Kernels

Andreas Dahlberg

December 2, 2018

Exercise 1

Exercise 1a)

The kernel we will investigate is

$$k(x,z) = (x^T z + 1)^2$$

where $x = (x_1, x_2)^T$ and $z = (z_1, z_2)^T$. We get that

$$\begin{aligned} k(x,z) &= (x_1 z_1 + x_2 z_2 + 1)^2 = x_1^2 z_1^2 + x_2^2 z_2^2 + 2x_1 z_1 x_2 z_2 + 2x_1 z_1 + 2x_2 z_2 + 1 \\ &= \langle (x_1^2, x_2^2, \sqrt{2}x_1 x_2, \sqrt{2}x_1, \sqrt{2}x_2, 1), (z_1^2, z_2^2, \sqrt{2}z_1 z_2, \sqrt{2}z_1, \sqrt{2}z_2, 1) \rangle_{\mathbb{R}^6} \\ &= \langle \phi(x), \phi(z) \rangle_{\mathbb{R}^6} \end{aligned}$$

where $\phi : \mathbb{R}^2 \to \mathbb{R}^6$ by $\phi(x) = (x_1^2, x_2^2, \sqrt{2}x_1 x_2, \sqrt{2}x_1, \sqrt{2}x_2, 1)$. This means that the new space is of dimension 6. We note that it is much more efficient to calculate $(x^T z + 1)^2$ than $\langle \phi(x), \phi(z) \rangle_{\mathbb{R}^6}$.

Exercise 1b)

Suppose $\langle w, \ell \rangle_H + b$ represents the obtained hyperplane from SVM in the new vector space H. For a new test point $x \in F$ where F is the original feature space, we can calculate

$$\langle w, \phi(x) \rangle_H + b = \langle \sum a_i \phi(x_i) y_i, \phi(x) \rangle_H + b = \sum a_i y_i \langle \phi(x_i), \phi(x) \rangle_H + b = \sum a_i y_i k(x_i, x) + b$$

and we classify the point x depending on the sign of this number. If, for example, k is a polynomial kernel then it is of the form $(x^T z + c)^d$ which means that we don't have to "visit" the new space when classifying a new test point. This is very good since the new space might be of very high dimension and then the computation of the inner product is expensive.

Exercise 2

Exercise 2a)

In this exercise we will prove that the linear kernel $k(x,z) = x^T z$ where $x, z \in \mathbb{R}^N$ is a kernel. We take some $n \in \mathbb{N}$ and arbitrary set $\{x_1, \ldots, x_n\} \subseteq \mathbb{R}^N$. It follows easily that the Gram matrix, K, is symmetric so it remains to prove that it is positive semidefinite. To do this we pick an arbitrary $a = (a_1, \ldots, a_n)^T$ and consider

$$\begin{aligned} a^T K a &= \sum_{i,j} a_i a_j k(x_i, x_j) = \sum_{i,j} \langle a_i x_i, a_j x_j \rangle \\ &= \langle \sum_i a_i x_i, \sum_j a_j x_j \rangle = || \sum_i a_i x_i ||^2 \geq 0 \end{aligned}$$

Which proves that $k(x,z) = x^T z$ is a kernel.

Exercise 2b)

We now have the map $\phi : \mathbb{R}^N \to H$ where H is a (Hilbert/inner-product) space. The goal of this exercise is to prove that $k(x,z) = \langle \phi(x), \phi(z) \rangle_H$ where $x, z \in \mathbb{R}^N$ is a kernel. We take some $n \in \mathbb{N}$ and arbitary set $\{x_1, \ldots, x_n\} \subseteq \mathbb{R}^N$. Since $k(x,z)$ is an inner product it follows that the Gram matrix, K, is symmetric so it remains to prove that it is positive semidefinite. To do this we pick an arbitrary $a = (a_1, \ldots, a_n)^T$ and consider

$$\begin{aligned} a^T K a &= \sum_{i,j} a_i a_j k(x_i, x_j) = \sum_{i,j} a_i a_j \langle \phi(x_i), \phi(x_j) \rangle_H \\ &= \sum_{i,j} \langle a_i \phi(x_i), a_j \phi(x_j) \rangle_H = \langle \sum_i a_i \phi(x_i), \sum_j a_j \phi(x_j) \rangle_H \\ &= || \sum_i a_i \phi(x_i) ||_H^2 \geq 0. \end{aligned}$$

Which proves that $k(x,z) = \langle \phi(x), \phi(z) \rangle_H$ is a kernel.

Exercise 2c)

We let K_j be the Gram matrix corresponding to the kernel k_j for some arbitrary set $\{x_1, \ldots, x_n\} \subseteq \mathbb{R}^N$ and arbitrary $n \in \mathbb{N}$. Since $k_3 = k_1 + k_2$ we have $K_3 = K_1 + K_2$ so that $K_3^T = K_1^T + K_2^T = K_1 + K_2 = K_3$ and for some $a = (a_1, \ldots, a_n)^T$ we have $a^T K_3 a = a^T (K_1 + K_2) a = a^T K_1 a + a^T K_2 a \geq 0$ since both K_1 and K_2 are positive semidefinite. This proves that k_3 is a kernel. Similarly, for $\lambda \geq 0$ we have $K_4 = \lambda K_1$ and $K_4^T = \lambda K_1^T = \lambda K_1 = K_4$ and $a^T K_4 a = \lambda a^T K_1 a \geq 0$ since K_1 is positive semidefinite and $\lambda \geq 0$. This proves that k_4 is a kernel.

Exercise 3

Exercise 3a)

In this exercise we prove that

$$k(x, x') = 7\langle x, x' \rangle^2 + 3 + x^T x' + \exp\left(-\frac{1}{2\sigma^2} ||x - x'||^2\right) \tag{1}$$

is a kernel. First we note that $\langle x, x' \rangle = x^T x'$, 1 and $\exp\left(-\frac{1}{2\sigma^2} ||x - x'||^2\right)$ are kernels (the linear kernel, the "constant" kernel and the Gaussian kernel). From exercice 2c) we know that multiplying a kernel by a positive constant gives us a new kernel and adding two kernels gives us a new kernel. It also holds that the product of two kernels is a kernel, e.g $\langle x, x' \rangle^2$ is a kernel. Combining these properties it follows that the function in equation (1) is a kernel.

Exercise 3b)

We let S and S' be all 2-mers from the two strings X and X' obtained by removing the first letter (G) from all 3-mers that start with G and have no other G's. We can define $k_{base}(s, s') = 1 + (s(1) \cdot s'(1) + s(2) \cdot s'(2))$ where for example $s(1)$ is the first letter in the 2-mer s and the operator $\cdot$ is defined as

$$s(j) \cdot s'(j) = \begin{cases} 1, & s(j) = s'(j) \\ 0, & s(j) \neq s'(j) \end{cases}$$

for $j = 1, 2$.

Exercise 3c)

By running the script *ex3c.py* we get $k(X_1, X_2) = 14$ and $k(X_1, X_3) = 6$.

Exercise 3d)

The algorithm I have implemented for computing the kernel will work even if the strings have unequal lengths or are not multiples of 3. We can note that if the two strings are very long then it is likely that $k(X, X')$ will have a high value. This means that the two strings can be quite dissimilar even though the kernel has a high value. For example, if $k(X, X') = 10$ and $|X| = |X'| = 10^6$ then they are not very similar but if $k(X, X') = 10$ and $|X| = |X'| = 8$ then they are quite similar. Hence, to account for this we can divide the kernel by the sum of the lengths of the two strings.

Finally, if a string ends with XXG or XGX (where X is just some letter from the alphabet) we will not take this G in to account (which could be bad since the G's are important for the similarity between the two strings) when considering the 3-mers that start with G. Hence, we lose some "similarity". We can solve this by, for example, extending XGX to $XGXY$ where Y is a letter not in the alphabet. Then the value of the kernel would increase with at least as many 3-mers, starting with G and have no other G's, as in the other string.