A. Kernels

1. Show that the dimension of the feature space associated to the polynomial kernel of degree \(d\), \(K: \mathbb{R}^N \times \mathbb{R}^N \rightarrow \mathbb{R}, K(x, x') = (x \cdot x' + c)^d\), with \(c > 0\), is
\[
\binom{N + d}{d}.
\]

Write \(K\) in terms of kernels \(k_i: (x, x') \rightarrow (x \cdot x')^i, i \in [0, d]\). What is the weight assigned to each \(k_i\) in that expression? How does it vary as a function of \(c\).

2. For \(\alpha \geq 0\), the kernel \(K_\alpha: (x, x') \rightarrow \sum_{k=1}^{N} \min(|x_k|^\alpha, |x'_k|^\alpha)\) over \(\mathbb{R}^N \times \mathbb{R}^N\) is used in image classification. Show that \(K_\alpha\) is PDS. To do that, you can proceed as follows.

(a) Use the fact that \((f, g) \mapsto \int_{t=0}^{+\infty} f(t)g(t)dt\) is an inner product over the set of measurable functions over \([0, +\infty)\) to show that \((x, x') \mapsto \min(x, x')\) is a PDS kernel (hint: associate an indicator function to \(x\) and another one to \(x'\)).

(b) Use the previous question to show that \(K_1\) is PDS and similarly \(K_\alpha\) with other values of \(\alpha\).

B. Boosting

1. Assume that the main weak learner assumption of AdaBoost holds. Let \(h_t\) be the base learner selected at round \(t\). Show that the base learner \(h_{t+1}\) selected at time \(t\) must be different from \(h_t\).

2. Let the training sample be \(S = ((x_1, y_1), \ldots, (x_m, y_m))\). Suppose we wish to penalize differently errors made on \(x_i\) versus \(x_j\). To do that, we associate some non-negative importance weight \(w_i\) to each point \(x_i\) and define the objective function \(F(\alpha) = \sum_{i=1}^{m} w_i e^{-y_i f(x_i)}\), where \(f = \sum \alpha_i h_i\) with the notation already used in class. Show that this function is convex and differentiable and use it to derive a boosting-type algorithm (give a clear description of the algorithm similar to that of AdaBoost presented in class).
C. Perceptron

1. The margin bound on the maximum number of updates presented in class for the perceptron algorithm was given for the special case \( \eta = 1 \). Consider now the general perceptron update \( w_{t+1} \leftarrow w_t + \eta y_t x_t \), where \( \eta > 0 \). With the same assumptions as for the theorem presented in class, prove a bound on the maximum number of mistakes. How does \( \eta \) affect the bound?

2. Suppose each input vector \( x_t, t \in [1, T] \), coincides with the \( t \)th unit vector of \( \mathbb{R}^T \). How many updates does it take the perceptron algorithm to converge? How does the number of updates (or mistakes) compare with the margin bound?