Machine Learning
Home assignment 2: Supervised learning

Solutions are by Olexander Chepurnoi and Yaroslava Lochman.

Problem 1. Нехай:

1. K_1 та $K2$ – кернели над векторами $\mathbb{R}^n \times \mathbb{R}^n$.
2. $a, b \in \mathbb{R}^+$ – дійсні додатне значення.
3. $f : \mathbb{R}^n \to \mathbb{R}$ – функція, що проектує вектор розмірністю n на дійсне число.
4. $\phi : \mathbb{R}^n \to \mathbb{R}^d$ функція, що проектує вектор розмірністю n на вектор розмірністю d .
5. K_3 — кернел над векторами $\mathbb{R}^d \times \mathbb{R}^d$.
6. $p(x)$ — многочлен зі змінною x та тільки додатними коефіцієнтами.

Для кожної з приведених функцій вкажіть, чи є вона кернелом, чи ні. Якщо ви думаєте, що функція є кернелом, доведіть це. Якщо думаєте, що не є, наведіть один контр-приклад.

Solution to the problem 1. Here we denote $\mathbf{K}$ as a Gram matrix of kernel K.

1. $K(x, z) = K_1(x, z) + K_2(x, z)$

   $$K(z, x) = K_1(z, x) + K_2(z, x) = K_1(x, z) + K_2(x, z) = K(x, z)$$

   So $\mathbf{K}$ is symmetric.

   $\mathbf{K} = \mathbf{K}_1 + \mathbf{K}_2$ due to properties of a linear transformation.

   $$\forall \alpha : \alpha^\top \mathbf{K} \alpha = \alpha^\top (\mathbf{K}_1 + \mathbf{K}_2)\alpha = \alpha^\top \mathbf{K}_1 \alpha + \alpha^\top \mathbf{K}_2 \alpha \geq 0$$

   So $\mathbf{K}$ is positive semi-definite.

   Answer: Yes

2. $K(x, z) = K_1(x, z) - K_2(x, z)$

   $$K(z, x) = K_1(z, x) - K_2(z, x) = K_1(x, z) - K_2(x, z) = K(x, z)$$

   So $\mathbf{K}$ is symmetric.

   $\mathbf{K} = \mathbf{K}_1 - \mathbf{K}_2$

   $$\alpha^\top \mathbf{K} \alpha = \alpha^\top (\mathbf{K}_1 - \mathbf{K}_2)\alpha = \alpha^\top \mathbf{K}_1 \alpha - \alpha^\top \mathbf{K}_2 \alpha$$

   It has no guarantee to be non-negative $\forall \alpha$. So $\mathbf{K}$ is not positive semi-definite for every $\{x_n\}$.

   Answer: No

3. $K(x,z) = aK_1(x,z)$

$$K(z,x) = aK_1(z,x) = aK_1(x,z) = K(x,z)$$

So $\mathbf{K}$ is symmetric.

$\mathbf{K} = a\mathbf{K}_1$

$$\forall \alpha : \alpha^\top \mathbf{K}\alpha = \alpha^\top (a\mathbf{K}_1)\alpha = a\alpha^\top \mathbf{K}_1 \alpha \geq 0$$

So $\mathbf{K}$ is positive semi-definite.

Answer: Yes

4. $K(x,z) = -aK_1(x,z)$

$$K(z,x) = -aK_1(z,x) = -aK_1(x,z) = K(x,z)$$

So $\mathbf{K}$ is symmetric.

$\mathbf{K} = -a\mathbf{K}_1$

$$\forall \alpha : \alpha^\top \mathbf{K}\alpha = \alpha^\top (a\mathbf{K}_1)\alpha = a\alpha^\top \mathbf{K}_1 \alpha \leq 0$$

So $\mathbf{K}$ is negative semi-definite. Thus K can not be a kernel function.

Answer: No

5. $K(x,z) = K_1(ax,bz)$

$$K(z,x) = K_1(az,bx) \neq K_1(ax,bz), \text{ when } a \neq b$$

So if $a \neq b$, the function is not symmetric so it can not be a kernel function. Only if $a = b$, K is a kernel function since $K(x,z) = K_1(ax,az) = \phi_1(ax)^\top \phi_1(az) = \hat{\phi}(x)^\top \hat{\phi}(z)$

$$\forall \alpha : \alpha^\top \mathbf{K}\alpha = \alpha^\top (a\mathbf{K}_1)\alpha = a\alpha^\top \mathbf{K}_1 \alpha \leq 0$$

Answer: No, except that $a = b$ (then yes).

6. $K(x,z) = K_1(x,z)K_2(x,z)$

$$K(z,x) = K_1(z,x)K_2(z,x) = K_1(x,z)K_2(x,z) = K(x,z)$$

$\mathbf{K} = \mathbf{K}_1 \odot \mathbf{K}_2$ (Hadamard product). By the Schur Product Theorem the hadamard product of two positive semidefinite matrices is also positive semidefinite. So $\mathbf{K}$ is positive semidefinite.
Answer: Yes

7. $K(x,z) = f(x)f(z)$

Here $f(\cdot)$ is a real-valued function which is a generalization of $\phi(\cdot)$. So already $K(x,z) = f(x)f(z) = f(x)^\top f(z)$ is an inner product in one-dimensional feature space, so it is a kernel function.

Answer: Yes

8. $K(x,z) = K_3(\phi(x), \phi(z))$

   Let $K_3(x,z) = \psi(x)^\top \psi(z))$

$$K(x,z) = K_3(\phi(x),\phi(z)) = \psi(\phi(x))^\top \psi(\phi(x))) = \nu(x)^\top \nu(x) \qquad \nu = \psi \circ \phi$$

   Answer: Yes

9. $K(x,z) = p(K_1(x,z))$

   The polynomial function can be expressed as a composition of functions such as addition, multiplication by a constant and power (which can be derived from multiplication of two kernels). So we may use functions that are already proven in **1** (addition), **3** (multiplication by a constant) and **6** (multiplication of two kernels, hence square and any power using mathematical induction).

   Answer: Yes

10. $K(x,z) = aK_1(x,z) - bK_2(x,z)$ п From the proof of **3** we see that $aK_1(x,z)$ and $bK_2(x,z)$ are kernels, but using the proof in **2** we conclude that their difference may not be a kernel function.

    Answer: No

11. $K(x,z) = -aK_1(x,z) - bK_2(x,z)$

    Again from the proof of **3**: $aK_1(x,z)$ and $bK_2(x,z)$ are kernels, from **1**: $aK_1(x,z) + bK_2(x,z)$ is kernel function as well, but using **4** we conclude that $K(x,z) = -aK_1(x,z) - bK_2(x,z)$ is not a kernel function.

    Answer: No

Problem 2. В лекції ми бачили, як функція кернелу аналітично реалізує проекцію ознак x в новий простір $\phi_1(x)$. Ми також можемо ускладнювати цю проекцію, проектуючи точку $\phi_1(x)$ в простір $\phi_2(\phi_1(x))$.

1. Якщо проекції з x в $\phi_1(x)$ та з $\phi_1(x)$ в $\phi_2(x)$ реалізує лінійна функція кернелу, якою буде функція кернелу, що реалізує проекцію з x в $\phi_2(\phi_1(x))$?

2. Якщо проекції з x в $\phi_1(x)$ та з $\phi_1(x)$ в $\phi_2(x)$ реалізує поліноміальна функція кернелу, якою буде функція кернелу, що реалізує проекцію з x в $\phi_2(\phi_1(x))$?

3. Якщо проекції з x в $\phi_1(x)$ та з $\phi_1(x)$ в $\phi_2(x)$ реалізує radial basis function кернел $e^{-\epsilon(x-z)2}$, якою буде функція кернелу, що реалізує проекцію з x в $\phi_2(\phi_1(x))$?

Solution to the problem 2. Let's denote:

$$K_1(x,y) = \phi_1(x)^\top \phi_1(y) \qquad K_2(x,y) = \phi_2(x)^\top \phi_2(y)$$

Therefore:

$$K(x,y) = \phi_2(\phi_1(x))^\top \phi_2(\phi_1(y)) = K_2(\phi_1(x), \phi_1(y))$$

1. Linear kernel : $K(x, y) = x^\top y + c$

$$K_1(x, y) = x^\top y + c_1 \qquad K_2(x, y) = x^\top y + c_2$$

$$K_2(\phi_1(x), \phi_1(y)) = \phi_1(x)^\top \phi_1(y) + c_2 = x^\top y + c_1 + c_2$$

Answer: $K(x, y) = x^\top y + c_1 + c_2$ – also linear

2. Polynomial kernel: $K(x, y) = (\lambda x^\top y + c)^d$

$$K(x, y) = (\lambda_2 \phi_1(x)^\top \phi_1(y) + c_2)^{d_2} = (\lambda_2(\lambda_1 x^\top y + c_1)^{d_1} + c_2)^{d_2}$$

Answer: $K(x, y) = (\lambda_2(\lambda_1 x^\top y + c_1)^{d_1} + c_2)^{d_2}$ – polynomial as well

3. RBF Kernel: $K(x, y) = e^{-\lambda \|x-y\|^2}$

$$K_1(x, y) = e^{-\lambda_1 \|x-y\|^2} = \phi_1(x)^\top \phi_1(y)$$

$$K_2(x, y) = e^{-\lambda_2 \|x-y\|^2} = \phi_2(x)^\top \phi_2(y)$$

$$K(x, y) = K_2(\phi_1(x), \phi_1(y)) = e^{-\lambda_2 \|\phi_1(x)-\phi_2(y)\|^2}$$

Having:

$$\|\phi_1(x) - \phi_1(y)\|^2 = \sum_k (\phi_1(x)_k - \phi_1(y)_k)^2 =$$

$$= \sum_k (\phi_1(x)_k)^2 + \sum_k (\phi_1(y)_k)^2 - 2\sum_k (\phi_1(x)_k \phi_1(y)_k)$$

$$= \phi_1(x)^\top \phi_1(x) + \phi_1(y)^\top \phi_1(y) - 2\phi_1(x)^\top \phi_1(y)$$

$$= K_1(x, x) + K_1(y, y) - 2K_1(x, y)$$

$$= e^{-\lambda_1 \|x-x\|^2} + e^{-\lambda_1 \|y-y\|^2} - 2e^{-\lambda_1 \|x-y\|^2}$$

$$= 2 - 2e^{-\lambda_1 \|x-y\|^2}$$

As result:

$$K(x, y) = e^{-\lambda_2 \|\phi_1(x)-\phi_2(y)\|^2} = e^{-\lambda_2(2-2e^{-\lambda_1 \|x-y\|^2})}$$

$$= e^{-2\lambda_2} e^{2\lambda_2 e^{-\lambda_1 \|x-y\|^2}} = \alpha e^{\beta e^{-\lambda_1 \|x-y\|^2}}$$

Problem 3. Розглянемо нейронну мережу з активаційною функцією сигмоїди на прихованому шарі (hidden layer):

$$g(x) = \frac{1}{1 + e^{-x}}$$

Покажіть, що існує нейронна мережа з активаційною функцією гіперболічного тангенсу, яка обраховує таку саму функцію, що і перша мережа.

$$tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$

Solution to the problem 3.

Let's do some rewriting for $tanh(x)$

$$tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} = \frac{e^x + e^x - e^x - e^{-x}}{e^x + e^{-x}} = 2\frac{e^x}{e^x + e^{-x}} - 1$$
$$= 2\frac{e^x e^{-x}}{e^x e^{-x} + e^{-x} e^{-x}} - 1 = 2\frac{1}{1 + e^{-2x}} - 1$$

As we can see, we received a modified version of sigmoid function (scaled and shifted). Now we can represent the $tanh(x)$ as:

$$tanh(x) = 2\frac{1}{1 + e^{-2x}} - 1 = 2g(2x) - 1$$

The transformations used there are linear. From this we can say that the neural network can use any of these two activation functions and learn the same target function.

However, the learning process is a bit different because *tanh* provides stronger gradients (that can speed up learning, assuming the normalized data – the derivatives for the values close to zero would be higher), and it is centered around zero avoiding bias in the gradients.