Kernel Ridge Regression & The Kernel trick

Least Squares:

$$\min_w \sum_{i=1}^{N} (y_i - x_i^T w)^2 + \frac{\lambda}{2}\|w\|_2^2 \quad \text{(ridge. Reg)}$$

closed form:

$$w^* = (X^T X + \lambda I_d)^{-1} X^T y$$

$X \in \mathbb{R}^{N \times d}$; $(X^T X + \lambda I_d)$: $d \times d$

Proof: $P \in \mathbb{R}^{M \times N}$ and $Q \in \mathbb{R}^{N \times M}$

$P(QP+I) = PQP + P = (PQ+I)P$

Assume that $(QP+I)$ and $(PQ+I)$ are invertible:

$(PQ+I)^{-1} P = P (QP+I)^{-1}$

for $P = X^T$ and $Q = \frac{1}{\lambda} X$

$\Rightarrow (\frac{1}{\lambda} X^T X + I)^{-1} X^T = X^T (\frac{1}{\lambda} X X^T + I)^{-1}$

$\Rightarrow (X^T X + \lambda I)^{-1} X^T = X^T (X X^T + \lambda I)^{-1}$

$$X = \begin{pmatrix} x_1^T \\ x_2^T \\ x_N^T \end{pmatrix} \begin{matrix} \leftarrow \text{feature} \\ \uparrow \text{col} \end{matrix} \quad N \times D$$

$$X^T = \begin{pmatrix} x_1 & x_2 & \cdots & x_N \end{pmatrix} \begin{matrix} \leftarrow \text{feature} \\ \uparrow \text{rows} \end{matrix} \quad D \times N$$

Alternative:

we will show:

$$w^* = X^T \underbrace{(\overbrace{X X^T}^{K \text{ "Kernel"}} + \lambda I_N)^{-1}}_{N \times N} y \quad \text{(Ridge Regression)}$$

$D \times N$

Why the alternative form:

1) Complexity:

- $(X^T X + \lambda I_d)^{-1} X^T y : O(d^3 + N d^2)$
- $X^T (X X^T + \lambda I_N)^{-1} y : O(N^3 + d N^2)$

2) Structure:

$$w^* = X^T \cdot \alpha_* \quad \text{where } \alpha_* = (X X^T + \lambda I_N)^{-1} y \rightarrow (N \times 1)$$

$$w^* = \sum \alpha_{*i} x_i$$

w^* is a comb. of the feature vectors

$w^* \in$ The space spanned by the feature vectors (Column space of X^T)

Representer theorem:

For any loss function $\mathcal{L}$, if $w^* = \min \sum_{i=1}^{N} \mathcal{L}(x_i^T w, y_i) + \frac{\lambda}{2}\|w\|_2^2$

then there exists $\alpha_* \in \mathbb{R}^N$ s.t. $\boxed{w^* = X^T \alpha_*}$

Kernelized Ridge Regr.

$$w^* = \arg\min_w \frac{1}{2}\|y - Xw\|_2^2 + \frac{\lambda}{2}\|w\|_2^2$$

Alternative form.:

$$\boxed{\alpha_* = \arg\min \frac{1}{2} \alpha^T (X X^T + \lambda I) \alpha - \alpha^T y}$$

$$\Rightarrow w^* = X^T \cdot \alpha_*$$

Proof:

By setting the gradient to 0, we obtain: $\alpha_* = (X X^T + \lambda I)^{-1} y$

and by the repr. th. we get $w^* = X^T \alpha_*$

Why is this of Interest?

- complexity (N vs D): if $N \gg D$ → Normal sol: if $N \ll D$ → Alt. sol:
- $K = X X^T$: Kernel Matrix → Kernel trick

Kernel function:

Feature map. $\phi : \mathbb{R}^d \rightarrow \mathbb{R}^{\tilde{d}}$ with $d \ll \tilde{d}$; $x_i \mapsto (\phi(x_i))_{i=1}^N$

$$K = \begin{pmatrix} x_1^T x_1 & x_1^T x_2 & \cdots \\ x_2^T x_1 & \cdots & \cdots \\ \cdots & \cdots & \cdots \end{pmatrix} = (x_i^T x_j)_{i,j} \quad N \times N$$

$$K = \begin{pmatrix} \phi(x_1)^T\phi(x_1) & \cdots & \phi(x_1^T)\phi(x_m) \\ \vdots & \ddots & \vdots \\ \phi(x_m^T)\phi(x_1) & \cdots & \phi(x_m^T)\phi(x_m) \end{pmatrix} \in \mathbb{R}^{N \times N}$$

Kernel fct II : $K(x,x') = \phi^T(x)\phi(x')$ (happy) $\longrightarrow O(d)$

Kernel trick: $K = (K(x_i, x_j))_{ij}$

Ex : 1) $K(x,x') = x^T x' = \phi^T(x)\phi(x') \Rightarrow \phi(x) = x$

2) $x, x' \in \mathbb{R}$, $K(x,x') = (xx')^2 \Rightarrow \phi(x) = x^2$, $\phi^T(x)\phi(x') = x^2 x'^2 = (xx')^2$

3) $x, x' \in \mathbb{R}^3$ $K(x,x') = (x_1x_1' + x_2x_2' + x_3x_3')^2 = \phi^T(x)\phi(x)$

$x = (x_1, x_2, x_3)$, $x' = (x_1', x_2', x_3')$

$\Rightarrow K(x,x') = (x_1x_1')^2 + (x_2x_2')^2 + (x_3x_3')^2 + 2x_1x_2x_1'x_2' + 2x_1x_3x_1'x_3' + 2x_2x_3x_2'x_3'$

$$\phi(x) = (x_1^2 \; x_2^2 \; x_3^2 \; \sqrt{2}x_1x_2 \; \sqrt{2}x_1x_3 \; \sqrt{2}x_2x_3)^T \in \mathbb{R}^6$$

$K(x,x') = \phi^T(x)\phi(x)$ (Polynomial Kernel)

4) **Radial basis fct :** $x, x' \in \mathbb{R}^d$

$K(x,x') = e^{-(x-x')^T(x-x')}$ $\quad O(d)$

2d assume $x, x' \in \mathbb{R}$

$\Rightarrow K(x,x') = e^{-(x-x')^2}$

$= e^{-x^2 - x'^2 + 2xx'}$

$= e^{-x^2} e^{-x'^2} e^{2xx'}$

$= e^{-x^2} e^{-x'^2} \sum_{k=0}^{+\infty} \frac{2^k x^k x'^k}{k!}$ (Taylor exp.)

$\phi(x) = e^{-x^2}\left(\ldots, \frac{\sqrt{2}^k}{\sqrt{k!}} x^k, \ldots\right)$

$\phi^T(x)\phi(x) = K(x,x')$ (inf. dim.)

Building new kernels from old ones:

- k_1, ϕ_1 and k_2, ϕ_2:

$$k(x,x') = \alpha k_1(x,x') + \beta k(x,x') \qquad \alpha, \beta \geq 0$$
$$= \alpha \phi_1^T(x)\phi_1(x') + \beta \phi_2^T(x)\phi_2(x')$$
$$= \begin{pmatrix} \sqrt{\alpha}\phi_1(x) & \sqrt{\beta}\phi_2(x) \end{pmatrix} \begin{pmatrix} \sqrt{\alpha}\phi_1(x') \\ \sqrt{\beta}\phi_2(x') \end{pmatrix}$$
$$= \phi^T(x)\phi(x')$$
$$\phi(x) = \begin{pmatrix} \sqrt{\alpha}\phi_1(x) \\ \sqrt{\beta}\phi_2(x) \end{pmatrix} \in \mathbb{R}^{d_1+d_2}$$

- $k(x,x') = k_1(x,x') \cdot k_2(x,x')$

$$\phi \in \mathbb{R}^{d_1 \times d_2}; \quad k(x,x') = \phi_1^T(x)\phi_1(x')\,\phi_2^T(x)\phi_2(x')$$
$$= \phi^T(x)\phi(x')$$
$$\phi(x) = \begin{pmatrix} (\phi_1(x))_1\,\phi_2(x) \\ \vdots \\ (\phi_1(x))_{d_1}\,\phi_2(x) \end{pmatrix} \in \mathbb{R}^{d_1 d_2}$$

$$\phi^T(x)\phi(x') = \sum_{i=1}^{d_1} (\phi_1(x))_i \phi_2(x)^T (\phi_1(x'))_i \phi_2(x')$$
$$= \sum_{i=1}^{d_1} (\phi_1(x))_i (\phi(x'))_i \phi_2^T(x)\phi_2(x')$$
$$= \phi_1^T(x)\phi_1(x')\,\phi_2^T(x)\phi_2(x') = k(x,x')$$

- There exist a lot of other rules.

Mercer's condition: Given k (kernel fct), How do we know if $\exists\, \phi$ s.t.:

$$k(x,x') = \phi^T(x)\phi(x')$$

Mercer's condition:

- $k(x,x') = k(x',x) \quad \forall x, x'$
- $\forall N \geq 0, \forall (x_i)_{i=1}^N$, $K = (k(x_i,x_j))_{i,j=1}^N$ K must be positive semi definit (PSD)

Predicting using Kernel:

$w^* \rightsquigarrow \phi^T(x) w^*$. d can be very large

$\phi^T(X) = \left(\phi(x_1) \cdots \phi(x_N)\right)_{D \times N}$

How to predict only using the Kernel:

$w^* = X^T \alpha_*$

$$\phi^T(x) w^* = \phi^T(x) \phi(X)^T \alpha_*$$

$$\phi^T(x) w^* = \sum_{i=1}^{N} k(x, x_i) \alpha_{*i} \rightarrow \text{never depends on } d$$

Rq: $y = \phi^T(x) w^*$ linear pred in the augmented space

$= f_{w^*}(x)$ non-linear in the original space. (φ)