1. (Kernels)

(a) Consider the constant function, \( k(x, y) = c \) for all \( x, y \). Is \( k \) a symmetric positive semidefinite kernel? If not, explain why not. If so, describe its reproducing kernel Hilbert space.

(b) For two symmetric, positive semidefinite kernels \( k_1, k_2 \) defined on the same space, let \( k \) be their minimum, \( k(u, v) = \min\{k_1(u, v), k_2(u, v)\} \). Is \( k \) also a symmetric positive semidefinite kernel?

(c) Consider a symmetric, positive semidefinite kernel, \( k \) defined on \( X \), and let \( K \) denote the kernel matrix corresponding to the set \( \{x_1, \ldots, x_n\} \subset X \) (that is, \( K_{i,j} = k(x_i, x_j) \)). Suppose that \( K \) has full rank. For \( u \in X \), define \( k_x(u) = (k(u, x_1), \ldots, k(u, x_n))^\top \), and for \( u, v \in X \), define

\[
\tilde{k}(u, v) = k(u, v) - k_x(u)^\top K^{-1} k_x(v).
\]

Show that \( \tilde{k} \) is also a symmetric, positive semidefinite kernel.

2. (SVMs and the indicator kernel)

Consider the hard-margin SVM with the kernel

\[
k(u, v) = 1[u = v].
\]

Suppose that we draw a sample \( (x_1, y_1), \ldots, (x_n, y_n) \), and all the \( x_i \) are distinct. If we represent the kernel as an inner product \( k(u, v) = \langle \Phi(u), \Phi(v) \rangle \), then we know that the SVM produces a classifier of the form

\[
f_n(x) = \text{sign}\left( \langle w^*, \Phi(x) \rangle \right),
\]

where

\[
w^* = \sum_{i=1}^n \alpha_i y_i \Phi(x_i).
\]

(a) Calculate the values of the optimal \( \alpha_i \)’s.

(b) If \( f_n \) is the SVM classifier, what is \( f_n(x_i) \) for a training example \( x_i \)?

(c) What is \( f_n(x) \) for an unseen \( x \), that is, an \( x \) that is not in \( \{x_1, \ldots, x_n\} \)?

We’ll see that the misclassification probability of this SVM classifier satisfies the upper bound

\[
\mathbb{E}R(f_n) = O(\min(\mathbb{E}\|w^*\|^2, \mathbb{E}k)/n),
\]

where \( k \) is the number of support vectors.

(d) What is the margin, \( 1/\|w^*\|? \)

(e) What is the number of support vectors, \( k \)?

(f) How are the answers to (2d) and (2e) consistent with your observations in (2b) and (2c)?

(g) How do your answers change if you consider the soft-margin SVM?

3. (Fast approximate primal SVMs) The support vector machine is based on the following optimization problem.

\[
\begin{align*}
\text{minimize}_{w \in \mathbb{R}^d} & \quad \frac{1}{2}\|w\|^2 + \frac{C}{n} \sum_{i=1}^n (1 - y_i w^x_i)_+.
\end{align*}
\]

where \( x_1, \ldots, x_n \in \mathbb{R}^d, y_1, \ldots, y_n \in \{-1, 1\} \), and \( C > 0 \). The usual way to express this as a quadratic program is to introduce \( n \) slack variables, \( \xi_i \geq 0 \), and \( n \) constraints. An alternative is to introduce a single slack variable and \( 2^n \) constraints.
(a) Show that the SVM optimization is equivalent to the following QP.

\[ \begin{align*}
& \text{minimize}_{w \in \mathbb{R}^d, \xi \in \mathbb{R}} & & \frac{1}{2} \|w\|^2 + C\xi \\
& \text{subject to} & & \forall b \in \{0, 1\}^n, \ \xi \geq \frac{1}{n} \sum_{i=1}^{n} b_i (1 - y_i w' x_i).
\end{align*} \tag{2} \]

It turns out that we can find approximate solutions to this exponentially large optimization problem in time linear in \(n\). The key insight is that the \(2^n\) constraints are not all equally important.

Consider an algorithm that starts with \(B_1 = \{0\} \subset \{0, 1\}^n\), and at iteration \(t\), solves the QP

\[ \begin{align*}
& \text{minimize}_{w_t \in \mathbb{R}^d, \xi_t \in \mathbb{R}} & & \frac{1}{2} \|w_t\|^2 + C\xi_t \\
& \text{subject to} & & \forall b \in B_t, \ \xi_t \geq \frac{1}{n} \sum_{i=1}^{n} b_i (1 - y_i w'_t x_i).
\end{align*} \tag{3} \]

and then sets \(B_{t+1} = B_t \cup \{b\}\), where \(b \in \{0, 1\}^n\) corresponds to the constraint in (2) that requires the largest value of \(\xi\) to make a feasible pair \((w_t, \xi)\), that is, it adds the \(b\) that maximizes

\[ J_t(b) := \frac{1}{n} \sum_{i=1}^{n} b_i (1 - y_i w'_t x_i). \]

The algorithm continues until \(J_t(b) - \xi_t \leq \epsilon\), (\(\epsilon\) is a parameter of the algorithm).

(b) Show that the solution \((w_t, \xi_t)\) returned by this algorithm has a smaller objective than the solution of (2), and that \((w_t, \xi_t + \epsilon)\) is feasible in (2). Hence show that \(w_t\) is an approximate solution to (1).

It turns out (we won’t prove this), that, under some mild assumptions on the data, this algorithm always terminates after some number of steps \(T\) that depends on \(\epsilon\) but not on \(n\) or \(d\). Since the QP (3) can be solved in time polynomial in the number of variables and constraints, this means that, for fixed \(\epsilon\) and \(d\), the algorithm runs in time \(O(n)\).

(c) Suppose now that the data is sparse, that is, many of the components of the \(x_i\) are zero. Define

\[ s = \frac{1}{n} \left| \{(i, j) : 1 \leq i \leq n, 1 \leq j \leq d, x_{i,j} \neq 0\} \right|, \]

where \(x_i = (x_{i,1}, \ldots, x_{i,d})\). Assume that \(s \ll d\) (that is, the data is sparse), and that \(d\) increases with \(n\) such that \(sn \geq d\) (because we do not bother to include features in the \(x_i\) that are always zero). Describe a version of the algorithm that runs in time \(O(sn)\) for fixed \(\epsilon\).