DSA5105 Lec2

Recall → ① Linear Reg

Linear Basis Regress

↳ $\phi: x \in \mathbb{R}^d \mapsto \phi(x) \in \mathbb{R}^p$

② Regularization $\begin{cases} \ell_1\text{-norm}: \text{LASSO} \\ \ell_2\text{-norm}: \text{Ridge} \end{cases}$

③ Apply linear model to classification task

Architecture

$x \xrightarrow{\phi(\cdot)} \phi(x) \in \mathbb{R}^M \xrightarrow{W \in \mathbb{R}^{k \times M}} W \cdot \phi(x)$ ↓ softmax → probability

↑

① change output activation function

↳ Softmax

② change Loss function

↳ CROSS-ENTROPY LOSS

☆ Equivalent

can view as

$\phi: \mathbb{R}^d \to \mathbb{R}^M$ + projection to $\mathbb{R}^K$

Feature Maps

→ For Reg, we use $\mathcal{H}_1 = \{ f: f(x) = \sum w_j \phi_j(x) \}$

→ For Classification, we use $\mathcal{H}_2 = \{ f: f(x) = g\left(\sum_{j=1}^{M} w_j \phi_j(x)\right) \}$

and $\phi_j: x \mapsto \phi_j(x) \in \mathbb{R}$

$w_j \in \mathbb{R}^K$

most commonly use one

$g(\cdot) \Rightarrow$ Softmax function to generate a prob. dist.

output of the model is a K-class probability

In general, $g(z_k) = \dfrac{h(z_k)}{\sum h(z_k)}$, $h > 0$

For classification task

Model Training

Notation

$\mathbb{R}^{M \times N} \ni \Phi = \begin{pmatrix} \phi_1(x_1), \dots, \phi_1(x_N) \\ \vdots \\ \phi_M(x_1) \dots \phi_M(x_N) \end{pmatrix}$

output ground-truth

① Loss function : $L_{CE}(y', y) = -\sum_{k=1}^{K} y_k \log y'_k$

$W = \begin{pmatrix} w_1^T \\ \vdots \\ w_M^T \end{pmatrix} \in \mathbb{R}^{M \times K}$

↓ empirical risk minimization

ERM → $\min_{W \in \mathbb{R}^{M \times K}} R_{emp}(W) = \min_W \frac{1}{N} \sum_{i=1}^{N} L(g(W^T \Phi)_i, y_i)$

the probability prediction for

② Training Acc vs Testing Acc

i-th data

$L(g(W^T\phi(x_i)), y_i)$

Model traning objective is based on traing set

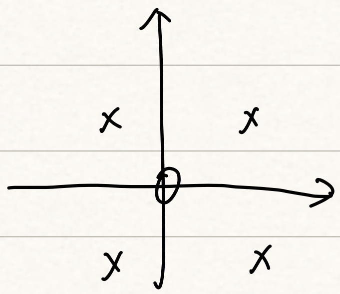
min : objective (traing set)

SS auxilary ?

$\boxed{\text{make all prediction of traing sample correct}}$

Today

① Importance of feature Map

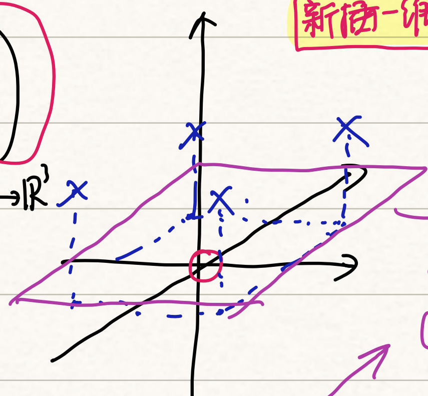


works for Radial-shaped data

$\boxed{\text{新增一维分层}}$ (idea)

$\phi_1 = \begin{pmatrix} x_1 \\ x_2 \\ \|x\| \end{pmatrix}$

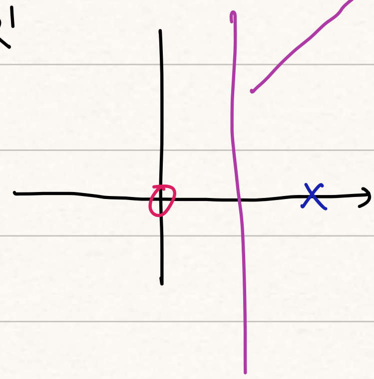
$\phi_1(\cdot) : \mathbb{R}^2 \to \mathbb{R}^3$



hyperplane to separate 2 classes (2 clusters)

$\phi_2(\cdot) : \mathbb{R}^2 \to \mathbb{R}^1$

$\phi_2 = \|\cdot\|_2$



② feature map ⟶ similarity meaure ⟵ $\boxed{\text{KERNEL}}$

$\langle \phi_i(x), \phi_j(x) \rangle$

③ $\boxed{\text{Ridge Regression}}$

Model: $f(x) = \sum_{j=0}^{M-1} w_j \phi_j(x)$ $\qquad \phi_j : \mathbb{R}^d \to \mathbb{R}$

$\boxed{\phi : \mathbb{R}^d \to \mathbb{R}^M}$

$$\hat{w} = (\phi^T\phi + \lambda I_M)^{-1}\phi^T y$$

$O(M)$ Computational Cost

$y = \Phi w$

$\Phi = \begin{pmatrix} \phi^T(x_1) \\ \vdots \\ \phi(x_N)^T \end{pmatrix} \in \mathbb{R}^{N \times M}$

→ **Punchline ⇒ Reformulation of Ridge Reg**

$\hat{w} = (\Phi^T\Phi + \lambda I_M)^{-1}\Phi^T y$ ← solution of Ridge Reg

Model: $\hat{f}(x) = w^T\phi(x)$

Let's show $\boxed{\hat{w} = \Phi^T(\Phi\Phi^T + \lambda I_N)^{-1} y}$

$$\Leftrightarrow \Phi^T(\Phi\Phi^T + \lambda I_N)^{-1} y = (\Phi^T\Phi + \lambda I_M)^{-1}\Phi^T y$$

$$\Leftrightarrow (\Phi^T\Phi + \lambda I_M)\Phi^T(\Phi\Phi^T + \lambda I_N)^{-1} y = \Phi^T y$$

$$\longrightarrow \Phi^T\Phi\Phi^T + \lambda I_M\Phi^T$$

$$= \Phi^T\Phi\Phi^T + \lambda\Phi^T I_N$$

$$= \Phi^T(\Phi\Phi^T + \lambda I_N)$$

$$\Leftrightarrow \Phi^T y = \Phi^T y \quad ✓$$

$\boxed{\Phi^T \in \mathbb{R}^{M \times N}}$

$\boxed{\Phi = \begin{pmatrix} \phi(x_1)^T \\ \vdots \\ \phi(x_N)^T \end{pmatrix} \in \mathbb{R}^{N \times M}}$

Thus $\hat{w} = \Phi^T(\Phi\Phi^T + \lambda I_N)^{-1} y$

$$\rightarrow \hat{f}(x) = \hat{w}^T\phi(x)$$

$$= \phi(x)^T\hat{w}$$

$$= \phi(x)^T\Phi^T(\Phi\Phi^T + \lambda I_N)^{-1} y$$

$$= (\Phi\phi(x))^T\alpha \quad \text{where } \alpha = (\Phi\Phi^T + \lambda I_N)^{-1} y$$

$$= \left[\begin{pmatrix} \phi(x_1)^T \\ \vdots \\ \phi(x_N)^T \end{pmatrix} \cdot \phi(x) \right]^T \alpha$$

$\boxed{\alpha \in \mathbb{R}^N}$

$$= \begin{bmatrix} \phi(x_1)^T \phi(x) \\ \vdots \\ \phi(x_N)^T \phi(x) \end{bmatrix}^T \alpha$$

$$= \sum_{i=1}^{N} \alpha_i \langle \phi(x_i), \phi(x) \rangle \Longrightarrow \boxed{\text{Dual Form of SVM}}$$

$\downarrow$ new interpretation of predict function

$\downarrow$ $\boxed{\text{based on similarity of feature map}}$

$\boxed{\text{complexity of } O(N)}$

recall: standard form of Ridge Reg $\rightarrow O(M)$

$$\hat{f}(x) = \sum_{j=0}^{M-1} \hat{w}_j \phi_j(x) \rightarrow \begin{cases} \text{no need of Data } x_1, \dots, x_n \\ O(M) \end{cases}$$

$$\hat{f}(x) = \sum_{i=1}^{N} \alpha_i \langle \phi(x_i), \phi(x) \rangle \quad \begin{cases} \text{need } x_1, \dots, x_N \text{ to compute similarity} \\ O(N) \end{cases}$$

$\alpha = (\Phi \Phi^T + \lambda I_N)^{-1} y$ $\leftarrow$ depends on feature map ONLY THROUGH $\boxed{\phi(x_j)^T \phi(x)}$

$\Phi = \begin{pmatrix} \phi(x_1)^T \\ \vdots \\ \phi(x_N)^T \end{pmatrix} \in \mathbb{R}^{N \times M}$

$\downarrow$

$\Longrightarrow$ can define directly

$K(x_j, x) \rightsquigarrow$ kernel function

$\downarrow$

similarity measure

and $\Phi \Phi^T := G$

$\downarrow$

$\boxed{G_{ij} = \langle \phi_i(x), \phi_j(x) \rangle}$

define $K(x, y)$ directly

$$\boxed{\begin{aligned} \hat{f}(x) &= \sum \alpha_i K(x_i, x) \\ \alpha &= (G + \lambda I_N)^{-1} y \end{aligned}}$$

$\boxed{G_{ij} = K(x_i, x_j)}$

How to choose a valid kernel? $\boxed{K(\cdot,\cdot)}$

① Necessary Condition
- 1. symmetry
- ~~2. Non-negative $K(x,x) \geq 0$~~
- 3. PSD for arbitrary choice of $(x_1, \ldots, x_n)$

② Mercer's THM tells us this is also Sufficient:

if $K(\cdot,\cdot)$ is SPD, then there exists $\varphi: \mathbb{R}^d \longrightarrow \mathbb{H}$

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

$\boxed{\text{IDEA:}}$ if we want to do Linear BASIS Ridge Reg.

we may not need to define feature map $\varphi(\cdot)$,

instead, we select a VALID kernel $K(\cdot,\cdot)$

and do: $\hat{f}(x) = \sum \alpha_i K(x, x_i)$

$$\alpha = (\Phi \Phi^T + \lambda I_N)^{-1} y \qquad !!!$$

Kernels & Feature Maps

① polynomial kernel

$$K(x, x') = (1 + x^T x')^2$$

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

② RBF kernel / Gaussian kernel

$$K(x,y) = \exp\left(-\frac{\|x-y\|^2}{\alpha}\right) \qquad x, y \in \mathbb{R}^d$$

Taylor:

$$\boxed{\exp(z) = \sum_{0}^{\infty} \frac{1}{k!} z^k}$$

$$\Downarrow$$

$$\exp\left(\frac{2xy}{\alpha}\right)$$

$$\sum_{0}^{\infty} \frac{1}{k!} \left(\frac{2xy}{\alpha}\right)^k$$

$$= \sum_{0}^{\infty} \frac{2^k}{k!\,\alpha^k} (x)^k (y)^k$$

$$= \exp\left(-\frac{x^2}{\alpha}\right) \exp\left(-\frac{y^2}{\alpha}\right) \exp\left(\frac{2xy}{\alpha}\right)$$

$$= \exp\left(-\frac{x^2}{\alpha}\right) \exp\left(-\frac{y^2}{\alpha}\right) \sum_{0}^{\infty} \frac{2^k}{\alpha^k \cdot k!} (x)^k (y)^k$$

$$= \sum_{k=0}^{\infty} \frac{2^k}{\alpha^k k!} \left(\exp\left(-\frac{x^2}{\alpha}\right) x^k\right) \left(\exp\left(-\frac{y^2}{\alpha}\right) y^k\right)$$

$$= \phi(x)^T \phi(y)$$

$$\phi(x) = \begin{pmatrix} \phi_0(x) \\ \vdots \\ \phi_n(x) \\ \vdots \end{pmatrix} \quad \text{where} \quad \boxed{\phi_i(x) = \sqrt{\frac{2^i}{\alpha^i \cdot i!}} \exp\left(-\frac{x^2}{\alpha}\right) \cdot x^i}$$

Gaussian feature map $\boxed{\varphi_{RBF} : \mathbb{R}^{d'} \longrightarrow \mathbb{R}^{\infty}}$

Kernel Recap

① Why we need kernel ?

→ Model (Kernel Ridge Regression)

$$\hat{f}(x) = \hat{w}^T \phi(x) \qquad \hat{w} = (\Phi^T \Phi + \lambda I_M)^{-1} \Phi^T y$$

$$= \Phi^T (\Phi \Phi^T + \lambda I_N)^{-1} y$$

$$= \Phi^T (G + \lambda I_N)^{-1} y$$

$$\Downarrow$$

$$\hat{f}(x) = (\Phi^T (G + \lambda I_N)^{-1} y)^T \phi(x)$$

$$= \phi(x)^T \Phi^T \cdot (G + \lambda I_N)^{-1} y$$

$$= \begin{pmatrix} \phi(x_1)^T \phi(x) \\ \vdots \\ \phi(x_n)^T \phi(x) \end{pmatrix}^T \cdot (G + \lambda I_N)^{-1} y$$

$$= \sum_{i=1}^{N} \alpha_i \, \phi(x_i)^T \phi(x) \qquad \boxed{\alpha = (G + \lambda I_N)^{-1} y}$$

$$:= \sum_{i=1}^{N} \alpha_i \, k(x_i, x)$$

② What can construct a valid kernel $k(x,y)$?

[E.g.] $k(x,y) = (1 + x^T y)^k$ if $x, y \in \mathbb{R}^d$

$$(1 + x_1 y_1 + \cdots + x_d y_d)^k = \sum_{\substack{k_0 + k_1 + \cdots + k_d = k \\ k_i \geq 0}} \binom{k}{k_0 \cdots k_d} \prod_{i=1}^{d} (x_i y_i)^{k_i}$$

$$= \begin{pmatrix} 1 \\ \sqrt{k}\, x_1 \\ \vdots \\ \sqrt{k}\, x_d \\ \vdots \\ \sqrt{\binom{k}{k_0 \cdots k_d}} \prod_{i=1}^{d} x_i^{k_i} \\ \vdots \end{pmatrix} \begin{pmatrix} 1 \\ \sqrt{k}\, y_1 \\ \vdots \\ \sqrt{k}\, y_d \\ \vdots \\ \sqrt{\binom{k}{k_0 \cdots k_d}} \prod_{i=1}^{d} y_i^{k_i} \\ \vdots \end{pmatrix} \Bigg\} \binom{k+d}{d}$$

$$\Rightarrow \phi(x) = \begin{pmatrix} \vdots \\ \sqrt{\binom{k}{k_0 \cdots k_d}} \prod_{i=1}^{d} x_i^{k_i} \\ \vdots \end{pmatrix} \in \mathbb{R}^{\binom{k+d}{d}}$$

☆ **Mercer's theorem**

$\boxed{k(x,y)} \to$ SPD kernel $\Longleftrightarrow$ the existence of feature map s.t $K(x,y) = \langle \phi(x), \phi(y) \rangle$

$\downarrow$

Defn of SPD kernel $k(x,y) : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$

$\Rightarrow$ consider $\begin{cases} k(x,y) \\ \{x_1, \dots, x_n\} \end{cases}$ $\Rightarrow$ matrix $K \in \mathbb{R}^{n \times n}$ and $K_{ij} = \langle \phi(x_i), \phi(x_j) \rangle$

$[\forall n$ and $\forall x_1, \dots, x_n]$

$\downarrow$

satisfy $\begin{cases} ① \text{ symmetric} \\ ② \text{ PSD} \end{cases}$

$\boxed{\text{Kernel trick}}$

③ Avoid introducing feature map, we can define a SPD kernel $K(x,y)$ to apply Kernel Ridge Regression Method and so on.

$\boxed{\text{Question}}$: How can we verify $K(x,y)$ is SPD ?

④ $\boxed{\text{Verify SPD Kernel}}$

a) through definition: [for simple SPD kernel]

→ for $\forall n$ & $\forall x_1, \ldots, x_n$, consider $K_{ij} = k(x_i, x_j)$

check K is $\begin{cases} \text{symmetric} \\ \text{PSD} \end{cases}$

(RBF kernel)

b) use SPD Kernel Closure Property: [for complicate SPD Kernel]

① scaling property: $k(x,y) = \lambda k_1(x,y)$

② addition property: $k(x,y) = k_1(x,y) + k_2(x,y)$

③ normalization property: $k(x,y) = g(x)\, k_1(x,y)\, g(y)$

④ limit property: $k(x,y) = \lim_{n \to \infty} k_n(x,y)$

⑤ product property: $k(x,y) = k_1(x,y)\, k_2(x,y)$