Notes on the Kernel Trick

Cyrus Samii
NYU Politics – Quant II
April 2017

Let's start simple and suppose a bivariate regression problem for which we have (y_i, x_i) draws. Suppose the functional relationship appears rather complicated, and so we want to entertain an Mth-order polynomial:

$$f(x_i, \beta) = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \ldots + \beta_M x_i^M = \sum_{k=0}^{M} w_k x_i^k.$$

The MMSE problem is

$$\min_{\beta} L(\beta) = \sum_{i=1}^{N} (y_i - f(x_i, \beta))^2 = (y - \mathbf{M}\beta)'(y - \mathbf{M}\beta),$$

where $\mathbf{M}$ is the matrix stacking the expansion vectors, $m(x_i) = (1, x_i, \ldots, x_i^M)$.

An issue with a model like this is the potential for overfitting. So, we might consider regularizing with a ridge penalty, yielding

$$\min_{\beta} L(\beta, \lambda) = (y - \mathbf{M}\beta)'(y - \mathbf{M}\beta) + \lambda\beta'\beta,$$

for $\lambda > 0$.

FOC:

$$\begin{aligned} \mathbf{M}'\mathbf{M}\hat{\beta} - \mathbf{M}'y + \lambda\hat{\beta} &= 0 \\ \Rightarrow \hat{\beta} &= (\mathbf{M}'\mathbf{M} + \lambda\mathbf{I})^{-1}\mathbf{M}'y \end{aligned}$$

which is the usual solution, incorporating the ridge penalty.

But we can also note from the FOC that,

$$\begin{aligned} \mathbf{M}'\mathbf{M}\hat{\beta} - \mathbf{M}'y + \lambda\hat{\beta} &= 0 \\ \Rightarrow \hat{\beta} &= \frac{1}{\lambda}(\mathbf{M}'y - \mathbf{M}'\mathbf{M}\hat{\beta}) \\ &= \mathbf{M}'\frac{1}{\lambda}(y - \mathbf{M}\hat{\beta}) \\ &= \mathbf{M}'\alpha(y), \end{aligned}$$

where

$$\alpha(y) = \frac{1}{\lambda}(y - \mathbf{M}\hat{\beta}).$$

Now substitute $\hat{\beta} = \mathbf{M}'\alpha(y)$ into the expression for $\alpha(y)$, and you get,

$$\begin{aligned}
\alpha(y) &= \frac{1}{\lambda}(y - \mathbf{M}\mathbf{M}'\alpha(y)) \\
\lambda\alpha(y) &= y - \mathbf{M}\mathbf{M}'\alpha(y) \\
\mathbf{M}\mathbf{M}'\alpha(y) + \lambda\alpha(y) &= y \\
(\mathbf{M}\mathbf{M}' + \lambda\mathbf{I})\alpha(y) &= y \\
\alpha(y) &= (\mathbf{M}\mathbf{M}' + \lambda\mathbf{I})^{-1}y.
\end{aligned}$$

This implies that we can generate a predicted value for covariate value x_j by using the expansion $m(x_j)$ and computing,

$$\begin{aligned}
\hat{y}_j &= m(x_j)'\hat{\beta} \\
&= m(x_j)'\mathbf{M}'\alpha(y) \\
&= m(x_j)'\sum_{i=1}^{N} m(x_i)\alpha_i \\
&= \sum_{i=1}^{N} \alpha_i(y)\langle m(x_j), m(x_i)\rangle
\end{aligned}$$

Note as well that in the expression,

$$\alpha(y) = (\mathbf{M}\mathbf{M}' + \lambda\mathbf{I})^{-1}y,$$

we have,

$$\mathbf{M}\mathbf{M}' = \begin{pmatrix} \langle m(x_1), m(x_1)\rangle & \langle m(x_1), m(x_2)\rangle & \cdots \\ \vdots & \ddots & \\ \langle m(x_1), m(x_N)\rangle & \cdots & \end{pmatrix}$$

As such, our $\hat{y}_j$ estimator can be cast in terms of a function that works in distances in the $m(\cdot)$-transformed covariate space. In other words, we are working with norms on the $m(\cdot)$ transformed space. Let's write,

$$k(x_j, x_i) = \langle m(x_i), m(x_j)\rangle$$

Then our estimator for $\hat{y}_j$ can be expressed as,

$$\hat{y}_j = \sum_{i=1}^{N} \alpha_i^k(y)k(x_j, x_i),$$

where the k superscript in $\alpha_i^k(y)$ denotes that we are using the $k(\cdot)$ function:

$$\alpha_i^k(y) = (\mathbf{K} + \lambda\mathbf{I})^{-1}y,$$

with $K_{ij} = k(x_i, x_j)$.

A $k(\cdot)$ function like this is known as a *kernel*, and serves as a way to measure potentially non-linear distances between points in space. Now, we have motivated the kernel in

terms of the $m(\cdot)$ function, which served as a polynomial expansion. As it happens, the representation

$$\hat{y}_j = \sum_{i=1}^{N} \alpha_i^k(y) k(x_j, x_i),$$

shows $\hat{y}_j$ as a sort of *kernel-weighted average* of the y values. Indeed, we can work with this representation directly, and consider kernels that are more immediately intuitive in the ways that they treat distance in the covariate space—e.g., Gaussian kernels, etc.

These results demonstrate a *duality* whereby we can go back and forth between thinking of regression estimators in terms of regression with series expansion or as kernel-weighted averages. Of course, everything here generalizes to the case of multiple regressors.