Kernel Method (SVD, Kernel PCA) (**PML** Ch. 3 & 5)
Machine Learning for Finance (FIN 570)

Instructor: Jaehyuk Choi

Peking University HSBC Business School, Shenzhen, China

2023-24 Module 3 (Spring 2024)

Linear regression in terms of kernel

- Reminded that the multivariate regression $y \sim X\boldsymbol{w}$:

$$\hat{\boldsymbol{y}} = \boldsymbol{X}\hat{\boldsymbol{w}} = \boldsymbol{H}\boldsymbol{y}, \quad \text{where} \quad \hat{\boldsymbol{w}} = \underbrace{\boldsymbol{S}\boldsymbol{X}^T\boldsymbol{y}}_{(p\times 1)},\ \boldsymbol{H} = \underbrace{\boldsymbol{X}\boldsymbol{S}\boldsymbol{X}^T}_{(N\times N)},\ \boldsymbol{S} = \underbrace{(\boldsymbol{X}^T\boldsymbol{X})^{-1}}_{(p\times p)}$$

- The estimation $\hat{y}_*$ for a new value $\boldsymbol{x}_*$ is obtained as

$$\hat{y}_* = \boldsymbol{x}_*\hat{\boldsymbol{w}} = \underbrace{\boldsymbol{x}_*\boldsymbol{S}\boldsymbol{X}^T}_{(1\times N)}\boldsymbol{y} = \sum_i \underbrace{(\boldsymbol{x}_*\boldsymbol{S}\boldsymbol{x}_i^T)}_{\text{scalar}}\, y_i = \sum_i \underbrace{\phi(\boldsymbol{x}_*)\phi(\boldsymbol{x}_i)^T}_{\text{inner product}}\, y_i = \sum_i \underbrace{K(\boldsymbol{x}_*, \boldsymbol{x}_i)}_{\text{kernel}}\, y_i$$

  where $\phi(\boldsymbol{x}) = \boldsymbol{x}\boldsymbol{S}^{1/2}$ is from $\mathbb{R}^p$ to $\mathbb{R}^p$.

- The kernel, $K(\boldsymbol{x}_*, \boldsymbol{x}_i)$,

$$K(\boldsymbol{x}_*, \boldsymbol{x}_i) = \boldsymbol{x}_*\boldsymbol{S}\boldsymbol{x}_i^T = \phi(\boldsymbol{x}_*)\phi(\boldsymbol{x}_i)^T$$

  is understood as the influence of a training sample $\boldsymbol{x}_i$ on a test sample $\boldsymbol{x}_*$.

- In linear regression, kernel is defined as the inner product between linear function $\phi(\boldsymbol{x})$.

Generalizing kernel

- Kernel does not need to use a linear feature map $\phi(\boldsymbol{x})$.
  E.g., polynomial function: $\phi(x) = (x, x^2, x^3, \cdots, x^d)$
- Kernel does not have to use an inner product as long as $K(\boldsymbol{x}, \boldsymbol{y})$ satisfy some conditions (e.g., higher value for close pair).
- Examples:
  - Polynomial kernel:

$$K(\boldsymbol{x} \in \mathbb{R}^2, \boldsymbol{y} \in \mathbb{R}^2) = (1 + \boldsymbol{x}\boldsymbol{y}^T)^2 = (1 + x_1 y_1 + x_2 y_2)^2 = \cdots$$
$$= (1, \sqrt{2}x_1, \sqrt{2}x_2, x_1^2, \sqrt{2}x_1x_2, x_2^2) \cdot (1, \sqrt{2}y_1, \cdots)$$
$$= \phi(\boldsymbol{x})\phi(\boldsymbol{y})^T, \quad \text{where} \quad \phi(\boldsymbol{x}) : \mathbb{R}^2 \to \mathbb{R}^4$$

  - Radial basis kernel (RBF):

$$K(\boldsymbol{x}, \boldsymbol{y}) = \exp\left(-\gamma \|\boldsymbol{x} - \boldsymbol{y}\|^2\right)$$

    The corresponding $\phi(\boldsymbol{x})$ exists, but is ∞-dimensional ($\mathbb{R}^p \to \mathbb{R}^\infty$).
  - Sigmoid kernel:

$$K(\boldsymbol{x}, \boldsymbol{y}) = \tanh\left(a\boldsymbol{x}\boldsymbol{y}^T + b\right)$$

Kernel PCA

- We extend PCA analysis to the feature map $\phi(\boldsymbol{x})$, but using the kernel $K(\boldsymbol{x}_i, \boldsymbol{x}_j)$ only (not $\phi(\cdot)$).
- The covariance matrix of $\phi(\boldsymbol{x})$ is given by

$$\boldsymbol{\Sigma} = \frac{1}{N}\phi(\boldsymbol{X})^T\phi(\boldsymbol{X}) \quad \text{assuming} \quad E(\phi(\boldsymbol{x})) = \boldsymbol{0}.$$

- A PCA direction $\boldsymbol{v}$ ($p \times 1$) and the eigenvalue λ satisfy

$$\lambda\boldsymbol{v} = \boldsymbol{\Sigma}\boldsymbol{v} = \frac{1}{N}\phi(\boldsymbol{X})^T\phi(\boldsymbol{X})\boldsymbol{v} \quad \Rightarrow \quad \boldsymbol{v} = \phi(\boldsymbol{X})^T\boldsymbol{a} \quad \text{for} \quad \boldsymbol{a} = \underbrace{\frac{1}{\lambda N}\phi(\boldsymbol{X})\boldsymbol{v}}_{(N\times 1)}$$

- Substituting $\boldsymbol{v}$ into $\lambda\boldsymbol{v} = \boldsymbol{\Sigma}\boldsymbol{v}$ and using $\boldsymbol{K} = \phi(\boldsymbol{X})\phi(\boldsymbol{X})^T$ ($N \times N$),

$$\phi(\boldsymbol{X})\left[\lambda\phi(\boldsymbol{X})^T\boldsymbol{a} = \frac{1}{N}\phi(\boldsymbol{X})^T\phi(\boldsymbol{X})\phi(\boldsymbol{X})^T\boldsymbol{a}\right]$$

$$\lambda N\boldsymbol{K}\boldsymbol{a} = \boldsymbol{K}^2\boldsymbol{a} \quad \Rightarrow \quad \lambda N\boldsymbol{a} = \boldsymbol{K}\boldsymbol{a}$$

- The vector $\boldsymbol{a}$ is an eigenvector of $\boldsymbol{K}$ with the eigenvalue λN.

- $\boldsymbol{K} = \phi(\boldsymbol{X})\phi(\boldsymbol{X})^T$ is called Gram matrix.
  $K_{ij} = K(\phi(\boldsymbol{x}_i), \phi(\boldsymbol{x}_j))$ is the kernel value between i-th and j-th samples.
- The PCA score (projection) of a new vector $\boldsymbol{x}_*$ on $\boldsymbol{v}$ is

$$y_* = \phi(\boldsymbol{x}_*)\boldsymbol{v} = \phi(\boldsymbol{x}_*)\phi(\boldsymbol{X})^T\boldsymbol{a} = \phi(\boldsymbol{x}_*)\sum_i \phi(\boldsymbol{x}_i)^T a_i = \sum_i K(\boldsymbol{x}_*, \boldsymbol{x}_i)a_i$$

- The derivation of $\boldsymbol{a}$ and the PCA score never use the function $\phi(\cdot)$.
- Compared to finding the eigenvector in the raw space, kernel PCA is heavier in computation.

$$\boldsymbol{\Sigma} = \frac{1}{N}\phi(\boldsymbol{X})^T\phi(\boldsymbol{X}) \quad (d \times d) \quad \text{versus} \quad \boldsymbol{K} = \phi(\boldsymbol{X})\phi(\boldsymbol{X})^T \quad (N \times N)$$

- Because $E(\phi(\boldsymbol{x})) \neq 0$, we obtain $\boldsymbol{K}'$ from the demeaned samples:

$$\phi'(\boldsymbol{x}_i) = \phi(\boldsymbol{x}_i) - \frac{1}{N}\sum_l \phi(\boldsymbol{x}_l)$$

The (i,j) component of $\boldsymbol{K}'$ is

$$\begin{aligned} K'_{ij} &= \left[\phi(\boldsymbol{x}_i) - \tfrac{1}{N}\textstyle\sum_l \phi(\boldsymbol{x}_l)\right]\left[\phi(\boldsymbol{x}_j) - \tfrac{1}{N}\textstyle\sum_l \phi(\boldsymbol{x}_l)\right]^T \\ &= K(\boldsymbol{x}_i,\boldsymbol{x}_j) - \tfrac{1}{N}\textstyle\sum_l K(\boldsymbol{x}_i,\boldsymbol{x}_l) - \tfrac{1}{N}\textstyle\sum_l K(\boldsymbol{x}_l,\boldsymbol{x}_j) + \tfrac{1}{N^2}\textstyle\sum_{l,m} K(\boldsymbol{x}_l,\boldsymbol{x}_m) \end{aligned}$$

Finally,

$$\boldsymbol{K}' = \boldsymbol{K} - \mathbf{1}_N\boldsymbol{K} - \boldsymbol{K}\mathbf{1}_N + \mathbf{1}_N\boldsymbol{K}\mathbf{1}_N,$$

where $\mathbf{1}_N$ is the $N\times N$ matrix whose components are $1/N$.

- We obtain the top PCA directions from $\boldsymbol{K}'$.

Kernel trick

- Linear ML methods can be generalized to non-linear methods by simply substituting $\boldsymbol{x}_i \boldsymbol{x}_j^T$ with $K(\boldsymbol{x}_i, \boldsymbol{x}_j)$.
- This is called kernel trick or kernel method.
- Kernel method is memory-based or instance-based algorithm because the method need to sum the influences from all training samples.
- The SVM with non-linear kernel function and kernel PCA are the two important examples.