Lecture 02: Reproducing Kernel Hilbert Spaces

Geoff Pleiss

Reproducing kernel Hilbert spaces (RKHS) are function spaces that play an important role in the analysis of neural networks and other machine learning models. These spaces contain "complex" non-linear functions, yet the spaces are surprisingly structured in a way that's amenable to theoretical analysis.

Most texts introduce RKHS from a functional analysis perspective. Here we will provide a simpler introduction, starting with spaces of finite-dimensional linear functions and gaining complexity. The goal is to elevate concepts from standard matrix-based linear algebra into abstract infinite-dimensional spaces of functions.

These notes are a brief introduction to RKHS, foregoing many important properties and theorems. See [Wainwright, 2019, Ch. 12] for a thorough reference.

---

1) Linear Functions and Inner Products

Consider the space of $\mathbb{R} \to \mathbb{R}$ functions

$$\mathcal{H} = \left\{ f(x) = \sum_{j=1}^{d} \left[\theta_{2j-1} \cos(jx) + \theta_{2j} \sin(jx) \right] \;:\; \theta_1, \ldots, \theta_{2d} \in \mathbb{R} \right\}$$

for some $d \in \mathbb{N}$. The space considers all linear functions that can be built off of a $2d$-dimensional Fourier basis expansion of x. Note that any function $f(\cdot) \in \mathcal{H}$ can be written as:

$$f(x) = \Bigg\langle \underbrace{\begin{bmatrix} \theta_1 \\ \vdots \\ \theta_{2d} \end{bmatrix}}_{\boldsymbol{\theta}}, \underbrace{\begin{bmatrix} \cos(x) \\ \vdots \\ \sin_{dx}(x) \end{bmatrix}}_{\boldsymbol{z}(x)} \Bigg\rangle, \tag{1}$$

where $\boldsymbol{\theta} \in \mathbb{R}^{2d}$ are the function parameters and $\boldsymbol{z} : \mathbb{R} \to \mathbb{R}^{2d}$ is the Fourier basis expansion function. We refer to $\boldsymbol{z}(x) \in \mathbb{R}^{2d}$ as a **feature representation** of x. Assuming the basis expansion is fixed, any $f(\cdot) \in \mathbb{R}^{2d}$ is entirely specified by $\boldsymbol{\theta}$, and so we can implicitly define $f(\cdot)$ through $\boldsymbol{\theta}$. We thus refer to $\boldsymbol{\theta}$ as the **function representation** of $f(\cdot)$.

Evaluating $f(\cdot)$ on any input x requires computing an inner product between two vectors: $\boldsymbol{\theta}$ and $\boldsymbol{z}(x)$. While this fact may seem straightforward, it unearths a lot of interesting complexities:

- The inner product we use to evaluate $f(x)$ can also be used to compare two functions. I.e., given $f(x) = \langle \boldsymbol{\theta}, \boldsymbol{z}(x) \rangle$ and $\tilde{f}(x) = \langle \tilde{\boldsymbol{\theta}}, \boldsymbol{z}(x) \rangle$, we can compute $\langle \boldsymbol{\theta}, \tilde{\boldsymbol{\theta}} \rangle$.
  - We can also use the same inner product to define a norm $\|\boldsymbol{\theta}\| = \langle \boldsymbol{\theta}, \boldsymbol{\theta} \rangle^{1/2}$.
- For any $x' \in \mathbb{R}$, the vector $\boldsymbol{z}(x')$ is also a $\mathbb{R}^{2d}$ vector and thus parameterizes a function in $\mathcal{H}$. (I.e. there exists some $k_{x'}(x) = \langle \boldsymbol{z}(x'), \boldsymbol{z}(x) \rangle$.)
  - In other words, for every x, we have a *function representation* $k_x(\cdot)$ in addition to its *feature representation* $\boldsymbol{z}(x)$!

---

2) From Inner Products on Vectors to Inner Products on Functions

Because there is a one-to-one mapping between vectors $\boldsymbol{\theta}, \tilde{\boldsymbol{\theta}}, \boldsymbol{z}(x') \in \mathbb{R}^{2d}$ to functions $f(\cdot), \tilde{f}(\cdot), k_{x'}(\cdot) \in \mathcal{H}$, we can define an *inner product on* $\mathcal{H}$ using our inner product over $\mathbb{R}^{2d}$:

$$\left\langle f(\cdot), \tilde{f}(\cdot) \right\rangle_{\mathcal{H}} := \left\langle \boldsymbol{\theta}, \tilde{\boldsymbol{\theta}} \right\rangle \qquad (f(\cdot) = \langle \boldsymbol{\theta}, \boldsymbol{z}(\cdot) \rangle, \quad \tilde{f}(\cdot) = \langle \tilde{\boldsymbol{\theta}}, \boldsymbol{z}(\cdot) \rangle)$$

Curiously, since $k_{x'}(\cdot) = \langle \boldsymbol{z}(x'), \boldsymbol{z}(\cdot) \rangle \in \mathcal{H}$, our inner product over $\mathcal{H}$ can be used to *evaluate* $\mathcal{H}$ functions!

$$f(x') = \langle f(\cdot), k_{x'}(\cdot) \rangle_{\mathcal{H}} = \langle \boldsymbol{\theta}, \boldsymbol{z}(\cdot) \rangle \qquad (f(\cdot) = \langle \boldsymbol{\theta}, \boldsymbol{z}(\cdot) \rangle, \quad k_{x'}(\cdot) = \langle \boldsymbol{z}(x'), \boldsymbol{z}(\cdot) \rangle)$$

We thus refer to $k_{x'}(\cdot)$ as the **evaluation function** for x'.

---

3) Dual (Data-Based) Representations and Kernel Functions

Given a set of $x_1, \ldots, x_{2d}$ so that $\boldsymbol{z}(x_1), \ldots, \boldsymbol{z}(x_{2d})$ spans $\mathbb{R}^{2d}$, any $\theta \in \mathbb{R}^{2d}$ can be defined as $\sum_{j=1}^{2d} \alpha_j \boldsymbol{z}(x_j)$ for some $\alpha_1, \ldots, \alpha_{2d}$, and thus any $f(\cdot) \in \mathcal{H}$ can be written as

$$f(\cdot) = \left\langle \left(\sum_{j=1}^{2d} \alpha_j \boldsymbol{z}(x_j) \right), \boldsymbol{z}(\cdot) \right\rangle = \sum_{j=1}^{2d} \alpha_j \left\langle \boldsymbol{z}(x_j), \boldsymbol{z}(\cdot) \right\rangle = \sum_{j=1}^{2d} \alpha_j \underbrace{\left\langle k_{x_j}(\cdot), k_x(\cdot) \right\rangle}_{:= k(x_j, x)}.$$

In other words, any function $f \in \mathcal{H}$ admits a **dual (data-based) representation** through the **kernel function** $k(\cdot, \cdot)$:

$$\mathcal{H} = \left\{ f(\cdot) = \sum_{j=1}^{2d} \alpha_j k(x_j, \cdot), : \ \alpha_j \in \mathbb{R}, \ x_j \in \mathbb{R} \right\}. \tag{2}$$

There is a deep connection between this dual representation and standard training of machine learning algorithms:

Theorem 1 (Representer Theorem [Kimeldorf and Wahba, 1970, Schölkopf et al., 2001])**.** *Given training data* $(x_1, y_1), \ldots, (x_n, y_n)$*, some loss function* $\ell(f(x), y)$*, and some regularization parameter* $\lambda > 0$*, the solution to the regularized training objective can be written as*

$$f^*(x) = \sum_{j=1}^{n} \alpha_j k(x_j, x)$$

for some $\alpha_1, \ldots, \alpha_n$*.*

---

4) Spectrum of the Kernel Function

The kernel function $k(x, x') = \langle \boldsymbol{z}(x), \boldsymbol{z}(x') \rangle$ has some curious properties.

- For any $x_1, \ldots, x_n$, the matrix

  $$\begin{bmatrix} k(x_1, x_1) & \ldots & k(x_1, x_n) \\ \vdots & \ddots & \vdots \\ k(x_n, x_1) & \ldots & k(x_n, x_n) \end{bmatrix} = \begin{bmatrix} \boldsymbol{z}(x_1)^\top \\ \vdots \\ \boldsymbol{z}(x_n)^\top \end{bmatrix} \begin{bmatrix} \boldsymbol{z}(x_1) & \ldots & \boldsymbol{z}(x_n) \end{bmatrix}$$

  is positive definite.

- $k(x, x')$ can be defined through the eigenvalues of the matrix $\mathbb{E}[\boldsymbol{z}(x)\boldsymbol{z}(x)^\top] \in \mathbb{R}^{2d \times 2d}$. Letting $\boldsymbol{V}\boldsymbol{\Lambda}\boldsymbol{V}^\top$ be an eigendecomposition of $\boldsymbol{\Sigma} := \mathbb{E}[\boldsymbol{z}(x)\boldsymbol{z}(x)^\top]$, define

$$\{\phi_j(\cdot) = \lambda_j^{-1/2} \langle \boldsymbol{v}_j, \boldsymbol{z}(\cdot) \rangle\}_{j=1}^{2d}$$

as **eigenfunctions** of $\mathcal{H}$ (where $\boldsymbol{v}_j$ and λ_j are the columns of $\boldsymbol{V}$ and diagonals of $\boldsymbol{\Lambda}$ respectively). Then:

$$\begin{aligned} k(x, x') = \langle \boldsymbol{z}(x), \boldsymbol{z}(x') \rangle &= \left(\boldsymbol{z}(x)^\top \boldsymbol{V} \boldsymbol{\Sigma}^{-1/2} \right) \boldsymbol{\Sigma} \left(\boldsymbol{\Sigma}^{-1/2} \boldsymbol{V} \boldsymbol{z}(x') \right) \\ &= \sum_{j=1}^{2d} \lambda_j \left(\lambda_j^{-1/2} \boldsymbol{v}_j^\top \boldsymbol{z}(x) \right) \left(\lambda_j^{-1/2} \boldsymbol{v}_j^\top \boldsymbol{z}(x) \right) \\ &= \sum_{j=1}^{2d} \lambda_j \phi_j(x) \phi_j(x'). \end{aligned} \tag{3}$$

Moreover, we can easily verify that:

$$\begin{aligned} \mathbb{E}[\phi_i(x)\phi_j(x)] &= \mathbb{E}[\left(\lambda_i^{-1/2} \lambda_j^{-1/2} \right) \boldsymbol{v}_i^\top \boldsymbol{z}(\boldsymbol{x}) \boldsymbol{z}(\boldsymbol{x})^\top \boldsymbol{v}_j] \\ &= \left(\lambda_i^{-1/2} \lambda_j^{-1/2} \right) \boldsymbol{v}_i^\top \boldsymbol{\Sigma} \boldsymbol{v}_j \\ &= \delta_{ij} = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases} \end{aligned}$$

This spectral representation of $k(\cdot, \cdot)$ contains lots of information about $\mathcal{H}$. Since $\boldsymbol{\Sigma}$ is positive definite, we know that $\lambda_1, \ldots, \lambda_d > 0$. If the eigenvalues decay quickly, then $\boldsymbol{\Sigma}$ is low-rank implying that many of the features in $\boldsymbol{z}(\cdot)$ are co-linear/redundant. This implies that $\mathcal{H}$ may be an "intrinsically low-dimensional" space (approximately few degrees of freedom) even though there are actually $2d$ parameters to fit.

---

5) Reproducing Kernel Hilbert Spaces

Why did we go through the trouble of defining:

- an inner product over functions,
- a data-based representation of functions, and
- an eigendecomposition of a function?

It turns out this is the right abstraction to define powerful spaces of functions with remarkably easy-to-analyze properties. Everything we just defined holds even if we change the feature representation or even if we take $d \to \infty$.

Definition 1 (Reproducing Kernel Hilbert Spaces (RKHS))**.** *Given a positive definite kernel function $k(\cdot, \cdot) : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$; i.e. a function that can be written as:*

$$k(x, x') = \sum_{i=1}^{\infty} \lambda_i \phi_i(x) \phi_i(x'), \qquad \lambda_1 \geq \lambda_2 \geq \ldots \geq 0, \quad \mathbb{E}[\phi_i(x)\phi_i(x')] = \delta_{ij},$$

a reproducing kernel Hilbert space $\mathcal{H}$ is the space[1] *of $\mathcal{X} \to \mathbb{R}$ functions that can be written as*

$$f(\cdot) = \sum_{i=1}^{n} \alpha_i k(\boldsymbol{x}_i, \cdot), \qquad n \in \mathbb{N}, \{\boldsymbol{x}_i\}_{i=1}^{n} \in \mathcal{X}. \tag{4}$$

The inner product associated with $\mathcal{H}$ is given by

$$\left\langle f(\cdot), \tilde{f}(\cdot) \right\rangle = \sum_{i=1}^{n} \sum_{j=1}^{\tilde{n}} \alpha_i \tilde{\alpha}_j k(\boldsymbol{x}_j, \tilde{\boldsymbol{x}}_j).$$

Note that we could have alternatively defined the RKHS using the *infinite-dimensional feature expansion* implied by the eigendecomposition of $k(\cdot, \cdot)$:

$$\mathcal{H} = \left\{ f(\cdot) = \sum_{j=1}^{\infty} \theta_j \left(\lambda_j^{1/2} \phi(\cdot) \right), \quad \theta_1, \theta_2, \ldots \in \mathbb{R} \right\}.$$

(As an exercise, you should show that these two definitions yield the same space of functions.) However, rather than dealing with infinite-dimensional vectors, we can instead deal with scalar kernel functions $k(\boldsymbol{x}, \boldsymbol{x}')$. This abstraction will yield simple closed-form expressions of neural network as well as straightforward analyses of their generalization properties.

The only portion of the feature expansion we will consider are the eigenvalues λ_1, λ_2, ... associated with $k(\boldsymbol{x}, \boldsymbol{x}')$. As discussed above, the rate of decay of this spectrum tells us about the relative complexity of $\mathcal{H}$, which will be necessary for the analysis of generalization.

References

G. S. Kimeldorf and G. Wahba. A correspondence between Bayesian estimation on stochastic processes and smoothing by splines. *The Annals of Mathematical Statistics*, 41(2):495–502, 1970.

B. Schölkopf, R. Herbrich, and A. J. Smola. A generalized representer theorem. In *International Conference on Computational Learning Theory*, pages 416–426, 2001.

M. J. Wainwright. *High-dimensional statistics: A non-asymptotic viewpoint*, volume 48. Cambridge University Press, 2019.

[1] Technically, $\mathcal{H}$ is the *completion* of the space defined by Eq. (4).