Local Regression Smoothing in p Dimensions

Stephen Vardeman

Analytics Iowa LLC

ISU Statistics and IMSE

Direct generalization of 1-D local regression

A direct generalization of 1-dimensional kernel smoothing to p dimensions might go roughly as follows. For D as before, and $\mathbf{x} \in \Re^p$, I might set

$$K_\lambda(\mathbf{x}_0, \mathbf{x}) = D\left(\frac{\|\mathbf{x} - \mathbf{x}_0\|}{\lambda}\right) \tag{1}$$

and fit linear forms locally by choosing $\alpha(\mathbf{x}_0) \in \Re$ and $\boldsymbol{\beta}(\mathbf{x}_0) \in \Re^p$ solving the problem

$$\underset{\alpha \text{ and } \boldsymbol{\beta}}{\text{minimize}} \sum_{i=1}^{N} K_\lambda(\mathbf{x}_0, \mathbf{x}_i)\left(y_i - \left(\alpha + \boldsymbol{\beta}'\mathbf{x}_i\right)\right)^2$$

and predicting as

$$\hat{f}_\lambda(\mathbf{x}_0) = \alpha(\mathbf{x}_0) + \boldsymbol{\beta}'(\mathbf{x}_0)\,\mathbf{x}_0$$

Structure assumptions

Kernel smoothing in p dimensions should be done only after standardizing the coordinates of $\mathbf{x}$, and can be effective as longs as N is not too small and p is not more than 2 or 3. For $p > 3$, the curse of dimensionality comes into play and N points usually just aren't dense enough in p-space to make direct use of kernel smoothing effective. For the method to be successful $\Re^p$ it will usually need to be applied under appropriate structure assumptions.

One way to apply additional structure to the p-dimensional kernel smoothing problem is to essentially reduce input variable dimension by replacing the kernel (1) with the "structured kernel"

$$K_{\lambda,\mathbf{A}}(\mathbf{x}_0, \mathbf{x}) = D\left(\frac{\sqrt{(\mathbf{x}-\mathbf{x}_0)' \mathbf{A} (\mathbf{x}-\mathbf{x}_0)}}{\lambda}\right)$$

for an appropriate non-negative definite matrix $\mathbf{A}$.

Structure assumptions cont.

For the eigen decomposition of

$$\mathbf{A} = \mathbf{V}\mathbf{D}\mathbf{V}'$$

write

$$(\mathbf{x} - \mathbf{x}_0)' \mathbf{A} (\mathbf{x} - \mathbf{x}_0) = \left(\mathbf{D}^{\frac{1}{2}}\mathbf{V}' (\mathbf{x} - \mathbf{x}_0)\right)' \left(\mathbf{D}^{\frac{1}{2}}\mathbf{V}' (\mathbf{x} - \mathbf{x}_0)\right)$$

This replaces use of $\mathbf{x}$ and $\Re^p$ distance from $\mathbf{x}$ to $\mathbf{x}_0$ to define weights, with $\mathbf{D}^{\frac{1}{2}}\mathbf{V}'\mathbf{x}$ and $\Re^p$ distance from $\mathbf{D}^{\frac{1}{2}}\mathbf{V}'\mathbf{x}$ to $\mathbf{D}^{\frac{1}{2}}\mathbf{V}'\mathbf{x}_0$. When some entries of $\mathbf{D}$ are 0 (or are nearly so), one basically reduces dimension from p to the number of large eigenvalues of $\mathbf{A}$ and defines weights in a space of that dimension (spanned by eigenvectors corresponding to non-zero eigenvalues) where the curse of dimensionality may *not* preclude effective use of kernel smoothing. "The trick" is, of course, identifying the right directions into which to project. (Searching for such directions is part of the Friedman "projection pursuit" idea.)