COMP 7070 Advanced Topics in AI and ML March 5, 2024

Lecture 8: Kernel tricks and RKHS

Instructor: Yifan Chen *Scribes: Jin Xiao, Ting Yang* *Proofreader: Zhanke Zhou*

Note: *LaTeX template courtesy of UC Berkeley EECS dept.*

8.1 Kernel trick

In machine learning, kernel machines are a class of algorithms for pattern analysis. Kernel methods owe their name to the use of kernel functions, which enable them to operate in a high-dimensional. implicit feature space without ever computing the coordinates of the data in that space but rather by simply computing the inner products between the images of all pairs of data in the feature space. This operation is often computationally cheaper than the explicit computation of the coordinates. This approach is called the "kernel trick":

$$\langle X, Z\rangle \Rightarrow K(x, z).$$

8.1.1 PSD Kernel

The most important kernel is the PSD kernel, which ensures that the optimization problems solved by learning algorithms have a global minimum, which is necessary for the convergence and stability of the algorithm.

The kernel matrix, or the Gram matrix, is a matrix where each element represents the kernel function applied to a pair of samples. Mathematically, the ij-th element is defined as $k(x_i, x_j)$ where k is the kernel function and x_i, x_j are data points from the dataset. This matrix encapsulates the pairwise similarities between all points in the dataset.

The Gram matrix is defined as:

$$G_{ij} = K(x_i, x_j) \quad \forall i, j \in \{1, 2, ..., n\}.$$

A matrix is positive semi-definite if $\forall z \in \mathbb{R}^n, z^T G z \geq 0$.

A kernel is considered PSD if its corresponding kernel matrix is positive semi-definite for any possible set of input data points.

Common examples of PSD kernels defined in Euclidean space $\mathbb{R}^d$ for example: Linear kernel, Polynomial kernel, Laplacian kernel.

8.1.1.1 Example: Linear Kernel

Definition: The linear kernel is defined as the dot product between two vectors:

$$K(x, y) = x^T y.$$

Proof of PSD property: For any finite set of points $\{x_1, x_2, ..., x_n\}$ and any $z \in \mathbb{R}^n$, For any vectors x_i and x_j, their inner product equals the dot product of the vectors, which is a bilinear form. This adheres to the definition of a PSD kernel because any vector's dot product with itself is always non-negative.

8.1.1.2 Example: Polynomial Kernel

Definition: For degree-d polynomials, the polynomial kernel is defined as

$$K(\mathbf{x}, \mathbf{y}) = (\mathbf{x}^T \mathbf{y} + r)^n, \quad \mathbf{x}, \mathbf{y} \in \mathbb{R}^d, r \geq 0, n \geq 1. \tag{8.1}$$

$$K(x,y) = (\langle x,y\rangle + c)^d$$
$$= \sum_{k=0}^{d} \binom{d}{k} \langle x,y\rangle^k c^{d-k}.$$

8.1.2 Mercer's theorem

Mercer's theorem is a fundamental result in functional analysis, particularly in the fields of kernel methods in machine learning and in the theory of integral equations. The theorem provides conditions under which a kernel function can be expressed as an infinite series expansion in terms of orthogonal basis functions.

$$\exists \text{ an orthonormal loss of } L^2(X) : \{f| \int_{\mathcal{X}} f^2(x)dx < \infty\}. \tag{8.2}$$

8.1.2.1 Property 1. Orthonormal Basis

An orthonormal basis in a Hilbert space (like $L^2(X)$, the space of square-integrable functions) is a set of basis vectors that are mutually orthogonal and of unit length. This means that for any two basis vectors ϕ_i and ϕ_j in the set:

1. Orthogonality: $\langle \phi_i, \phi_j \rangle = 0$ for $i \neq j$, where $\langle \cdot, \cdot \rangle$ denotes the inner product.
2. Normalization: $\langle \phi_i, \phi_i \rangle = 1$. That means for any function f in L^2 space, we can represent it by a linear combination of ϕ_i

$$f \in L^2(X) : f = \sum_{i=1}^{\infty} \beta_i \cdot \phi_i.$$

8.1.2.2 Property 2. Non-negative Eigen Values

In the context of Mercer's theorem, eigenvalues $\{u_i\}$ correspond to the weights associated with each orthogonal basis function ϕ_i in the expansion of the kernel function. These eigenvalues are non-negative due to the positive semi-definiteness of the kernel $K(x,y)$. The integration form is as follows:

$$\int_x \mathrm{K}\left(X,z\right) \cdot \phi_j\left(z\right) \mathrm{d}z = u_j \cdot \phi_j\left(x\right). \tag{8.3}$$

The inner product form is:

$$< \mathrm{K}\left(X,\right), \phi_j\left(\cdot\right) >= u_j \cdot \phi_j\left(X\right). \tag{8.4}$$

8.1.2.3 Property 3

Now we combine the above two properties, the kernel function K will be

$$K(x,y) = \sum_{i=1}^{\infty} \lambda_i \phi_i(x) \phi_i(y). \tag{8.5}$$

This is the essence of Mercer's Theorem, which means for any given kernel function, we can always find a sequence u_i and bases ϕ_j, so that we can expand the kernel function into the above form.

8.1.3 Reproducing Kernel Hilbert Space

8.1.3.1 Definition of RKHS function

A Reproducing Kernel Hilbert Space (RKHS) is a Hilbert space, which means it is a complete inner product space. It means you can always find a finite inner product for their Hilbert Space. We according have the norm for this space.

$$< \cdot, \cdot >_H \Rightarrow \|f\|_H^2 =< f, f >_H . \tag{8.6}$$

RKHS is actually a function. We assume the function f:

$$f \in \mathcal{H}.$$

This is very similar to Neural Network.

8.1.3.2 Guassion kernel and Matern Kernel with RKHS

The RKHS associated with the Gaussian kernel has some remarkable properties:

1. Universality: The RKHS of the Gaussian kernel is dense in the space of continuous functions on a compact set, meaning that any continuous function can be approximated arbitrarily well by functions in the RKHS.

2. Infinite Dimensionality: The RKHS of the Gaussian kernel is infinite-dimensional, allowing it to capture a wide range of functions.

8.1.3.3 Kernel reproducing property

The reproducing kernel K has two key properties:

1.Reproduction: For every $x \in X$ and $f \in \mathcal{H}$, we have

$$f(x) = \langle f, K(x, \cdot) \rangle_{\mathcal{H}}.$$

2.Positivity: For every finite set of points $\{x_1, x_2, \ldots, x_n\}$ in X and any set of coefficients $\{c_1, c_2, \ldots, c_n\}$ in $\mathbb{R}$, the kernel satisfies

$$\sum_{i=1}^{n} \sum_{j=1}^{n} c_i c_j K(x_i, x_j) \geq 0.$$

The kernel K is also symmetric, i.e.,

$$K(x, y) = K(y, x).$$

and the function $K(x, \cdot)$ is in $\mathcal{H}$ for every $x \in X$.

8.1.3.4 Construct the RKHS with Certain Property

$$\begin{aligned}
\tilde{\mathcal{H}} &= \{f \mid f(x) = \sum_{i=1}^{n} a_i, k(x_i, x), \forall \{x_i\}_{i=1}^{n} \subseteq \mathcal{X}\} \\
f &= \sum_{i}^{n} a_i, k(x_{i-i}), f = \sum_{i}^{n} a_i k(x_i, \cdot) \\
< f, F >_{\tilde{\mathcal{H}}} &= \sum_{j=1}^{n} \sum_{k=1}^{n} a_j \cdot \bar{a_k} \cdot K(x_j, \bar{x_k}) \\
< f, k(z, \cdot) >_{\tilde{\mathcal{H}}} &= \sum_{j=1}^{n} a_j \cdot K(x_j, z) = f(z), \forall f \in \tilde{\mathcal{H}}.
\end{aligned} \tag{8.7}$$

8.1.4 The Representer Theorem

In statistical learning theory, a representer theorem is any of several related results stating that a minimizer f^* of a regularized empirical risk functional defined over a reproducing kernel Hilbert space can be represented as a finite linear combination of kernel products evaluated on the input points in the training set data.

$$f^* = \underset{f \in \mathcal{H}}{\operatorname{argmin}} C\left(x_i, y_i, f(x_i)_{i=1}^{n}\right) + \Omega\left(||f||_{\mathcal{H}}\right). \tag{8.8}$$

where Ω is increasing on R^+, for example, $\frac{2}{\lambda}||f||$. Then, a representation of the form is as follows:

$$f^* = \sum_{i=1}^{n} \alpha_i K(x_i, \cdot), \text{where } \alpha_i \in \mathbb{R} \text{ for all } 1 \leq i \leq n. \tag{8.9}$$

which can reduce the dimension.

High-level Proof. $\forall f \in \mathcal{H}, \mathcal{H} \subseteq L_{\mathcal{X}}$, it can be represented by $f = \sum_{i=1}^{\infty} \beta_i \phi_i$. Decompose f as

$$f = f^* + f^{\perp}, \text{s.t. } f^* = \sum_{i=1}^{n} \alpha_i K(x_i, \cdot), \left\langle f^{\perp}, k(x_i, \cdot) \right\rangle_{\mathcal{H}} = 0. \tag{8.10}$$

Therefore, for $\forall i \in [n]$, we can get

$$f(x_i) = \left\langle K(x_i, \cdot), f^* + f^{\perp} \right\rangle_{\mathcal{H}} \tag{8.11}$$

$$= \langle K(x_i, \cdot), f^* \rangle_{\mathcal{H}} \tag{8.12}$$

$$= f^*(x_i). \tag{8.13}$$

Then

$$C(x_i, y_i, f(x_i)) = C(x_i, y_i, f^*(x_i)). \tag{8.14}$$

$$\Omega(||f||_{\mathcal{H}}) = \Omega(\sqrt{||f^*||_{\mathcal{H}}^2 + ||f^{\perp}||_{\mathcal{H}}^2}) >= \Omega(||f^*||_{\mathcal{H}}). \tag{8.15}$$

Replacing f by f^* would give a smaller object value, which means the minizer must be in the form of f^*.

8.2 Apply kernel trick

8.2.1 Rethink the kernel trick

Before we apply it, let's rethink the kernel trick we mentioned.

$$\langle X, Z \rangle \Rightarrow K(X, Z). \tag{8.16}$$

$$\varphi \colon \mathbb{R}^d \to \mathcal{H} \Leftrightarrow \varphi(x) = \sum_{i=1}^{\infty} \sqrt{\mu_i \cdot \phi_i(x) \cdot \phi_i(\cdot)} \tag{8.17}$$

$$\Rightarrow \varphi(x) \text{ correspond to a feature map} \tag{8.18}$$

$$\Rightarrow \begin{bmatrix} \sqrt{\mu_1 \cdot \phi(x)} \\ \sqrt{\mu_2 \cdot \phi(x)} \\ \vdots \end{bmatrix}_{\infty \times 1}. \tag{8.19}$$

$\Phi(X)$ is $n \times \infty$ matrix. We can have a more comprehensive understanding of kernel tricks.

$$\hat{Y} = X\beta \Rightarrow \Phi(X) \cdot \tilde{\beta}. \tag{8.20}$$

Due to the representor theorem: $\tilde{\beta} = \Phi^T(X)_n \cdot \alpha_{n \times 1}$. We can obtain

$$X\beta \Rightarrow \cdot \Phi(X) \cdot \Phi^T(X)\alpha = K_{n \times n} \cdot \alpha_{n \times 1}. \tag{8.21}$$

8.2.2 Example: Kernel Ridge Regression(KRR)

$$min_{f \in \mathcal{H}} \frac{1}{n} \cdot [Y - f(x)]^T \cdot [Y - f(x)] + \lambda ||f||_{\mathcal{H}}^2. \tag{8.22}$$

by using representor theorem $f = \sum_{i=1}^{n} \alpha_i K(x_i, \cdot)$, we can get

$$f(x_j) = \langle f, K(x_j, \cdot) \rangle_{\mathcal{H}} = \sum_{i=1}^{n} \alpha_i \cdot K(x_i, x_j) \tag{8.23}$$

$$\Rightarrow f(X) = K\alpha \tag{8.24}$$

$$\Rightarrow ||f||_{\mathcal{H}}^2 = \langle f, f \rangle_{\mathcal{H}} = \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j K(x_i, x_j) = \alpha^T K \alpha. \tag{8.25}$$

Then

$$(8.22) \Leftrightarrow min_\alpha \frac{1}{n}(Y - K\alpha)^T(Y - K\alpha) + \lambda \cdot \alpha^T K\alpha \tag{8.26}$$

$$\Rightarrow \text{(derivative) } \frac{2}{n} \cdot (-K)(Y - K\alpha) + 2\lambda K\alpha = 0 \tag{8.27}$$

$$\Rightarrow (K + n\lambda I)\alpha = Y \tag{8.28}$$

$$\Rightarrow \alpha^* = (K + n\lambda I)^{-1}Y. \tag{8.29}$$

Finally, we can get

$$\hat{f}(X) = K \cdot \alpha^* = K(K + n\lambda I)^{-1}Y \tag{8.30}$$

$$\hat{f}(Z) = \sum_{i=1}^{n} \alpha_i^* \cdot K(x_i, Z). \tag{8.31}$$

8.3 Neural Tangent Kernel

1. For a kernel estimator: $f(\theta(t), x_i) = \langle \theta(x_i, \theta(t)) \rangle_{l^2(N)}$, we can compute:

$$\frac{\partial l(\theta(t))}{\partial \theta(t)} = \sum_{i=1}^{n} \frac{\partial l}{\partial f(\theta, x_i)} \cdot \Phi(x_i). \tag{8.32}$$

For an NN:

$$\frac{\partial l(\theta(t))}{\partial \theta(t)} = \sum_{i=1}^{n} \frac{\partial l}{\partial f(\theta, x_i)} \cdot \frac{\partial f(\theta(t), x_i)}{\partial \theta}, \tag{8.33}$$

$$NTK(x_i, x_j) = \langle \Phi(x_i), \Phi(x_j) \rangle) = \left\langle \frac{\partial f(\theta(t), x_i)}{\partial \theta}, \frac{\partial f(\theta(t), x_j)}{\partial \theta} \right\rangle. \tag{8.34}$$

remark:

- NTK, the name comes from "tangent model", which is a kernel that describes the evolution of deep artificial neural networks during their training by gradient descent.

$$f(\theta(t), x_i) \approx f(\theta(0), x_i) + \Phi^T(x_i)(\theta(t) - \theta(0)). \tag{8.35}$$

- In kernel method, $\Phi(x_i)$ would not change with time. For NTK, $\left\langle \frac{\partial f(\theta(t), x_i)}{\partial \theta(t)}, \frac{\partial f(\theta(t), x_j)}{\partial \theta(t)} \right\rangle$ would roughly remain in dynamics.

- $|\theta|$ should be large. $\Phi(x)_{\infty \times 1}$. For infinitely-wide neural networks, the Neural Tangent Kernel (NTK) can be effectively applied. For sufficiently wide neural networks, the NTK remains a useful tool. However, in practical scenarios such as pretraining models, the applicability of the NTK may not be guaranteed.

8.4 Attention as a kernel estimator

Recall the definition of attention, then we can get:

$$f(q_i) = \frac{\exp(\langle q_i, k_j \rangle) \cdot v_i}{\sum_{j=1}^{n} \exp(\langle q_i, k_j \rangle)} \tag{8.36}$$

$$= \frac{K(q_i, k_j) \cdot v_j}{\sum_{j=1}^{n} K(q_i, k_j)} \Leftrightarrow \text{NW estimator in kernel methods.} \tag{8.37}$$

$$f(Q) = D^{-1}\tilde{K}(Q, K) \cdot V \text{ where } D = diag(\tilde{K}(Q, K) \cdot I_n). \tag{8.38}$$