $\boldsymbol{\alpha}$ as:

$$\boldsymbol{\alpha} = K_\beta(\boldsymbol{x}, \boldsymbol{x})^{-1}\mathbf{y},$$

where $\mathbf{y} = [y_1, \ldots, y_m]^T$ contains the labels. Let $G = K_\beta(\boldsymbol{x}, \boldsymbol{x})^{-1}$ denote the inverse kernel matrix, with elements g_{ij}. Given that $K_\beta(\boldsymbol{x}, \boldsymbol{x})$ is symmetric, G is also symmetric and has a constant diagonal structure. Thus, we can normalize G as:

$$G = c\begin{bmatrix} 1 & g_{12} & \cdots & g_{1m} \\ g_{21} & 1 & \cdots & g_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ g_{m1} & g_{m2} & \cdots & 1 \end{bmatrix}.$$

Using this normalized inverse matrix G, we express α_i as:

$$\alpha_i = \sum_{j=1}^{m} g_{ij}y_j = y_i + \sum_{i \neq j} g_{ij}y_j.$$

Substituting $\boldsymbol{\alpha}$ back into $f(\mathbf{t})$, we get:

$$f(\mathbf{t}) = \sum_{i=1}^{m}\left(y_i K_\beta(\boldsymbol{x}_i, \mathbf{t}) + \sum_{i \neq j} g_{ij}y_j K_\beta(\boldsymbol{x}_i, \mathbf{t})\right).$$

To simulate a 1-NN classifier, we aim for $f(\mathbf{t})$ to be dominated by the nearest point to $\mathbf{t}$, say $\boldsymbol{x}_{i^*}$. This requires:

$$\exp\left(-\beta\|\boldsymbol{x}_{i^*} - \mathbf{t}\|^2\right) > \exp\left(-\beta\|\boldsymbol{x}_i - \mathbf{t}\|^2\right) \quad \text{for all } i \neq i^*,$$

ensuring that $\boldsymbol{x}_{i^*}$ has the largest influence on $f(\mathbf{t})$. A large and positive β is needed to enforce this sharp decay, making the kernel value significant only for the nearest point. With a sufficiently large β, the expression for $f(\mathbf{t})$ simplifies as contributions from non-nearest points become negligible:

$$f(\mathbf{t}) \approx \sum_{i=1}^{m} y_i K_\beta(\boldsymbol{x}_i, \mathbf{t}).$$

To fully simulate 1-NN, we need $y_{i^*}K_\beta(\boldsymbol{x}_{i^*}, \mathbf{t})$ to dominate:

$$K_\beta(\boldsymbol{x}_{i^*}, \mathbf{t}) > \sum_{i \neq i^*} K_\beta(\boldsymbol{x}_i, \mathbf{t}).$$

This can be ensured by choosing β large enough so that only the nearest neighbor contributes meaningfully to $f(\mathbf{t})$. Therefore, we can define a function $\beta = \hat{\beta}(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_m, \mathbf{t})$ with a lower bound that satisfies this condition. By appropriately choosing β (based on distances in the dataset), we can use the Gaussian kernel to simulate a 1-NN classifier, where $f(\mathbf{t})$ relies almost exclusively on the label of the nearest neighbor $\boldsymbol{x}_{i^*}$.

11.a

Let $C \subset \mathcal{X}$ be a subset of $\mathcal{X}$ with cardinality (i.e. number of elements) $|C| = 2n$. The misclassification can be express as

$$\begin{aligned} \mathcal{E}_\rho(f_i) &= \sum_{\mathcal{C}\times\mathcal{Y}} \mathbf{1}_{\{f_i(x)\neq y\}} \rho(\{(x,y)\}), \\ \Rightarrow \quad \mathcal{E}_\rho(f_i) &= \sum_{\mathcal{C}\times\mathcal{Y}} \mathbf{1}_{\{f_i(x)\neq y\}} \frac{1}{2n}. \end{aligned} \tag{4}$$

In this case, $y = f_i(x)$ for non-zero density. If we substitute this into above equation, we have $\mathbf{1}_{\{f_i(x)\neq y\}}$ becomes zero. For the whole set, we can find f such $\mathbf{1}_{\{f_i(x)\neq f\}}$. This means we have a misclassification. The lowest error or mistake we can take is zero.

$$\inf \mathcal{E}_{\rho_i}(f) = 0. \tag{5}$$

Thus, we have

$$\mathcal{E}_{\rho_i}(f_i) = \inf_{f:\mathcal{X}\to\mathcal{Y}} \mathcal{E}_{\rho_i}(f) = 0. \tag{6}$$

11.b

The expectation can be expressed as:

$$\mathbb{E}_{S\sim\rho_i^n} \mathcal{E}_{\rho_i}(A(S)) = \sum_{S\in C^n} \mathcal{E}_{\rho_i}(A(S)) \cdot \mathbb{P}_{S\sim\rho_i^n}(S) \tag{7}$$

In the question, we find that the probability with respect to S is uniform. Thus, we have

$$\mathbb{P}_{S\sim\rho_i^n}(S_j) = \frac{1}{k} = \frac{1}{(2n)^n}. \tag{8}$$

which shows

$$\mathbb{E}_{S\sim\rho_i^n} \mathcal{E}_{\rho_i}(A(S)) = \frac{1}{k} \sum_{j=1}^{k} \mathcal{E}_{\rho_i}\left(A\left(S_j^i\right)\right) \tag{9}$$

Using Hint 2

$$\max_\ell \alpha_\ell \geq \frac{1}{m} \sum_{\ell=1}^{m} \alpha_\ell \geq \min_\ell \alpha_\ell, \tag{10}$$

we can rewrite Equation 9 as

$$\begin{aligned} \max_{i=1,\ldots,T} \mathbb{E}_{S\sim\rho_i^n} \mathcal{E}_{\rho_i}(A(S)) &\geq \frac{1}{T} \sum_{i=1}^{T} \frac{1}{k} \sum_{j=1}^{k} \mathcal{E}_{\rho_i}\left(A\left(S_j^i\right)\right), \\ \Rightarrow \max_{i=1,\ldots,T} \mathbb{E}_{S\sim\rho_i^n} \mathcal{E}_{\rho_i}(A(S)) &\geq \frac{1}{k} \sum_{j=1}^{k} \frac{1}{T} \sum_{i=1}^{T} \mathcal{E}_{\rho_i}\left(A\left(S_j^i\right)\right), \\ \Rightarrow \max_{i=1,\ldots,T} \mathbb{E}_{S\sim\rho_i^n} \mathcal{E}_{\rho_i}(A(S)) &\geq \min_{j=1,\ldots,k} \frac{1}{T} \sum_{i=1}^{T} \mathcal{E}_{\rho_i}\left(A\left(S_j^i\right)\right). \end{aligned} \tag{11}$$

Here we rearrange the summation and show the inequality.

11.c

The risk can be expressed as

$$\mathcal{E}_{\rho_i}\left(A\left(S_j^i\right)\right) = \frac{1}{2n} \sum_{x\in C} \mathbf{1}_{\left\{A\left(S_j^i\right)(x)\neq f_i(x)\right\}}. \tag{12}$$

From the hint, we lower bound the risk with respect to the errors only over S_j'.

$$\mathcal{E}_{\rho_i}\left(A\left(S_j^i\right)\right) \geq \frac{1}{2n} \sum_{v\in R_j} \mathbf{1}_{\left\{A\left(S_j^i\right)(v)\neq f_i(v)\right\}}, \tag{13}$$

as $R_j = \{v_1, \ldots, v_p\}$ is the subset of points of C that not belong to S_j. Using the same trick in (b), we sum over i on both sides and rearrange the summation:

$$\begin{aligned}
&\frac{1}{T}\sum_{i=1}^{T}\mathcal{E}_{\rho_i}\left(A\left(S_j^i\right)\right) \geq \frac{1}{T}\sum_{i=1}^{T}\frac{1}{2n}\sum_{v\in R_j}\mathbf{1}_{\left\{A\left(S_j^i\right)(v)\neq f_i(v)\right\}},\\
\Rightarrow &\frac{1}{T}\sum_{i=1}^{T}\mathcal{E}_{\rho_i}\left(A\left(S_j^i\right)\right) \geq \frac{1}{T}\sum_{i=1}^{T}\frac{1}{2n}\sum_{v\in R_j}\mathbf{1}_{\left\{A\left(S_j^i\right)(v)\neq f_i(v)\right\}},\\
\Rightarrow &\frac{1}{T}\sum_{i=1}^{T}\mathcal{E}_{\rho_i}\left(A\left(S_j^i\right)\right) \geq \frac{1}{2}\left(\frac{1}{n}\sum_{v\in R_j}\frac{1}{T}\sum_{i=1}^{T}\mathbf{1}_{\left\{A\left(S_j^i\right)(v)\neq f_i(v)\right\}}\right),\\
\Rightarrow &\frac{1}{T}\sum_{i=1}^{T}\mathcal{E}_{\rho_i}\left(A\left(S_j^i\right)\right) \geq \frac{1}{2}\min_{v\in R_j}\frac{1}{T}\sum_{i=1}^{T}\mathbf{1}_{\left\{A\left(S_j^i\right)(v)\neq f_i(v)\right\}}.
\end{aligned} \tag{14}$$

11.d

For any $v \in R_j$, we partition the set of all functions $\mathcal{Y}^C = \{f_1, \ldots, f_T\}$ into $T/2$ disjoint pairs $(f_i, f_{i'})$, such that: $f_i(x) \neq f_{i'}(x)$ if and only if $x = v$. In that case, any pair $(f_i, f_{i'})$ need to satisfy:

$$\mathbf{1}_{\left\{A\left(S_j^i\right)(v)\neq f_i(v)\right\}} + \mathbf{1}_{\left\{A\left(S_j^{i'}\right)(v)\neq f_{i'}(v)\right\}} = 1. \tag{15}$$

This is because $A\left(S_j^i\right)(v) = f_i(v)$, then $A\left(S_j^{i'}\right)(v) \neq f_{i'}(v)$, and vice versa. Then, we can find

$$\frac{1}{T}\sum_{i=1}^{T}\mathbf{1}_{\left\{A\left(S_j^i\right)(v)\neq f_i(v)\right\}} = \frac{1}{T}\times\frac{T}{2} = \frac{1}{2}, \tag{16}$$

as the original summation is turned into $\frac{T}{2}$ pairs of 1.

11.e

Consider the random variable $Y = 1 - Z$. Since $Z \in [0, 1]$, we have $Y \in [0, 1]$. Using Markov's inequality:

$$\mathbb{P}(Y \geq a) \leq \frac{\mathbb{E}[Y]}{a}. \tag{17}$$

Substituting $\mathbb{E}[Y] = 1 - \mu$, we have

$$\mathbb{P}(1 - Z \geq a) \leq \frac{1-\mu}{a}, \tag{18}$$

which can be rearranged as

$$\mathbb{P}(Z \leq 1 - a) \leq \frac{1-\mu}{a}. \tag{19}$$

Using the fact $\mathbb{P}(Z > 1 - a) = 1 - \mathbb{P}(Z \leq 1 - a)$, we get

$$\begin{aligned}
&\mathbb{P}(Z > 1 - a) \geq 1 - \frac{1-\mu}{a},\\
\Rightarrow &\mathbb{P}(Z > 1 - a) \geq \frac{\mu - (1-a)}{a}.
\end{aligned} \tag{20}$$

11.f

To prove

$$\mathbb{P}_{S\sim\rho^n}\left(\mathcal{E}_\rho(A(S)) > \frac{1}{8}\right) \geq \frac{1}{7}, \tag{21}$$

one can write it the form we derived in (e):

$$\mathbb{P}_{S\sim\rho^n}\left(\mathcal{E}_\rho(A(S)) > 1 - \frac{7}{8}\right) \geq \frac{\mathbb{E}[\mathcal{E}_\rho(A(S))] - \frac{1}{8}}{\frac{7}{8}}. \tag{22}$$

For (c) and (d), we get

$$\begin{aligned} \mathbb{E}_{S\sim\rho_i^n}\mathcal{E}_{\rho_i}(A(S)) &\geq \frac{1}{2}\frac{1}{T}\sum_{i=1}^{T}\mathbf{1}_{\{A(S_j^i)(v)\neq f_i(v)\}}, \\ \Rightarrow \mathbb{E}_{S\sim\rho_i^n}\mathcal{E}_{\rho_i}(A(S)) &\geq \frac{1}{4}. \end{aligned} \tag{23}$$

Substitute $\mathbb{E}[\mathcal{E}_\rho(A(S))] = \frac{1}{4}$, we have

$$\begin{aligned} \mathbb{P}_{S\sim\rho^n}\left(\mathcal{E}_\rho(A(S)) > 1 - \frac{7}{8}\right) &\geq \frac{\frac{1}{4} - \frac{1}{8}}{\frac{7}{8}}, \\ \Rightarrow \mathbb{P}_{S\sim\rho^n}\left(\mathcal{E}_\rho(A(S)) > \frac{1}{8}\right) &\geq \frac{1}{7}. \end{aligned} \tag{24}$$

11.g.i

For any algorithm $A : S \mapsto (f : \mathcal{X} \to \mathcal{Y})$ and for any integer $n \in \mathbb{N}$, there always exists a distribution ρ such that:

1. The optimal function f^* for ρ achieves zero error: $\inf_{f:\mathcal{X}\to\mathcal{Y}} \mathcal{E}_\rho(f) = 0$,
2. The probability of A 's misclassification error exceeding a threshold is bounded below:

$$\mathbb{P}_{S\sim\rho^n}\left(\mathcal{E}_\rho(A(S)) > \frac{1}{8}\right) \geq \frac{1}{7} \tag{25}$$

11.g.ii

Let us use the summary in g(i). The definition of learnability in the question requires the inequality:

$$\mathbb{P}_{S\sim\rho^n}\left(\mathcal{E}_\rho(A(S)) \leq \epsilon\right) \geq 1 - \delta, \tag{26}$$

which states that the algorithm A must, with high probability, achieve a risk close to the optimal risk for a given hypothesis space $\mathcal{H}$. Equivalently, using the complement of this event, we have:

$$\mathbb{P}_{S\sim\rho^n}\left(\mathcal{E}_\rho(A(S)) > \epsilon\right) \leq \delta. \tag{27}$$

Now, using the results of the No-Free-Lunch theorem summarized in g(i), we found that for the space of all functions $\mathcal{Y}^\mathcal{X}$ and for any algorithm A, there exists a distribution ρ such that:

$$\mathbb{P}_{S\sim\rho^n}\left(\mathcal{E}_\rho(A(S)) > \frac{1}{8}\right) \geq \frac{1}{7}. \tag{28}$$

This result shows that the error probability is bounded below by $\frac{1}{7}$, contradicting the condition required for learnability in the definition (where δ must be arbitrarily small for sufficiently large n). Therefore, we conclude that the space $\mathcal{Y}^\mathcal{X}$ of all functions $f : \mathcal{X} \to \mathcal{Y}$ is not learnable.

This contradiction arises because the No-Free-Lunch theorem shows that no algorithm can perform uniformly well across all possible distributions ρ over $\mathcal{X} \times \mathcal{Y}$ without additional assumptions about the structure of the hypothesis space or the data distribution.

11.g.iii

The No-Free-Lunch theorem asserts that no single algorithm can perform optimally across all tasks. To achieve learnability and strong performance, it is essential to explore different algorithms and tailor them to specific problems. Additionally, making appropriate assumptions is crucial for certain tasks, as they guide the learning process. Moreover, understanding and leveraging the inherent structure of the dataset is key to achieving better results.