

## 1 Kernels

For a function  $k(X_i, X_j)$  to be a valid kernel, it suffices to show either of the following conditions is true:

1.  $k$  has an inner product representation:  $\exists \Phi : \mathbb{R}^d \rightarrow \mathcal{H}$ , where  $\mathcal{H}$  is some (possibly infinite-dimensional) inner product space such that  $\forall X_i, X_j \in \mathbb{R}^d$ ,  $k(X_i, X_j) = \langle \Phi(X_i), \Phi(X_j) \rangle$ .
2. For every sample  $X_1, X_2, \dots, X_n \in \mathbb{R}^d$ , the Gram matrix

$$K = \begin{bmatrix} k(X_1, X_1) & \cdots & k(X_1, X_n) \\ \vdots & k(X_i, X_j) & \vdots \\ k(X_n, X_1) & \cdots & k(X_n, X_n) \end{bmatrix}$$

is positive semidefinite. For the following parts you can use either condition (1) or (2) in your proofs.

- (a) Given two valid kernels  $k_a$  and  $k_b$ , show that their sum

$$k(X_i, X_j) = k_a(X_i, X_j) + k_b(X_i, X_j)$$

is a valid kernel.

- (b) Given a valid kernel  $k_a$ , show that

$$k(X_i, X_j) = f(X_i)f(X_j)k_a(X_i, X_j)$$

is a valid kernel.

- (c) Given a positive semidefinite matrix  $A$ , show that  $k(X_i, X_j) = X_i^\top A X_j$  is a valid kernel.
- (d) Show why  $k(X_i, X_j) = X_i^\top (\text{rev}(X_j))$  (where  $\text{rev}(x)$  reverses the order of the components in  $x$ ) is *not* a valid kernel.
- (e) Suppose you have  $m$  black box kernel functions, you don't know what is in these kernel functions, but you can evaluate them at any two vectors. For resource reasons, you can only invoke one call to a kernel function when doing on-line classification. Naturally, you want to combine your  $m$  kernel functions into one kernel function. How would you build a potentially more powerful classification method (kernel) using these black box kernel functions?

## 2 Curse of Dimensionality

We have a training set:  $(\mathbf{x}^{(1)}, y^{(1)}), \dots, (\mathbf{x}^{(n)}, y^{(n)})$ , where  $\mathbf{x}^{(i)} \in \mathbb{R}^d$ . Our nearest neighbor classifier is:

$$\text{class}(\mathbf{x}) = y^{(i^*)} \quad \text{where } \mathbf{x}^{(i^*)} \text{ is the nearest neighbor of } \mathbf{x}.$$

Assume any data point  $\mathbf{x}$  is inside the Euclidean ball of radius 1, i.e.  $\|\mathbf{x}\|_2 \leq 1$ . To be confident in our prediction, we want the distance between  $\mathbf{x}$  and its nearest neighbor to be small, within some positive  $\varepsilon$ :

$$\|\mathbf{x} - \mathbf{x}^{(i^*)}\|_2 \leq \varepsilon \quad \text{for all } \|\mathbf{x}\|_2 \leq 1. \quad (1)$$

For this condition to hold, at least how many data points should be in the training set? How does this lower bound depend on the dimension  $d$ ?

Hint: Think about the volumes of the hyperspheres in  $d$  dimensions.