CS 189 Introduction to Machine Learning
Fall 2017 DIS11

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: $\exists\, \Phi : \mathbb{R}^d \to \mathscr{H}$, where $\mathscr{H}$ 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 $X_1, X_2, \ldots, X_n \in R^d$, 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.

Solution: We can show that K is positive semidefinite: $x^\top K x = x^\top (K_a + K_b) x = x^\top K_a x + x^\top K_b x \geq 0$

(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.

Solution: We can show k admits a valid inner product representation:

$$k(X_i, X_j) = f(X_i) f(X_j) \langle \Phi_a(X_i), \Phi_a(X_j) \rangle = \langle f(X_i) \Phi_a(X_i), f(X_j) \Phi_a(X_j) \rangle = \langle \Phi(X_i), \Phi(X_j) \rangle$$

where $\Phi(x) = f(x) \Phi_a(x)$.

(c) Given a positive semidefinite matrix A, show that $k(X_i, X_j) = X_i^\top A X_j$ is a valid kernel.

Solution: We can show k admits a valid inner product representation:

$$k(X_i, X_j) = X_i^\top A X_j = X_i^\top P D^{1/2} D^{1/2} P^\top X_j = \langle D^{1/2} P^\top X_i, D^{1/2} P^\top X_j \rangle = \langle \Phi(X_i), \Phi(X_j) \rangle$$

where $\Phi(x) = D^{1/2} P^\top x$

(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.

Solution: We have that $k((-1,1),(-1,1)) = -2$, but this is invalid since if k is a valid kernel then $\forall x,\ k(x,x) = \langle \Phi(x), \Phi(x) \rangle \geq 0$.

(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?

Solution: In the parts above, we have some rules about possible ways to combine valid kernel functions. With these tules, we can construct potentially better kernel through cross-validation.

2 Curse of Dimensionality

We have a training set: $(\mathbf{x}^{(1)}, y^{(1)}), \ldots, (\mathbf{x}^{(n)}, y^{(n)}), \textit{where } \mathbf{x}^{(i)} \in \mathbb{R}^d$. 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 ε:

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

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.

Solution: Let B_0 be the ball centered at the origin, having radius 1 (inside which we assume our data lies). Let $B_i(\varepsilon)$ be the ball centered at $\mathbf{x}^{(i)}$, having radius ε. For inequality (1) to hold, for any point $\mathbf{x} \in B_0$, there must be at least one index i such that $\mathbf{x} \in B_i(\varepsilon)$. This is equivalent to saying that the union of $B_1(\varepsilon), \ldots, B_n(\varepsilon)$ covers the ball B_0. Let $\text{vol}(B)$ indicate the volume of object B, then we have

$$\sum_{i=1}^{n} \text{vol}(B_i(\varepsilon)) = n\text{vol}(B_1(\varepsilon)) \geq \text{vol}(\cup_{i=1}^{n} B_i(\varepsilon)) \geq \text{vol}(B_0).$$

This implies

$$n \geq \frac{\text{vol}(B_0)}{\text{vol}(B_1(\varepsilon))} = \frac{c(1^d)}{c\varepsilon^d} = \frac{1}{\varepsilon^d}$$

Where the constant c is dependent on the formula for the volume of a hypersphere in d dimensions. This lower bound suggests that to make an accurate prediction on high-dimensional input, we need exponentially many samples in the training set. This exponential dependence is sometimes called the *curse of dimensionality*. It highlights the difficulty of using non-parametric methods for solving high-dimensional problems.