EE5605 : Kernel Methods, Spring 2022

CS21MDS14025

Souradeep Debnath

Q1: PROVE THAT :

$$\tilde{K} = K - 2\mathbb{1}_{1/n} K + \mathbb{1}_{1/n} K \mathbb{1}_{1/n}$$

SOLUTION: We know, for $\phi : \mathcal{X} \to \mathcal{H}$

Kernel PCA is given by

$$C_\phi = \frac{1}{n}\sum_{i=1}^{n} \phi(x_i)\,\phi(x_i)^T$$

and K matrix is defined as

$$K_{il} = k(x_i, x_l) = \langle \phi(x_i), \phi(x_l) \rangle$$

Now, in the above formulation, we had

assumed, $$\sum_{i=1}^{n} \phi(x_i) = 0$$

But what happens if the above assumption is violated ?

Then, we would need to center the mapped features, i.e. we calculate new $\tilde{\phi}(x_K)$ such

that $$\tilde{\phi}(x_i) = \phi(x_i) - \frac{1}{n}\sum_{K=1}^{n} \phi(x_K)$$

Correspondingly, we get a new kernel $\tilde{k}$ such that

$$\tilde{k}(x_i, x_j) = \tilde{\phi}(x_i)^T \tilde{\phi}(x_j)$$

$$= \left[\phi(x_i) - \frac{1}{n}\sum_{k=1}^{n} \phi(x_k)\right]^T \left[\phi(x_j) - \frac{1}{n}\sum_{k=1}^{n} \phi(x_k)\right]$$

$$\Rightarrow \left[\phi^T(x_i) - \frac{1}{n}\sum_{k=1}^{n} \phi^T(x_k)\right]\left[\phi(x_j) - \frac{1}{n}\sum_{k=1}^{n} \phi(x_k)\right]$$

$$\left[\because\ A^T - B^T = (A-B)^T\right]$$

$$= \phi^T(x_i)\,\phi(x_j) - \frac{1}{n}\sum_{k=1}^{n} \phi^T(x_i)\,\phi(x_k) - \frac{1}{n}\sum_{k=1}^{n} \phi^T(x_k)\,\phi(x_j) + \frac{1}{n^2}\sum_{k=1}^{n} \phi^T(x_k)\,\phi(x_k)$$

$$= K(x_i, x_j) - \frac{1}{n}\sum_{k=1}^{n} K(x_i, x_k) - \frac{1}{n}\sum_{k=1}^{n} K(x_j, x_k) + \frac{1}{n^2}\sum_{l,k=1}^{n} K(x_l, x_k)$$

$$\longrightarrow (1)$$

Now suppose an arbitrary matrix

$$M = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$$ where $a, b, c, d \in \mathbb{R}$

$$\Rightarrow \frac{1}{n} M = \begin{bmatrix} a/n & b/n \\ c/n & d/n \end{bmatrix} \longrightarrow \text{(ii)}$$

Now $$\begin{bmatrix} \frac{1}{n} & 0 \\ 0 & \frac{1}{n} \end{bmatrix} \begin{bmatrix} a & b \\ c & d \end{bmatrix} = \mathbb{1}_{1/n} M$$

$$= \begin{bmatrix} \frac{a}{n} + c \times 0 & \frac{b}{n} + d \times 0 \\ a \times 0 + c/n & b \times 0 + d/n \end{bmatrix}$$

$$= \begin{bmatrix} a/n & b/n \\ c/n & d/n \end{bmatrix} \longrightarrow \text{(iii)}$$

from (ii) & (iii) $\Rightarrow$

$$\boxed{\frac{1}{n} M = \mathbb{1}_{1/n} M} \longrightarrow \text{(iv)}$$

Now applying this result (iv) to eq (i)

$\cancel{\tilde{K}(x_i, x_j)}$

$$\tilde{K} = K - \mathbb{1}_{1/n} K - \mathbb{1}_{1/n} K + \mathbb{1}_{1/n} K \mathbb{1}_{1/n}$$

$$\Rightarrow \boxed{\tilde{K} = K - 2\,\mathbb{1}_{1/n} K + \mathbb{1}_{1/n} K \mathbb{1}_{1/n}}$$ HENCE PROVED

~~where~~

WHY is this useful ?

In the class, for the sake of simplicity we had assumed

$$\frac{1}{n}\sum_{k=1}^{n} \phi(x_k) = 0$$

In many cases, this assumption is FALSE.

Therefore in a general case, in stead of K we would need to use $\tilde{K}$ as derived earlier.

$\boxed{\text{ANSWER}}$

Q2 APPLY THE REPRESENTER THEOREM TO SOLVE KERNEL SVM

We know that for linear SVM, the Primal form of objective is:
we wanna minimize L_p where

$$L_p = \frac{1}{2}\|\beta\|^2 - \sum_{i=1}^{n} \alpha_i \left[y_i \left(\langle \beta, x_i \rangle + \beta_0 \right) - 1 \right] \quad \longrightarrow (1)$$

$x \in \mathbb{R}^d$, $\beta \in \mathbb{R}^d$, $\beta_0 \in \mathbb{R}$

β, β_0 parameters of the model

β^*, β_0^* comes from $\underset{\beta, \beta_0}{\operatorname{argmin}} L_p$.

Since this a convex optimization problem, I can simply set gradient $= 0$

$\nabla_\beta L_p = 0 \Rightarrow$

$$\Rightarrow \|\beta\| - \sum_{i=1}^{n} \alpha_i y_i x_i = 0$$

$$\Rightarrow \beta^* = \sum \alpha_i y_i x_i \quad \longrightarrow (2)$$

$$\nabla_{\beta_0} L_p = 0 \Rightarrow \sum \alpha_i y_i = 0 \quad \longrightarrow (3)$$

Now inputting (3) & (2) into (1) [L_p] we get our dual form

$$L_D = \frac{1}{2}\langle \beta^*, \beta^* \rangle + \sum_{i=1}^{n} \alpha_i - \sum_{i=1}^{n} \alpha_i y_i \langle \beta^*, x_i \rangle$$

$$= \frac{1}{2}\left(\left\langle \sum_{i=1}^{n} \alpha_i y_i x_i, \sum_{j=1}^{n} \alpha_j y_i x_j \right\rangle\right) + \sum_{i=1}^{n} \alpha_i$$

$$- \sum_{i=1}^{n} \alpha_i y_i \left\langle \sum_{j=1}^{n} \alpha_j y_j x_j, x_i \right\rangle$$

$$= \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\left(\sum_{i}\sum_{j} \alpha_i \alpha_j y_i y_j \langle x_i, x_j \rangle\right) \rightarrow (4)$$

We need to α s.t. L_D is Maximized.

Now since there is an inner product between datapoints, for getting the Kernel version of Dual, we can replace ~~ϕ~~ $\langle x_i, x_j \rangle$ with

$$\langle \phi(x_i), \phi(x_j) \rangle = K(x_i, x_j)$$

$\therefore$ L_D for SVM Kernel becomes

$$L_D = \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i}\sum_{j} \alpha_i \alpha_j y_i y_j k(x_i, x_j)$$

which is the desired form.