Lecture 15: Introduction to Kernel Methods

So far in the course we have seen a number of different algorithms & methods for analysis of data. A recurring theme has been representations of the form

$$f(\underline{x}) = \sum_j c_j \psi_j(\underline{x}), \quad c_j \in \mathbb{R}, \mathbb{C}$$

- Fourier series $f(t) = \sum_j c_j \exp\left(\frac{i\pi k t}{L}\right)$
- Wavelets $f(t) = \sum_j \sum_k c_{jk} \psi_{jk}(t)$
- PCA $\underline{f} = \sum_j \lambda_j \underline{\psi}_j$
- Ridge Regression $\hat{f}(\underline{x}) = \sum_j \hat{\beta}_j \psi_j(\underline{x})$,

$$\hat{\beta} = \underset{\beta}{\operatorname{argmin}} L(X\beta, Y) + \lambda \|\beta\|.$$

In this section of the course we will attempt to unite all of these approaches within the framework of Kernel methods.

This is an advanced topic so it needs some effort to understand but the resulting knowledge is useful & the implementation is straightforward.

15.1 Some preliminaries & definitions

To set the ideas in place we will mostly focus on the SL setting

$$\text{dataset} \quad \begin{pmatrix} X = \{\underline{x}_0, \dots, \underline{x}_{N-1}\} \\ Y = \{y_0, \dots, y_{N-1}\} \end{pmatrix}$$

where $\underline{x}_j \in \mathbb{R}^d$, $y_j \in \mathbb{R}$. Our goal is to approx. a mapping f so that $\hat{f}(X) \approx Y$.

In what follows we will present a method for this task that generalizes what we have seen so far in the context of ridge regression.

Def^n: A function $K : \mathbb{R}^n \times \mathbb{R}^n \longrightarrow \mathbb{R}$ is called a Kernel.

We say K is non-negative definite & symmetric (NDS) if

- $K(\underline{x}, \underline{x}') = K(\underline{x}', \underline{x}), \quad \forall \underline{x}, \underline{x}' \in \mathbb{R}^n$
- For any set of points $(\underline{x}_0, \dots, \underline{x}_M) \in \mathbb{R}^n$ the matrix $(K)_{ij} := K(\underline{x}_i, \underline{x}_j) \in \mathbb{R}^{M \times M}$ is NDS.

Thm (Mercer) If K is NDS and

$$\int_{\mathbb{R}^n} K(\underline{x}, \underline{x}) \, d\underline{x} < +\infty$$

then there exists an orthonormal basis $\{\psi_j\}_{j=0}^{\infty} \in L^2(\mathbb{R}^n)$ so that

$$K(\underline{x}, \underline{x}') = \sum_{j=0}^{\infty} \lambda_j \psi_j(\underline{x}) \psi_j(\underline{x}').$$

the numbers $\lambda_j \geq 0$ are the eigenvalues while the ψ_j are eigenfunction.

This is a generalization of the eigendecomposition of NDS matrices:

If $A \in \mathbb{R}^{M \times M}$ is NDS then $A = Q \Lambda Q^{-1}$

$$\Rightarrow A = \sum_j \lambda_j \underline{q}_j \underline{q}_j^T.$$

Define the functions

$$F_j(\underline{x}) := \sqrt{\lambda_j} \, \psi_j(\underline{x}), \quad F_j : \mathbb{R}^n \longrightarrow \mathbb{R}$$

along with the mapping

$$F(\underline{x}) := \left(F_0(\underline{x}), F_1(\underline{x}), F_2(\underline{x}), \dots \right) \in \ell^2$$

$$\ell^2 := \left\{ \{c_j\}_{j=0}^{\infty} \;\middle|\; c_j \in \mathbb{R}, \; \sum_{j=0}^{\infty} c_j^2 < +\infty \right\}.$$

Then F is called a **feature map of the kernel K.**

Observe that by Mercer's thm we simply have

$$K(\underline{x}, \underline{x}') = \left(F(\underline{x}), F(\underline{x}')\right)_{\ell^2} = \sum_{j=0}^{\infty} F_j(\underline{x}) F_j(\underline{x}')$$

when we define the ℓ^2 inner product

$$\left(\{a_j\}_{j=0}^{\infty}, \{b_j\}_{j=0}^{\infty}\right)_{\ell^2} = \sum_{j=0}^{\infty} a_j b_j$$

Side note: I am using a restricted definition of a feature map. In general the feature maps are not unique & any function $\Phi: \mathbb{R}^n \to H$ that satisfies

$$K(\underline{x}, \underline{x}') := \langle \Phi(\underline{x}), \Phi(\underline{x}') \rangle_H$$

is called a feature map for K. Here H is any inner-product space with inner-prod $\langle \cdot, \cdot \rangle_H$.

** We won't need this broader def^n in the course.

Def^n Given an NDS kernel K we define its reproducing Kernel Hilbert space (RKHS) as the space of functions $f: \mathbb{R}^n \to \mathbb{R}$

that have the form

$$f(\underline{x}) = \sum_{j=0}^{\infty} c_j F_j(\underline{x})$$

that satisfy $\sum_{j=0}^{\infty} c_j^2 < +\infty$.

We write $\mathcal{H}_K$ to denote this RKHS & equip it with the norm $\|f\|_{\mathcal{H}_K} := \left(\sum_{j=0}^{\infty} c_j^2\right)^{1/2}$. Hence,

$$\langle f, f' \rangle_{H_K} := \left(\{c_j\}_{j=0}^{\infty}, \{c_j'\}_{j=0}^{\infty}\right)_{\ell^2}, \quad f = \sum_j c_j F_j, \quad f' = \sum_j c_j' F_j$$

$$\mathcal{H}_K = \left\{ f : \mathbb{R}^n \to \mathbb{R} \;\middle|\; f = \sum_j c_j F_j, \; \sum_j c_j^2 < +\infty \right\}.$$

The intuition for the use of RKHS's is as follows:

- Essentially, the features F_j dictate what the functions inside $\mathcal{H}_K$ look like. So in practice we can build our kernel by prescribing its features.

  Think about trigonometric functions $F_j(\underline{x}) = \sin(j\underline{x})$ or wavelets $F_j(\underline{x}) = \psi_{a_j, b_j}(\underline{x})$, etc.

- alternatively, we can define the kernel directly & not worry about the features at all, ex.

  Gaussian kernel ("squared exponential") $K(\underline{x}, \underline{x}') = \exp\left(-\frac{\|\underline{x} - \underline{x}'\|^2}{2\ell^2}\right)$

• Mercer's thm tells us that defining K directly is equivalent to choosing its features F_j & in both cases the RKHS exists.

functions in $\mathcal{H}_K$ have a lot of nice properties, for ex

we have $K(\underline{x}, \underline{x}') = \sum_j F_j(\underline{x}) F_j(\underline{x}')$ & so,

for a fixed $\underline{x}^* \in \mathbb{R}^n$ & $f \in \mathcal{H}_K$ we have that

$$f(\underline{x}^*) = \sum_{j=0}^{\infty} c_j F_j(\underline{x}^*) = \left(\{c_j\}_{j=0}^{\infty}, \{F_j(\underline{x}^*)\}_{j=0}^{\infty} \right)_{\ell^2}$$

$$= \langle f, K(\underline{x}^*, \cdot) \rangle_{\mathcal{H}_K}$$

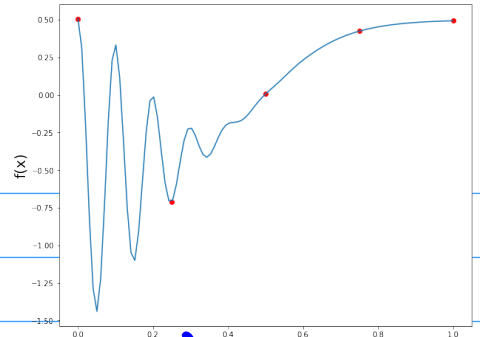
This is called the reproducing property of $\mathcal{H}_K$.

15.1 Kernel Interpolation

Our primary application domain for kernels is to use them for penalization/regularization in regression tasks.

As a preliminary step we first consider

The interpolation problem.



Given $\{x_0, \dots, x_{N-1}\} \in \mathbb{R}$ & $\{y_0, \dots, y_{N-1}\} \in \mathbb{R}$ find $f \in \mathcal{H}_K$ s.t $f(x_j) = y_j$ that has minimal $\mathcal{H}_K$ norm.

Observe, since $f \in \mathcal{H}_K$ then $f = \sum_{j=0}^{\infty} c_j F_j$ & so we need to solve

$$\sum_{j=0}^{\infty} c_j F_j(x_i) = y_i \,, \quad i = 0, \dots, N-1.$$

But this is an underdetermined prob. & has infinitely many solutions. Hence, the reason why we look for the solution with minimal $\mathcal{H}_K$ norm.

Therefore we wish to solve

$$\text{minimize} \sum_{j=0}^{\infty} c_j^2$$

$$\text{s.t.} \sum_{j=0}^{\infty} c_j F_j(x_i) = y_i$$

But this formulation is a bit problematic since there may be infinitely many features F_j. Alternatively, we consider the kernel formulation of the problem

> every $f \in \mathcal{H}_K$ can be written as $f(\underline{x}) = \sum_{j=0}^{\infty} a_j K(\tilde{\underline{x}}_j, \underline{x})$ for coeffs $a_j \in \mathbb{R}$ & points $\tilde{\underline{x}}_j \in \mathbb{R}^d$.

Then our interpolation problem takes the form

$a_j \sum_k F_k(\tilde{x}_j) F_k(x)$

$$\text{minimize} \quad \left\| \sum_j a_j K(\tilde{x}_j, x) \right\|_{\mathcal{H}_K}^2$$

$$\text{s.t.} \quad \sum_{j=0}^{\infty} a_j K(\tilde{x}_j, x_i) = y_i$$

Then, there exists a thm that tells us that the minimizer of this problem is precisely of the form

$$f(x) = \sum_{j=0}^{N-1} a_j K(x_j, x)$$

The interpolation constraints then tell us that

$$\sum_{j=0}^{N-1} a_j K(x_j, x_i) = y_i$$

$$\Rightarrow \quad \Theta a = Y,$$

$a = (a_0, \dots, a_{N-1})$

$Y = (y_0, \dots, y_{N-1})$

$\Theta_{ji} = K(x_j, x_i)$

Thus, $a = \Theta^{-1} Y$ $\quad \Theta \in \mathbb{R}^{N-1 \times N-1}$ is invertible if K is strictly pos. def. & the x_j are distinct.

- Matrix Θ is called the Kernel matrix.
- We can directly compute

$$\|f\|^2_{\mathcal{H}_k} = \left\| \sum_{j=0}^{N-1} a_j K(\underline{x}_j, \cdot) \right\|^2_{\mathcal{H}_k}$$

$$= \left\langle \sum_{j=0}^{N-1} a_j K(\underline{x}_j, \cdot), \sum_{n=0}^{N-1} a_n K(\underline{x}_n, \cdot) \right\rangle_{\mathcal{H}_k}$$

reproducer's property

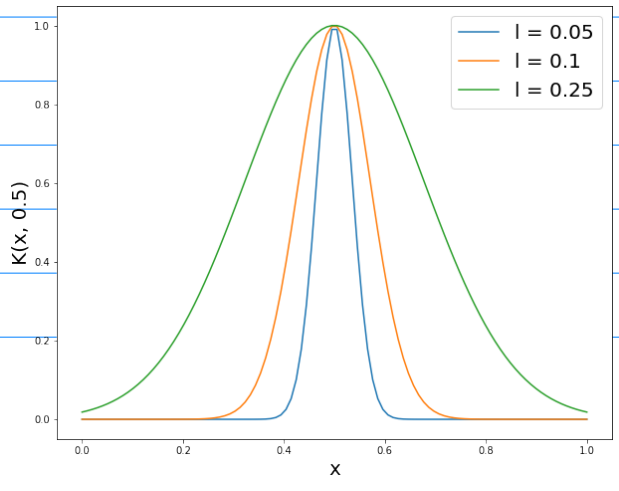
$$= \sum_{j=0}^{N-1} \sum_{n=0}^{N-1} a_j a_n \langle K(\underline{x}_j, \cdot), K(\underline{x}_n, \cdot) \rangle_{\mathcal{H}_k}$$

$$= \sum_{j,n=0}^{N-1} a_j a_n K(\underline{x}_j, \underline{x}_n) = \underline{a}^T \Theta \underline{a}$$

$$= \underline{Y}^T \Theta^{-1} \underline{Y}$$

For our demo we will pick the Gaussian kernel

$$K(x, x') = \exp\left(\frac{-|x - x'|^2}{l^2}\right)$$



The choice of this kernel has a profound impact on the quality of the interpolant.