Chapter 4, Part 4: Constructing Kernels

Advanced Topics in Statistical Machine Learning

Tom Rainforth

Hilary 2024

rainforth@stats.ox.ac.uk

Constructing Kernels

There are three equivalent ways of constructing a kernel:

- Defining a feature map $\varphi(x)$ and then taking the inner product: $k(x, x') = \langle \varphi(x), \varphi(x') \rangle_{\mathcal{H}}$
- As a positive definite function: $\sum_i \sum_j a_i a_j k(x_i, x_j) \geq 0$
- By choosing an RKHS $\mathcal{H}_k$ and then considering the unique reproducing kernel associated with $\mathcal{H}_k$

Today we will use these ideas to introduce some basic rules of constructing kernels and some example kernels, along with some demonstrations of how functions in their corresponding RKHSs behave

Mappings Between Spaces

Lemma 1 (Mappings between spaces)

Given a map $A : \mathcal{X} \to \widetilde{\mathcal{X}}$ *and kernel* k *on* $\widetilde{\mathcal{X}}$*, then* $k(A(x), A(x'))$ *is a kernel on* $\mathcal{X}$*.*

Proof.

If k is a kernel then $k(A(x), A(x')) = \langle \varphi(A(x)), \varphi(A(x')) \rangle_{\mathcal{H}}$ which is a kernel with features $\varphi(A(x))$. $\square$

This result is important when we want to define kernels on inputs that do not live in the reals (i.e. $\mathcal{X} \not\subseteq \mathbb{R}^p$): we can project our inputs into the space of reals and then apply a standard kernel.

Sum Rule of Kernels

Lemma 2 (Sums of kernels are kernels)

Given kernels k_1 and k_2 on $\mathcal{X}$ and positive constants $\alpha_1, \alpha_2 > 0$, then $k = \alpha_1 k_1 + \alpha_2 k_2$ is also a kernel on $\mathcal{X}$.

Proof.

If k_1 and k_2 are positive definite, this implies

$$\begin{aligned} &\sum_i \sum_j a_i a_j k(x_i, x_j) \\ &\qquad = \alpha_1 \sum_i \sum_j a_i a_j k_1(x_i, x_j) + \alpha_2 \sum_i \sum_j a_i a_j k_2(x_i, x_j) \\ &\qquad \geq 0 \quad \forall x_i \in \mathcal{X}, \forall a_i \in \mathbb{R} \end{aligned}$$

and so k is also positive definite. $\square$

Note: $k_1 - k_2$ need not be a kernel

Product Rule of Kernels

Lemma 3 (Products of kernels are kernels)

Given k_1 on $\mathcal{X}$ and k_2 on $\mathcal{Y}$, then

$$k\left(\left(x,y\right),\left(x',y'\right)\right) = k_1\left(x,x'\right)k_2\left(y,y'\right)$$

is a kernel on $\mathcal{X} \times \mathcal{Y}$. Moreover, if $\mathcal{X} = \mathcal{Y}$, then

$$k\left(x,x'\right) = k_1\left(x,x'\right)k_2\left(x,x'\right)$$

is a kernel on $\mathcal{X}$.

Proof.

Requires some technicalities beyond the scope of the course, see notes for some intuition. □

Visualizing Kernels and RKHSs

- We know that functions in $\mathcal{H}_k$ are of the form $f(x) = \sum_{i=1}^{r} a_j k(x, x_i)$ (or pointwise limits of these)
- Solutions to ERM problems will further have $r = n < \infty$ and the x_i will be our datapoints

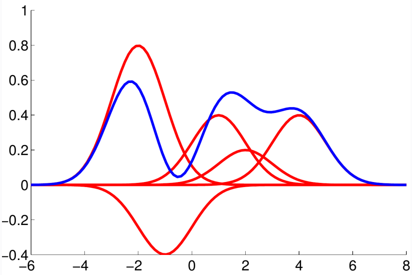

Figure 1: Visualizing $\mathcal{H}_k$ for $k(x, x') = \exp\left(-\frac{1}{2\gamma^2}\|x - x'\|_2^2\right)$

This RBF kernel has an RKHS corresponding to infinitely differentiable functions

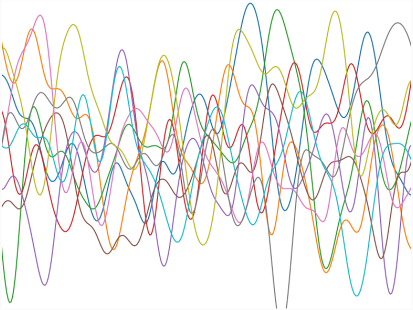

Figure 2: Example functions from $\mathcal{H}_k$ for RBF kernel (allowing $r = \infty$ but with restrictions on $\|f\|_{\mathcal{H}_k}$). Source:

https://stackoverflow.com/questions/46334298/kernel-function-in-gaussian-processes

Matérn Kernels

Allowing infinite differentiable functions is often overly restrictive, Matérn kernels allow for less smooth functions.

The introduce an additional hyperparameter ν and are $s-$times differentiable if an only if $\nu > s$.

Though we omit their full form here (see notes), we note they have simplified forms when $\nu = s + 1/2$:

- $\nu = \frac{1}{2}$: $k(x, x') = \exp\left(-\frac{1}{\gamma}\left\|x - x'\right\|_2\right)$,
- $\nu = \frac{3}{2}$: $k(x, x') = \left(1 + \frac{\sqrt{3}}{\gamma}\left\|x - x'\right\|_2\right)\exp\left(-\frac{\sqrt{3}}{\gamma}\left\|x - x'\right\|_2\right)$.

Exercise: prove that $f \in \mathcal{H}_{k_\nu}$ is s times differentiable for these ν

Matérn Kernels

- As $\nu \to \infty$ the Matérn kernel converges to the RBF kernel
- $\|f\|^2_{\mathcal{H}_k}$ directly penalizes their derivatives, e.g. for $\nu = 3/2$

$$\|f\|^2_{\mathcal{H}_k} \propto \int f''(x)^2 dx + \frac{6}{\gamma^2} \int f'(x)^2 dx + \frac{9}{\gamma^4} \int f(x)^2 dx.$$

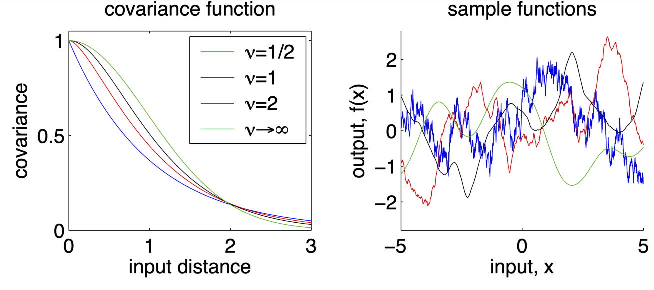

Figure 3: Characterization of Matérn kernels. Source: Rasmussen and Williams, Gaussian Processes for Machine Learning, 2005

Other Example Kernels

- **Constant** $k(x, x') = c$
- **Linear**: $k(x, x') = x^\top x'$
- **Polynomial**: $k(x, x') = (c + x^\top x')^m$, $c \in \mathbb{R}$, $m \in \mathbb{N}$ ($m = 1$ gives affine kernel)
- **Periodic (1d)**: $k(x, x') = \exp\left(-\frac{2\sin^2(\pi|x-x'|/p)}{\gamma^2}\right)$, period p, $\gamma > 0$
- **Laplace**: $k(x, x') = \exp\left(-\frac{1}{\gamma}\|x - x'\|_2\right)$, $\gamma > 0$ (equivalent to Matérn $1/2$, associated with Brownian motion)
- **Rational quadratic**: $k(x, x') = \left(1 + \frac{\|x-x'\|_2^2}{2\alpha\gamma^2}\right)^{-\alpha}$, $\alpha, \gamma > 0$ (see derivation in notes)

Kernel Ridge Regression

Kernel ridge regression is the kernelized version of regularized least squares linear regression

$$
\begin{aligned}
f^* &= \underset{f \in \mathcal{H}_k}{\arg\min} \left(\sum_{i=1}^{n} \left(y_i - \langle f, k(\cdot, x_i) \rangle_{\mathcal{H}}\right)^2 + \lambda \|f\|_{\mathcal{H}_k}^2 \right) \\
&= \underset{f \in \mathcal{H}_k}{\arg\min} \left(\sum_{i=1}^{n} \left(y_i - f(x_i)\right)^2 + \lambda \|f\|_{\mathcal{H}_k}^2 \right) \\
&= \sum_{i=1}^{n} \alpha_i k(\cdot, x_i),
\end{aligned}
$$

by the representer theorem.

See examples sheet for how we find the α_i.

Hyperparameters

- Even if the RKHS is very general, hyperparameters can still heavily influence what is learned in practice for **finite** data
- In particular, the common parameters of the length scale γ and regularization strength λ can be particularly important.

[Coding examples]

Limitations of Kernels in High Dimensions

- Many common kernels are based only on Euclidean distances between points in the original space, e.g. $k(x, x') = \exp\left(-\frac{1}{2\gamma^2}\|x - x'\|_2^2\right)$
- This can lead to poor performance in high dimensions where such pairwise distances are not very informative: all points may be quite far away from each other
- This is not a limitation of kernel methods per se, put reflects the difficulty of constructing appropriate kernels for high dimensional problems
    - Here the machine learning challenge is typically more that of asserting which points are similar than it is of ensuring our predictor is sufficiently powerful; using the kernel trick is of limited help in this endeavor

Further Reading

- Go have a play: these things are super easy to code up and have a mess around with them is a good way to develop an understanding
- Chapter 4 of Carl Edward Rasmussen and Christopher Williams. **Gaussian Processes for Machine Learning**. The MIT Press, 2005, `http://www.gaussianprocess.org/gpml/chapters/` (will require some knowledge of Gaussian processes that we will cover later in the course)