Reproducing Kernel Hilbert Spaces

David Sharkansky

2024-03-14

A common problem in statistics is trying to find the right function to fit a set of data points. This can be a simple problem involving a linear regression, sometimes with a quick transformation of variables. However, many real-life problems are too complicated to make assumptions about the shape of the function. There may be several predictor variables whose impact on the response variable is unclear.

One solution to this problem is choosing the best function to fit the data within a reproducing kernel Hilbert space (RKHS). By using a RKHS, we gain access to a larger family of functions than if we had made more assumptions about the data.

A RKHS is a function space with an inner product and is based around a kernel function $K(x, y)$. The kernel function must have the property that a matrix with entries determined K must be positive semidefinite. That is, given two sets of points $\{x_n\}$ and $\{y_n\}$, the matrix A with $A_{ij} = K(x_i, y_j)$ must be positive semidefinite.

Additionally, the kernel of a RKHS must satisfy what is called the reproducing property. Fixing x, $K(x, y)$ is a function of y contained in the RKHS and satisfies the equation $\langle f, K(x, y)\rangle = f(y)$. Given any function that has the properties described, we can construct a unique RKHS.

If we have a kernel K, we can use it to generate a function to fit a set of points $\{(x_i, y_i)\}_{i=1}^n$. We can use the matrix $\boldsymbol{K}$ given by $\boldsymbol{K}_{ij} = K(x_i, x_j)$ to generate the function $\widehat{f}$ if $y \in \text{range}(\boldsymbol{K})$. Our function is given by $\widehat{f}(\cdot) = \sum_{i=1}^n \widehat{\alpha}_i K(\cdot, x_i)$, where $\boldsymbol{K}\widehat{\alpha} = y$.

To account for errors, we want our function to be smoother to avoid overfitting the points. When calculating $\widehat{\alpha}$, we can add a penalization parameter λ. Then we have $\widehat{\alpha} = (\boldsymbol{K} + \lambda I_n)^{-1} y$, where I_n is the identity matrix of size n. We adjust λ to make our function as smooth as we need.

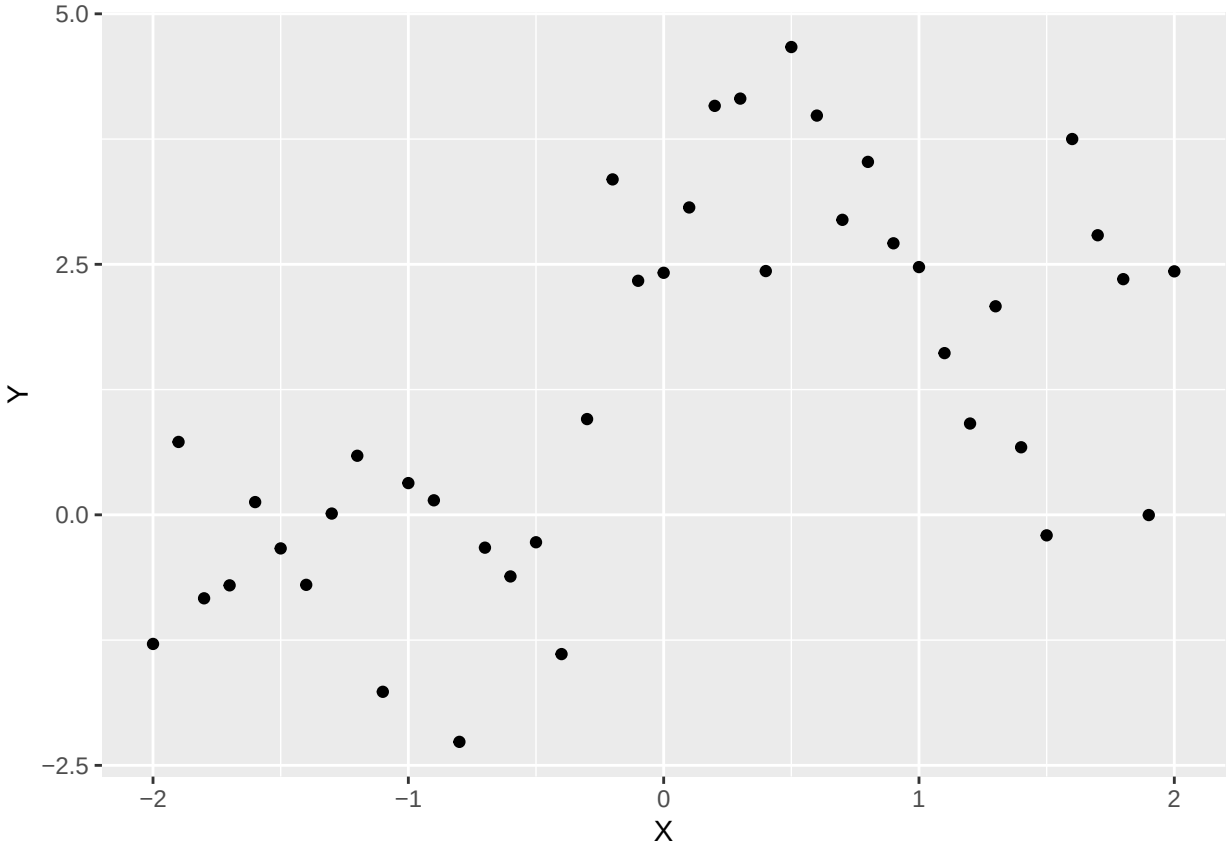
One example of a kernel we can use is the linear kernel $K(x, y) = 1 + xy$. Using this kernel, we can generate any linear function. We can also extend this to multiple dimensions as $K(x, y) = 1 + \langle x, y\rangle$. Adding an exponent gives polynomial kernel $K(x, y) = (1 + \langle x, y\rangle)^p$, which has polynomials of degree p or less.

Another popular choice of kernel is the Gaussian kernel $K(x, y) = \exp(-\frac{1}{2\sigma^2}||x-y||^2)$, where σ is a bandwidth parameter. The Gaussian kernel is relatively small, as it consists of infinitely differentiable functions.

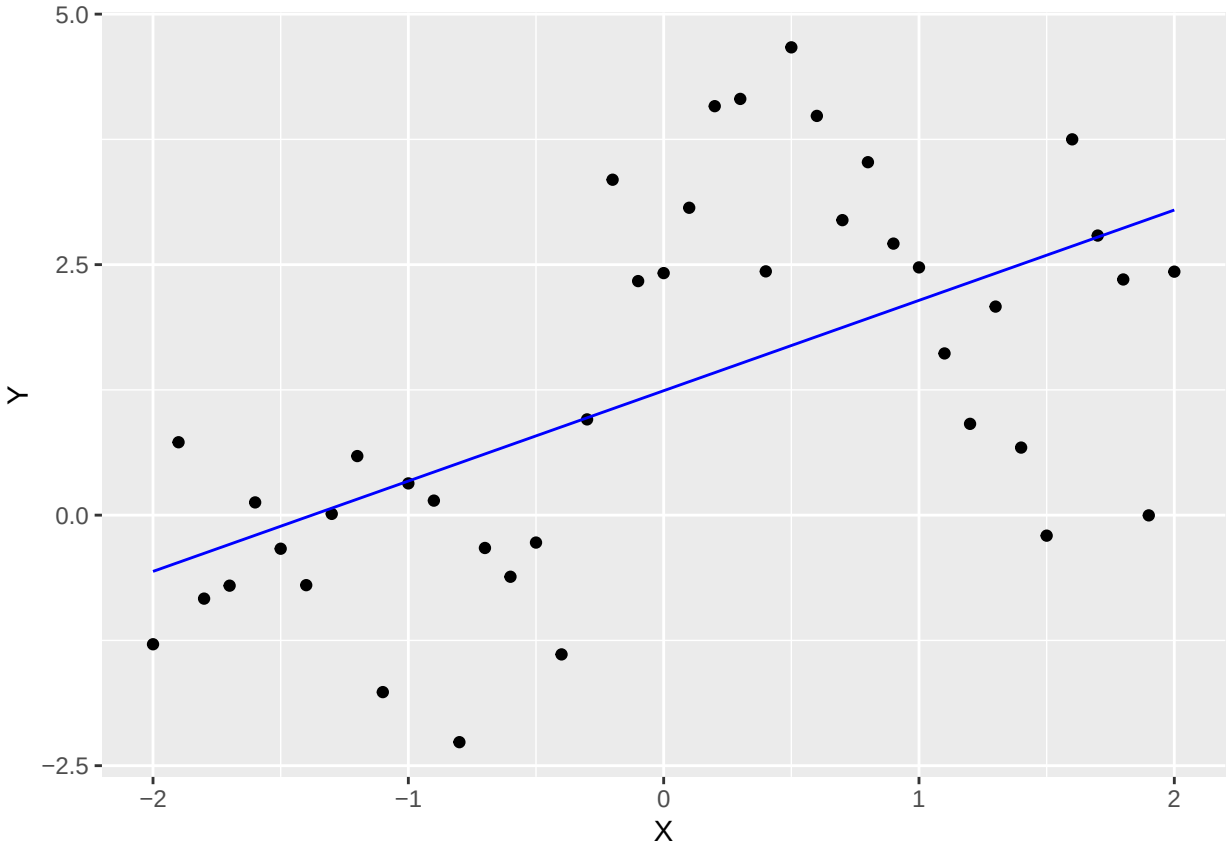
If we want a larger space of functions, we can use a Sobolev space, which contains all n-times differentiable functions. The first-order Sobolev space of once-differentiable functions is generated from the kernel $K(x, y) = 1+\min\{x, y\}$. There also exist higher-order Sobolev spaces of functions that are n-times differentiable for any fixed integer n.

Now that we have explained what a RKHS is, we will apply the concepts described here.

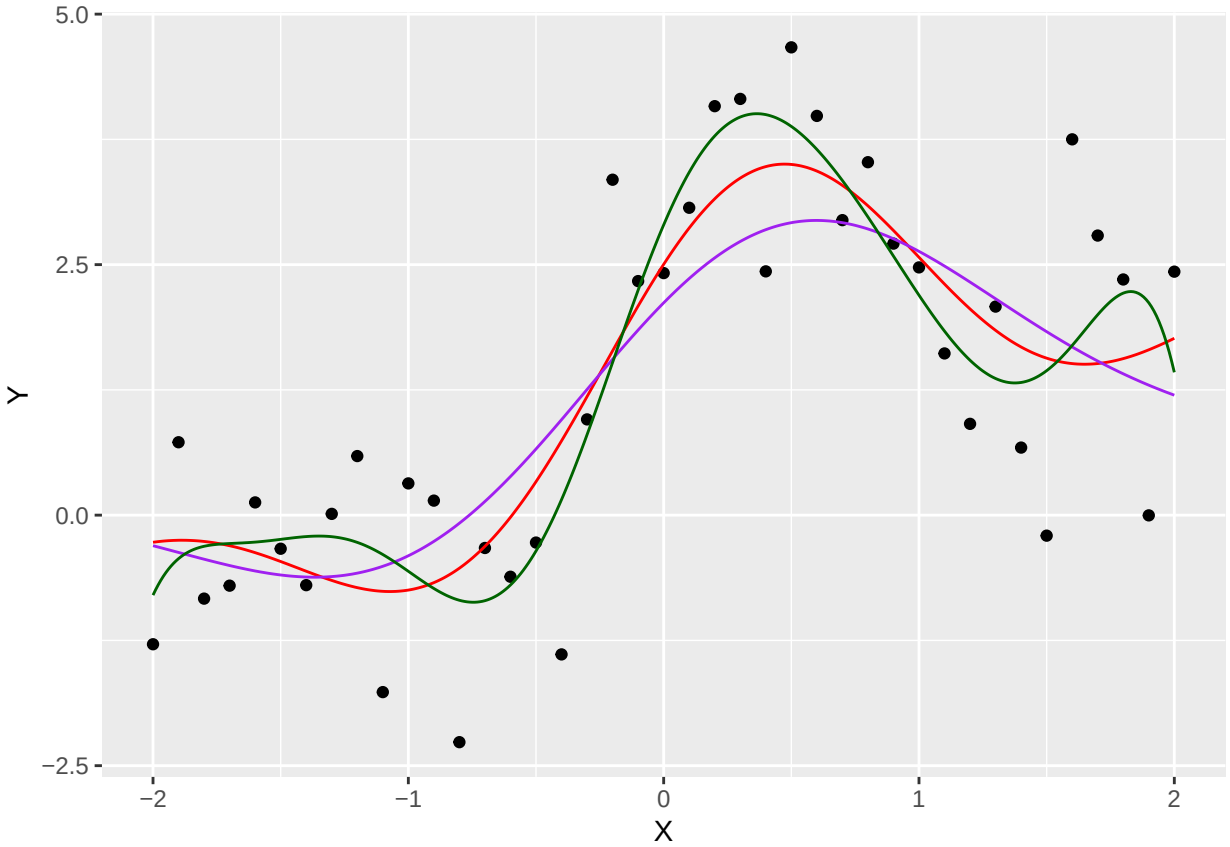
To demonstrate how to use a RKHS to fit a function, let's start with a set of points. We will take an evenly-spaced set of x-values and generate the y-values from $f(x) = \exp(x^3 - 3x^2 + 2x + 1) + \epsilon$, where $\epsilon \sim N(0, 1)$.



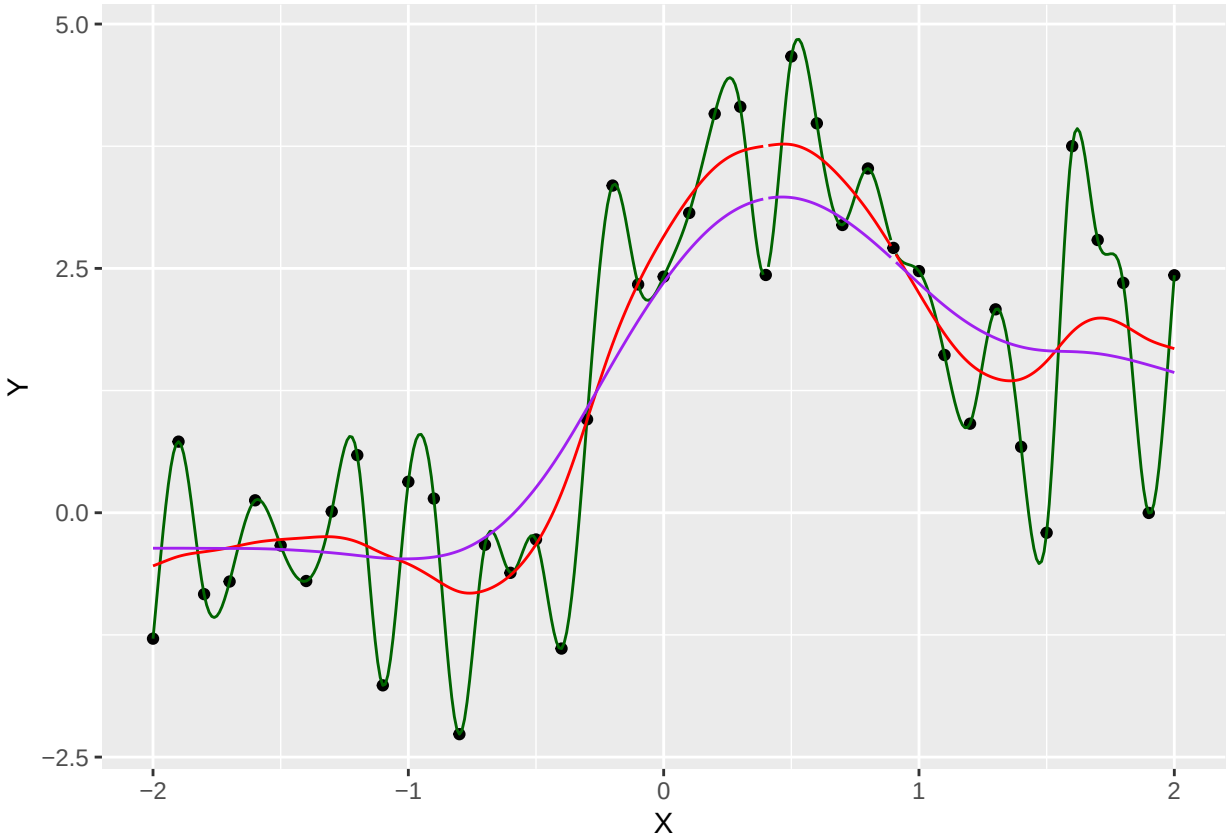
It is not clear what kind of function would best fit these data. There does appear to be some kind of positive correlation but we may not be able to make any assumptions about the best fit function. First we will try the linear kernel.



This function is not a great fit for the data points. In the middle of the range of points, the function underestimates y, but overestimates at the ends of the range. Next we will try the Gaussian kernel with a few different values for the penalization parameter λ.



Here we see that the Gaussian kernel does a better job of approximating the data. The green function has no penalization, while red and purple have $\lambda = 0.1$ and $\lambda = 1$ respectively. We can see how removing the penalization parameter results in a more erratic function, while increasing λ gives a smoother curve. Now we will try a Sobolev kernel.



With no penalization, the Sobolev space is able to perfectly fit the data points. However, it is very unlikely that the true function would resemble this. Here, the choice of λ is very important so that our function is able to predict future data points more accurately.