MA4270. Lecture 7.

Last time: Kernel.

$X \in \mathbb{R}^d$: Original feature vector / may not be linearly separable (Affinely) in $\mathbb{R}^d$.

- One way is SVM with slackness ⇓ Hinge loss
- Other way → Kernel.

$d=1$: Map $x \in \mathbb{R}^d \xrightarrow{\phi}$ (feature map) $\phi(x)$: e.g. polynomial kernel.

- $\phi(x) = [1, \sqrt{2}x, x^2] \longrightarrow$ quadratic
- $\phi(x) = [1, \sqrt{3}x, \sqrt{3}x^2, x^3] \longrightarrow$ cubic

yield non-linear regression function, e.g.

$$y = \theta_0 + \theta_1 \sqrt{4}x + \theta_2 \sqrt{6}x^2 + \theta_3 \sqrt{4}x^3 + \theta_4 x^4.$$

When $d=2$. $x = (x_1, x_2)^T \in \mathbb{R}^2 \longrightarrow \phi(x) = \begin{bmatrix} 1 \\ \sqrt{2}x_1 \\ \sqrt{2}x_2 \\ \sqrt{2}x_1x_2 \\ x_1^2 \\ x_2^2 \end{bmatrix}$

$$\langle \phi(x), \phi(z) \rangle = \begin{bmatrix} 1 \\ \sqrt{2}x_1 \\ \sqrt{2}x_2 \\ \sqrt{2}x_1x_2 \\ x_1^2 \\ x_2^2 \end{bmatrix}^T \begin{bmatrix} 1 \\ \sqrt{2}z_1 \\ \sqrt{2}z_2 \\ \sqrt{2}z_1z_2 \\ z_1^2 \\ z_2^2 \end{bmatrix}$$

High dimension $\in \mathbb{R}^6$

$(1 + x_1z_1 + x_2z_2)^3$

$$\phi(x) = \begin{pmatrix} 1 \\ \sqrt{3}x_1 \\ \sqrt{3}x_2 \\ \sqrt{3}x_1^2x_2 \\ \sqrt{3}x_2^2x_1 \\ x_2^3 \\ x_1^3 \end{pmatrix} \quad \begin{pmatrix} 1 \\ \sqrt{3}x_2 \\ \sqrt{3}x_1 \\ \sqrt{3}x_2^2 \\ \sqrt{3}x_1^2 \end{pmatrix}$$

$$= 1 + 2x_1z_1 + 2x_2z_2 + 2(x_1z_1)(x_2z_2) + (x_1z_1)^2 + (x_2z_2)^2$$

$$= (1 + x_1z_1 + x_2z_2)^2 = (1 + \langle \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}, \begin{pmatrix} z_1 \\ z_2 \end{pmatrix} \rangle)^2.$$

$$= (1 + \langle x, z \rangle)^p \rightsquigarrow \text{takes } O(dp) \text{ time.}$$

($\langle x, z \rangle$ → d; p → p)

Why do we consider Inner Product ?

→ Consider a kernelized linear regression model. ⇒ Just need the calculation of Inner product among $\{\phi(x_t)\}_{t=1}^n \subseteq \mathbb{R}^D$.

$$y = \theta^T \phi(x) + \varepsilon$$

feature map

Goal: Show that estimation of $\hat{\theta}$ using MLE ONLY involves the inner product among feature vectors $\phi(x_t)$

→ Regularized Least Square Prob.

$$J(\theta) = \sum_{t=1}^n (y_t - \theta^T \phi(x_t))^2 + \lambda \|\theta\|_2^2$$

↓ important.

Penalized log-likelihood Criterion

$$\frac{\partial J}{\partial \theta} = -2 \sum_{t=1}^n (y_t - \theta^T \phi(x_t)) \phi(x_t) + 2\lambda\theta = 0$$

$\alpha_t \rightsquigarrow$ prediction difference

$$\Rightarrow \hat{\theta} = \frac{1}{\lambda} \sum_{t=1}^n \alpha_t \cdot \phi(x_t) \in \mathbb{R}^D \rightsquigarrow \text{high dimension}$$

In order to determine $\hat{\theta}$ in $\mathbb{R}^d$, we only have to specify n numbers α_t $t = 1, 2, \ldots, n$

$$\Updownarrow$$

$$\hat{\theta} \in \text{span}\{\phi(x_t)\}_{t=1}^n$$

$\hat{\theta}$ is determined only by $\{\alpha_t\}_{t=1}^n$ instead of $\dim(\phi) \in \mathbb{R}^D$

$(D \gg d,\ D \gg n)$

$$\alpha = \lambda(\lambda I + K)^{-1} y$$

$$\Downarrow$$

$$\hat{\theta} = \frac{1}{\lambda} \sum_{t=1}^n \alpha_t \phi(x_t)$$

$$\Downarrow$$

prediction

$$y' = \hat{\theta}^T \phi(x') = \left[\frac{1}{\lambda} \sum_{t=1}^n \alpha_t \phi(x_t)\right]^T \phi(x') = \frac{1}{\lambda} \sum_{t=1}^n \alpha_t \langle \phi(x_t), \phi(x') \rangle$$

$\phi : \mathbb{R}^d \rightarrow \mathbb{R}^D$

$D \gg d$

Q: How to find α_t? ⇒ Answer { kernel matrix K.

$$\alpha_t = y_t - \underbrace{\theta^T \phi(x_t)}_{\text{prediction difference}} = y_t - \left[\frac{1}{\lambda}\sum_{s=1}^{n} \alpha_s \phi(x_s)\right]^T \phi(x_t)$$

$$= y_t - \frac{1}{\lambda}\sum_{s=1}^{n} \alpha_s \underbrace{\langle \phi(x_s), \phi(x_t)\rangle}_{K_{st}}$$

Note: α_t depends on y_t as well as the inner product

Def [Kernel matrix]: Given $\{x_t\}_{t=1}^n \subseteq \mathbb{R}^d$ $\phi: \mathbb{R}^d \to \mathbb{R}^p$

$$K = \begin{bmatrix} \langle \phi(x_1), \phi(x_1)\rangle & \cdots & \langle \phi(x_1), \phi(x_n)\rangle \\ \langle \phi(x_2), \phi(x_1)\rangle & & \\ \vdots & & \vdots \\ \langle \phi(x_n), \phi(x_1)\rangle & \cdots & \langle \phi(x_n), \phi(x_n)\rangle \end{bmatrix}_{n\times n}$$

$$= \Phi^T\Phi$$

$$[K]_{st} = \langle \phi(x_s), \phi(x_t)\rangle$$

Ex: Prove that K is positive semi-definite. → definition.

Now, $\alpha_t = y_t - \frac{1}{\lambda}\sum_{s=1}^{n} \alpha_s \langle \phi(x_t), \phi(x_s)\rangle$

$$= y_t - \frac{1}{\lambda}\sum_{s=1}^{n} \alpha_s K_{st} = K_{ts}.$$

$$\underline{\alpha} = \begin{pmatrix} \alpha_1 \\ \vdots \\ \alpha_n \end{pmatrix} \qquad \underline{y} = \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix}.$$

then we have:

$$\underline{\alpha} = \underline{y} - \frac{1}{\lambda} K\underline{\alpha} \Rightarrow \left(I + \frac{1}{\lambda}K\right)\underline{\alpha} = \underline{y}.$$

→ positive definite

$$\Rightarrow \hat{\alpha} = (I + \frac{1}{\lambda} K)^{-1} y \quad \text{(invert)}$$

$$= \lambda (\lambda I + K)^{-1} y$$

Punchline:

To solve α, we just need $\begin{cases} y \rightarrow \text{traget value.} \\ K \rightarrow \text{kernel matrix} \end{cases}$

$\Downarrow$

only involves **Inner Product** among $\{\phi(x_t)\}_{t=1}^n$

(2).

After finding the $\{\alpha_t\}_{t=1}^n$ ($\leftarrow \begin{cases} y \\ K \end{cases}$), how do we predict a target given a test sample x'?

$$y = \hat{\theta}^T \phi(x') = \underbrace{\left(\frac{1}{\lambda} \sum_{t=1}^n \alpha_t \phi(x_t)\right)^T}_{\hat{\theta}^T} \phi(x')$$

$$= \frac{1}{\lambda} \sum_{t=1}^n \alpha_t \langle \phi(x_t), \phi(x') \rangle$$

$$= \frac{1}{\lambda} \sum_{t=1}^n \alpha_t K(x_t, x') \quad \rightarrow \text{prediction}$$

Calculate $K(x_t, x')$ for all $t = 1, 2, \ldots, n$

$= \langle \phi(x'), \phi(x_t) \rangle$.

Conclusion: $\begin{cases} \text{finding } \{\alpha_t\}_{t=1}^n \\ \text{prediction given sample } x' \end{cases}$

$\Rightarrow$ Both need $\langle \phi(\cdot), \phi(\cdot) \rangle$

Def: A (valid) kernel function $K: \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ (can be abstract)

is a function s.t $\exists$ a feature map $\phi: \mathbb{R}^d \to \mathbb{R}^D$

s.t $\forall x, x' \in \mathbb{R}^d$.

$$K(x, x') = \langle \phi(x), \phi(x') \rangle$$

Actually we don't care this!

↳ if this value is large, then x & x' are similar!

[Rmk]: While we have written the original and kernelized feature space as $\mathbb{R}^d$ & $\mathbb{R}^D$, neither of the spaces need to be finite dimension.

Prop: Let K_1, K_2 ($\mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$) be valid kernel functions. Let $f: \mathbb{R}^d \to \mathbb{R}$ be an arbitrary func. Then the following are valid kernel func.

i) $K(x, x') = f(x) K_1(x, x') f(x')$

ii) $K(x, x') = K_1(x, x') + K_2(x, x')$

iii) $K(x, x') = K_1(x, x') K(x, x')$

Intuition

$\begin{cases} \phi^{(1)}: \mathbb{R}^d \to \mathbb{R}^{D_1} \\ \phi^{(2)}: \mathbb{R}^d \to \mathbb{R}^{D_2} \end{cases}$ $\phi^{(1)}(x)\,\phi^{(2)}(x)$ $\mathbb{R} \times \mathbb{R}$

$\sum_{D_1}$ $\sum_{D_2}$

$\langle \phi^{(1)}(x), \phi^{(1)}(x') \rangle \langle \phi^{(2)}(x), \phi^{(2)}(x') \rangle$

$\langle \phi(x), \phi(x') \rangle$

$\sum_{D_1 D_2} \phi^{(i)}(x)\, \phi^{(j)}(x') \sum \phi^{(i)}(x')$

Pf: Let $\phi^{(1)}$ $\phi^{(2)}$ be the feature map associated to K_1 & K_2

$\Leftrightarrow \forall i = 1, 2, \quad K_i(x, x') = \langle \phi^{(i)}(x), \phi^{(i)}(x') \rangle$

i). Let $\phi(x) = \phi^{(1)}(x)\, f(x) : \mathbb{R}^d \to \mathbb{R}^D$

↳ scaler.

Goal: prove ϕ is the feature map of K

$$\langle \phi(x), \phi(x') \rangle = \langle \phi^{(1)}(x), \phi^{(1)}(x') \rangle \cdot f(x) \cdot f(x')$$

$$= f(x) \langle \phi^{(1)}(x), \phi^{(1)}(x') \rangle f(x')$$

$$= f(x)\, K_1(x, x')\, f(x')$$

$$= K(x, x')$$

$\Rightarrow$ K is a valid kernel function.

ii) Let $\phi(x) = \begin{bmatrix} \phi^{(1)}(x) \\ \phi^{(2)}(x) \end{bmatrix} : \mathbb{R}^d \longrightarrow \mathbb{R}^{D}$

$$\langle \phi(x), \phi(x') \rangle = \left\langle \begin{bmatrix} \phi^{(1)}(x) \\ \phi^{(2)}(x) \end{bmatrix}, \begin{bmatrix} \phi^{(1)}(x') \\ \phi^{(2)}(x') \end{bmatrix} \right\rangle$$

$$= \langle \phi^{(1)}(x), \phi^{(1)}(x') \rangle + \langle \phi^{(2)}(x), \phi^{(2)}(x') \rangle$$

$$= K_1(x, x') + K_2(x, x')$$

iii) $K(x, x') = K_1(x, x')\, K_2(x, x')$ is a (valid) kernel function

$$\phi(x) = \phi^{(1)}(x)\, \phi^{(2)}(x)^T : \mathbb{R}^d \longrightarrow \mathbb{R}^{D_1 \times D_2}$$

$$\langle \phi(x), \phi(x) \rangle = \sum_{i,j} \phi_{ij}(x) \cdot \phi_{ij}(x')$$

matrix inner product

$$= \sum_{ij} \phi_i^{(1)}(x)\, \phi_j^{(2)}(x)\, \phi_i^{(1)}(x')\, \phi_j^{(2)}(x')$$

$$= \left(\sum_i \phi_i^{(1)}(x)\, \phi_i^{(1)}(x') \right) \left(\sum_j \phi_j^{(2)}(x)\, \phi_j^{(2)}(x') \right)$$

$$= K_1(x, x')\; K_2(x, x')$$

Claim: The function $K : \mathbb{R}^d \times \mathbb{R}^d \longrightarrow \mathbb{R}$ given by:

$K(x,x') = \exp\left(-\frac{\beta}{2}\|x-x'\|^2\right)$ is a kernel.

Ex: Use the above Prop. to prove this!

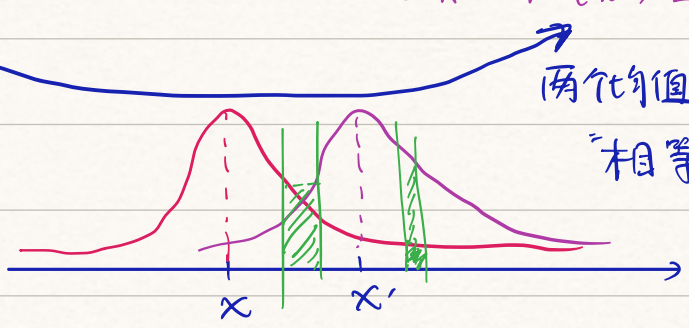
Rmk: This is known as the Radial basis kernel.

Roughly Speaking,

$K(x,x') \cong Pr(X \approx X')$ $\begin{cases} X \sim N(x, \frac{1}{2\beta}) \\ X' \sim N(x', \frac{1}{2\beta}) \end{cases}$

兩個的值分別是 x, x' 的 Gaussian r.v.

"相等" 的可能性。



$$= \lim_{\Delta \to 0} \int_{\mathbb{R}} Pr(X \in [z, z+\Delta], X' \in [z, z+\Delta])\, dz$$

Consider the functions $\phi_z(x) = C(\beta)\, N(z; x, \frac{1}{2\beta})$ $z \in \mathbb{R}$

$\phi : \mathbb{R}^d \longrightarrow$ ~~$\mathbb{R}^\infty$~~ Function Space

$\underline{x} \longrightarrow \phi_z(x) \Rightarrow$ a function of z.

$$\langle \phi(\underline{x}), \phi(\underline{x}') \rangle = \int_{\mathbb{R}} \phi_z(\underline{x})\, \phi_z(\underline{x}')\, dz = \exp\left(-\frac{1}{2\beta}(x-x')^2\right).$$

$\langle f, g \rangle = \int_{\mathbb{R}} f(t) g(t) dt$

$$= \int_{\mathbb{R}} C(\beta)^2 \underbrace{N(z; x, \frac{1}{2\beta})}_{Pr(X \approx z)} \underbrace{N(z; x', \frac{1}{2\beta})}_{Pr(X' \approx z)} dz$$

$$\propto \int_{\mathbb{R}} \exp(-\beta(z-x)^2) \exp(-\beta(z-x')^2)\, dz$$

NEED TO CHECK

$$= \exp(-\beta(x^2+(x')^2)) \int_{\mathbb{R}} \exp(-\beta z^2 + 2\beta z(x+x') - \beta z^2)\, dz$$

$$= \exp(-\beta(x^2+(x')^2)) \int_{\mathbb{R}} \exp(-\beta(2z^2 - 2z(x+x')))\, dz$$

$$= \exp\left(-\frac{\beta}{2}(x-x')^2\right) \propto K(x,x')$$

The feature maps for the RBK (Radial Basis Kernel) $\exp\left(-\frac{\beta}{2}\|x-x'\|^2\right)$ are normal prob. density function $N(\cdot; x, \frac{1}{2\beta})$

Conclusion:

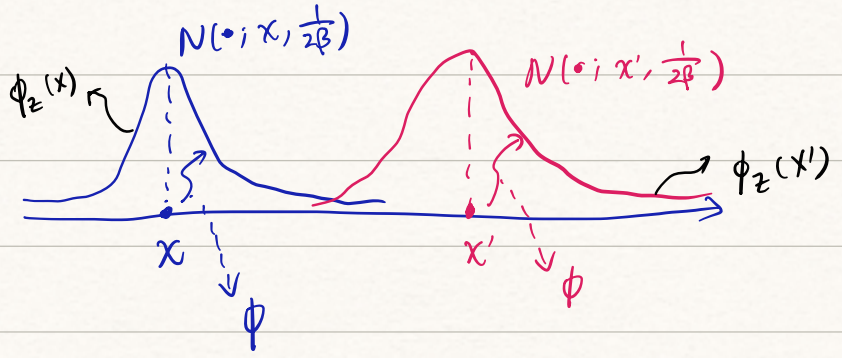
Origin: $\langle \phi(x'), \phi(x) \rangle = \int_z \phi_z(x)\, \phi_z(x')\, dz$

Now: we have $K(x,x') = \exp\left(-\frac{\beta}{2}\|x-x'\|^2\right)$

$\phi : \mathbb{R}^d \longrightarrow$ Function Space

$x \longmapsto \{N(z; x, \frac{1}{2\beta}) : z \in \mathbb{R}\} \triangleq \{\phi_z(x)\}$

this is the variable of function $\phi_z(x)$



String Kernels (Lodhi et al., JMLR 2012)

Training samples "car", "cat", "bat" & "bar"

Map each sample to a length-8 feature vector

Fix $\lambda \in (0,1)$, weighting factor : account for compactness of substring in a string

e.g. "car" is present in "castard" and "card" but is more compact in "card"

Substrings $k=2$. (Feature map)

	ca	ct	at	ba	bt	cr	ar	br
$\phi(\text{cat})$	λ^2	λ^3	λ^2	0	0	0	0	0
$\phi(\text{car})$	λ^2	0	0	0	0	λ^3	λ^2	0
$\phi(\text{bat})$	0	0	λ^2	λ^2	λ^3	0	0	0
$\phi(\text{bar})$	0	0	0	λ^2	0	0	λ^2	λ^3

$$K(\text{car}, \text{cat}) = \langle \phi(\text{car}), \phi(\text{cat}) \rangle = \lambda^4 \quad \text{(kernel)}$$

$$\hat{K}(\text{car}, \text{cat}) = \frac{K(\text{car}, \text{cat})}{\sqrt{K(\text{car}, \text{car})}\sqrt{K(\text{cat}, \text{cat})}} = \frac{\lambda^4}{2\lambda^4 + \lambda^6} = \frac{1}{2+\lambda^2}$$

$\Downarrow$ also kernel

String Subsequence kernel (SSK)

Σ : finite alphabet $\{a, b, \ldots, z\}$

s.t : strings - consisting of letters from Σ

$|s|$ $|t|$: length of strings

st: string formed by s & t.

u: subseq of s if $\exists\ (i_1, \dots, i_{|u|}) = \underline{i}$

$1 \le i_1 \le i_2 \le \dots \le i_{|u|} \le |s|$ s.t $u_j = s_{i_j}$ $\forall j = 1, 2, \dots, |u|$

$u = s[\underline{i}]$

$\ell(\underline{i}) = i_{|u|} - i_1 + 1$

[Eg:] s = segmentation u = set

$\underline{i} = (1, 2, 7)$ $\ell(\underline{i}) = 7 - 1 + 1 = 7$

$$F_n = \mathbb{R}^{|\Sigma^n|} \qquad \phi_u(s) = \sum_{\underline{i}: u = s[\underline{i}]} \lambda^{\ell(\underline{i})}$$

↑ Cartesian product

[Eg:] s = homolulu u = lu

$\phi_u(s) = 2\lambda^2$

$$K_n(s,t) = \sum_{u \in \Sigma^n} \langle \phi_u(s), \phi_u(t) \rangle$$

$$= \boxed{\sum_{u \in \Sigma^n}} \sum_{\underline{i}: s[\underline{i}] = u} \sum_{\underline{j}: t[\underline{j}] = u} \lambda^{\ell(\underline{i}) + \ell(\underline{j})}$$

$\Downarrow$

$O(|\Sigma|^n)$

[Rmk].

This exponential dependence on n can be reduced to linear

in n by using dynamic programing

⇒ Trick : $K_n = f(K_{n-1})$