STATISTICAL MACHINE LEARNING (WS2024/2025)
PROBABLY APPROXIMATELY CORRECT LEARNING

Assignment 1. Let the observation $\boldsymbol{x} \in \mathcal{X} = \mathbb{R}^n$ and the hidden state $y \in \mathcal{Y} = \{+1, -1\}$ be generated by a multivariate normal distribution

$$p(\boldsymbol{x}, y) = p(y) \, \frac{1}{(2\pi)^{\frac{n}{2}} \det(\boldsymbol{C}_y)^{\frac{1}{2}}} e^{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_y)^T \boldsymbol{C}_y^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_y)}$$

where $\boldsymbol{\mu}_y \in \mathbb{R}^n$, $y \in \mathcal{Y}$, are mean vectors, $\boldsymbol{C}_y \in \mathbb{R}^{n \times n}$, $y \in \mathcal{Y}$, are covariance matrices and $p(y)$ is a prior probability. Assume that the model parameters are unknown and we want to learn a strategy $h \in \mathcal{X} \to \mathcal{Y}$ which minimizes the probability of misclassification. To this end we use a learning algorithm $A \colon \cup_{m=1}^{\infty} (\mathcal{X} \times \mathcal{Y})^m \to \mathcal{H}$ which returns a strategy h from the class $\mathcal{H} = \{h(\boldsymbol{x}) = \operatorname{sign}(\langle \boldsymbol{w}, \boldsymbol{x} \rangle + b) \mid \boldsymbol{w} \in \mathbb{R}^n, b \in \mathbb{R}\}$ containing all linear classifers.

a) What is the approximation error in case that $\boldsymbol{C}_+ = \boldsymbol{C}_-$?

b) Is the approximation error going to increase or decrease if $\boldsymbol{C}_+ \neq \boldsymbol{C}_-$?

c) Give example(s) of distribution $p(x, y)$ such that the approximation error is zero when using the class $\mathcal{H}$.

Assignment 2. Assume we train a Convolution Neural Network (CNN) classifier $h \colon \mathcal{X} \to \mathcal{Y}$ to predict a digit $y \in \mathcal{Y} = \{0, 1, \ldots, 9\}$ from an image $x \in \mathcal{X}$. We train the CNN by the Stochastic Gradient Descent (SGD) algorithm using 100 epochs. After each epoch we save the current weights so that at the end of training we have a set $\mathcal{H} = \{h_t \colon \mathcal{X} \to \mathcal{Y} \mid i = 1, \ldots, 100\}$ containing 100 different CNN classifiers. Let $h_{\mathcal{H}} \in \arg\min_{h \in \mathcal{H}} R(h)$ denote the best CNN in $\mathcal{H}$ which, however, we cannot find directly because $R(h)$ is unknown. Therefore, we approximate the best CNN classifier $h_{\mathcal{H}}$ by $\hat{h} \in \arg\min_{h \in \mathcal{H}} R_{\mathcal{V}^m}(h)$ selected based on the validation error

$$R_{\mathcal{V}^m}(h) = \frac{1}{m} \sum_{i=1}^{m} [\![y^j \neq h(x^j)]\!] \,,$$

computed from the validation set $\mathcal{V}^m = \{(x^i, y^i) \in (\mathcal{X} \times \mathcal{Y}) \mid i = 1, \ldots, m\}$ i.i.d. drawn from $p(x, y)$. The validation set $\mathcal{V}^m$ has not been used by SGD for training.

a) What is the minimum number of examples in the validation set $\mathcal{V}^m$ required to confidently assert, with a 95% probability, that the generalization error $R(\hat{h})$ of the found CNN classifier $\hat{h}$ exceeds the generalization error $R(h_{\mathcal{H}})$ of the best CNN classifier $h_{\mathcal{H}}$ by no more than 1% ?

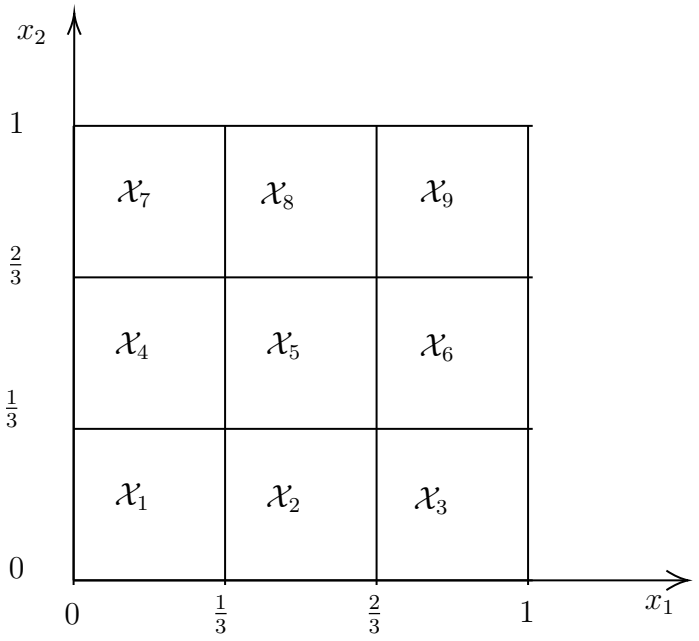

FIGURE 1. Example of partitioning of $(n = 2)$-dimensional input space $\mathcal{X} = [0, 1]^2$ into $K = D^n = 9$ hypercubes when $D = 3$, i.e. the hypercubes $\mathcal{X}_1 \cup \mathcal{X}_2 \cup \cdots \cup \mathcal{X}_9$ are non-overlapping squares with side $\frac{1}{3}$.

b) Assume that you train M CNN classifiers with different architectures, each trained by 100 SGD epochs. You use the validation error to choose the optimal combination of architecture and number of epochs. How many validation examples are required to provide the same guarantees as in assignment a)?

Assignment 3. Assume a two-class classification problem, $\mathcal{Y} = \{-1, +1\}$, with 0/1-loss $\ell(y, y') = [\![y \neq y']\!]$, when the input space $\mathcal{X} = [0, 1]^n$ is the n-dimensional hypercube. Let us partition $\mathcal{X}$ into K equally sized bins $\mathcal{X}_j$, $j \in \{1, \ldots, K\}$, each bin being itself an axis-parallel hypercube, such that

$$\mathcal{X} = \cup_{k=1}^{K} \mathcal{X}_k \quad \text{and} \quad \mathcal{X}_k \cap \mathcal{X}_j = \emptyset \,, \forall k \neq j \,,$$

where $K = D^n$ and $D \in \{1, 2, 3, \ldots, \}$ is the number of hypercubes along a coordinate axis. Figure 1 shows an example of such a partitioning in $(n = 2)$-dimensional input space when $D = 3$. A histogram classifier $h \colon \mathcal{X} \to \{-1, +1\}$, parametrized by a vector $\boldsymbol{w} \in \{-1, +1\}^K$, assigns an input x into class w_k where k identifies the hypercube $\mathcal{X}_k$ containing the input $\boldsymbol{x} \in \mathcal{X}_k$. That is, the histogram classifier reads

$$h(\boldsymbol{x}) = \sum_{k=1}^{K} w_k [\![\boldsymbol{x} \in \mathcal{X}_k]\!] \,. \tag{1}$$

Let

$$\mathcal{H}_K = \left\{ h\colon \mathcal{X} \to \{-1,+1\} \mid h(\boldsymbol{x}) = \sum_{k=1}^{K} w_k [\![\boldsymbol{x} \in \mathcal{X}_k]\!] \,, \boldsymbol{w} \in \{-1,+1\}^K \right\}$$

denote a hypothesis space composed of all histogram classifiers that partition the input space into K hypercubes. Assume we have a training set $\mathcal{T}^m = \{(\boldsymbol{x}^1, y^1), \ldots, (\boldsymbol{x}^m, y^m)\} \in (\mathcal{X} \times \{+1,-1\})^m$ drawn i.i.d. from some unknown $p(\boldsymbol{x}, y)$.

a) How does the number of strategies $|\mathcal{H}_K|$ depend on the number of bins on D and n?

b) Assume the number of bins K is fixed. The ERM based algorithm transforms the learning of the histogram classifier into the optimization problem

$$h_m \in \operatorname*{Arg\,min}_{h \in \mathcal{H}_K} R_{\mathcal{T}^m}(h) \quad \text{where} \quad R_{\mathcal{T}^m}(h) = \frac{1}{m} \sum_{j=1}^{m} \ell(y^j, h(x^j)) \,. \tag{2}$$

Design a computationaly tractable algorithm which solves the problem (2).

c) Let $\delta \in (0,1)$. Let $h_{\mathcal{H}} \in \arg\min_{h \in \mathcal{H}_K} R(h)$ be the best histogram classifier in $\mathcal{H}_K$. Find an upper bound on the estimation error $R(h_m) - R(h_{\mathcal{H}})$ which holds with probability $1 - \delta$ at least.