MOVING BEYOND MOVING AVERAGES IN DATA INTERPOLATION

ANTHONY D. BLAOM

This article discusses a method for interpolating data that is a clever fusion of two methods you have probably encountered before: moving averages, and simple linear regression. Before describing this method, known as *locally linear regression*, we review moving average interpolation.

1. Moving averages

Suppose that we seek an interpolating function $y(x)$ for the five data points $(x_1, y_2), (x_2, y_2), \ldots (x_5, y_5)$ tabulated below:

i	1	2	3	4	5
x_i	-2	-1	0	1	2
y_i	0.5	1.5	2	1.5	0.5

We will refer to the x_i's as *inputs* and the y_i's as *outputs*. Then, in the method of moving averages, one declares $y(x)$ to be the average of all those outputs y_i for which the corresponding input x_i is sufficiently close to x, in the sense that $|x - x_i| < h$. Here h is a parameter fixed before-hand and called the *bandwidth*. Figure 1 shows the data tabulated above, together with the moving average interpolator for $h = 1.5$.

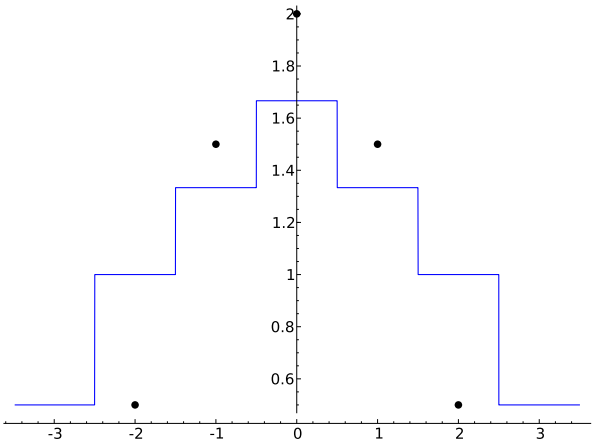

FIGURE 1. A moving average interpolator for the data shown in the previous figure. Here the bandwidth is $h = 1.5$.

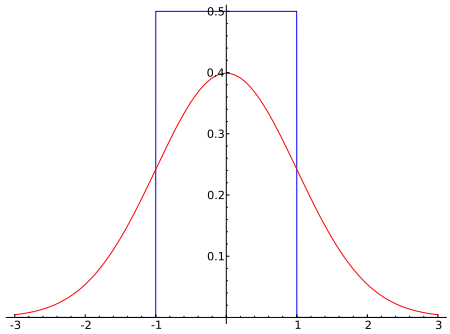

FIGURE 2. Comparison of the boxcar (blue) and Gaussian (red) kernels.

The interpolator $y(x)$ could be criticized on two grounds: it is not particularly smooth; and, for small bandwidths, $y(x)$ need not be defined, because there may be *no* x_i sufficiently close to x, and so we averaging an empty set of numbers. Before addressing this, let us first express $y(x)$ using an explicit formula: If we let $k(x)$ denote the so-called *boxcar* function, defined by

$$k(x) = \begin{cases} \frac{1}{2} & \text{if } |x| < 1 \\ 0 & \text{if } |x| \geq 1 \end{cases},$$

then we can write

$$y(x) = \frac{k(\frac{x-x_1}{h})y_1 + k(\frac{x-x_2}{h})y_2 + k(\frac{x-x_3}{h})y_3 + k(\frac{x-x_4}{h})y_4 + k(\frac{x-x_5}{h})y_5}{k(\frac{x-x_1}{h}) + k(\frac{x-x_2}{h}) + k(\frac{x-x_3}{h}) + k(\frac{x-x_4}{h}) + k(\frac{x-x_5}{h})}. \quad (1)$$

For example, with $h = 1.5$, this formula predicts

$$y(0.25) = \frac{\frac{1}{2}y_2 + \frac{1}{2}y_3 + \frac{1}{2}y_3}{\frac{1}{2} + \frac{1}{2} + \frac{1}{2}} = \frac{y_1 + y_2 + y_3}{3},$$

as required.

To address the criticisms above we replace the boxcar function — known as the *kernel* of the interpolator — with a "smeared out" version (see Figure 2). Any smooth, positive, symmetric function having the same total integral (namely one) will serve our purposes; we choose the Gaussian function,

$$k(x) = \frac{1}{\sqrt{2\pi}} e^{-x^2/2}, \quad (2)$$

but there are several other commonly used kernels. The interpolator is defined by Equation (1) as before, which may be compactly written

$$y(x) = \sum_{i=1}^{N} l_i(x) y_i, \quad \text{where } l_i(x) = \frac{k(\frac{x-x_i}{h})}{\sum_{i=1}^{N} k(\frac{x-x_i}{h})}.$$

Here N is the total number of data points, in this case five. Gaussian kernel interpolators, for three different bandwidths, are shown in Figure 3 below.

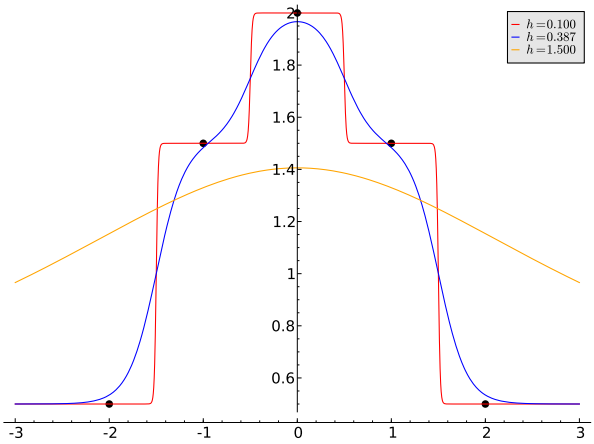

FIGURE 3. Gaussian kernel moving average interpolation. Interpolators for three different bandwidths h are shown.

The yellow curve corresponds to the interpolator with the same bandwidth as the boxcar interpolator in Figure 1 above.

Figure 3 suggests that in the limit $h \to 0$, the interpolator fits the data perfectly, and is constant in the neighborhood of each input data point x_i. We will return to this aspect of moving average interpolation in §2.

When interpolating real data one must avoid the problem of overfitting and choose a non-zero bandwidth, generally selected by cross-validation, which we discuss in §4.

2. Locally linear regression

We assume the reader is familiar with simple linear regression, i.e., with the construction of a straight line interpolator $y(x) = mx + c$ minimizing the *training error*,

$$\hat{E} = \frac{1}{N}\sum_{i=1}^{N}(y(x_i) - y_i)^2. \tag{3}$$

Consider for a moment a cruder form of interpolation, in which one seeks a *constant* function $y(x) = c$ (i.e., horizontal line) minimizing the training error $\hat{E}$. Substituting $y(x) = c$ into (3), we minimize $\hat{E}$ by differentiating with respect to c and equating to zero. This gives $c = \frac{1}{N}\sum_{i=1}^{N} y_i$, i.e., the average value of the outputs. Fine, but not very interesting.

Now *fix* a particular input value x, and repeat the above "constant regression" exercise; but, to make the interpolation more relevant for inputs close to x, *penalize most those squared residuals* $(y(x_i) - y_i)^2$ *at inputs* x_i *closest to* x. We can do this by defining a x-dependent training error $\hat{E}(x)$

as the following weighted-sum of squared residuals:

$$\hat{E}(x) = \frac{1}{N}\sum_{i=1}^{N} k\left(\frac{x_i - x}{h}\right)(y(x_i) - y_i)^2. \tag{4}$$

Here k is the Gaussian kernel defined in (2) and $h > 0$ a parameter controlling how localized the error penalty should be. Substituting c for $y(x_i)$ on the right of (4), and using calculus as before to find the c value minimizing $\hat{E}(x)$, one derives the result,

$$c = \sum_{i=1}^{N} l_i(x)y_i, \quad \text{where } l_i(x) = \frac{k(\frac{x-x_i}{h})}{\sum_{i=1}^{N} k(\frac{x-x_i}{h})}.$$

But this is precisely the prediction at x of the moving average interpolator! For this reason, moving average interpolation is also known as *locally constant regression*.

We can now explain the locally constant behavior of moving average interpolators observed in §1 (for small bandwidth h): As the bandwidth decreases, the penalty for residuals in the training error $\hat{E}(x)$, at some input x close to x_i, is dominated by the single contribution $k(0)(y(x_i) - y_i)^2$. So, close to x_i, the constant-valued interpolator is $y = y_i$.

Locally *linear* regression improves on locally constant regression in the obvious way: For each input x, the output prediction $y(x)$ is obtained by finding the straight line $y'(x') = mx' + c$ minimizing the weighted-sum of squared residuals $\hat{E}(x)$, and then setting $y(x) = y'(x)$. When the dust settles (see the Appendix for details), one obtains an interpolator $y(x)$ with a similar form:

$$y(x) = \sum_{i=1}^{N} l_i(x)y_i, \quad \text{where } l_i(x) = \frac{g_i(x)}{\sum_{j=1}^{N} g_j(x)} \tag{5}$$

$$\text{and } g_j(x) = \left(\sum_{i=1}^{N}(x_i - x_j)(x_i - x)K\left(\frac{x_i - x}{h}\right)\right)K\left(\frac{x_j - x}{h}\right). \tag{6}$$

Locally linear regression interpolators for the five-point data set analyzed in §1 are plotted in Figure 4. Notice that in the limit $h \to 0$, the interpolator becomes a piece-wise *linear* interpolator fitting the data perfectly.

3. Comparison of locally constant and locally linear regression

Figures 5 and 6 show the results of applying locally constant and locally linear regression to experiments performed on six subjects injected with the drug Indomethacin [3]. In both cases the bandwidth was selected by generalized cross-validation (see §4 below).

The performance of the two interpolators can be compared by comparing their generalized cross-validation errors, E_{GCV}. In the locally constant case, we have $\sqrt{E_{\text{GCV}}} = 0.0200\mu\text{g/ml}$; locally linear regression is only a tad better,

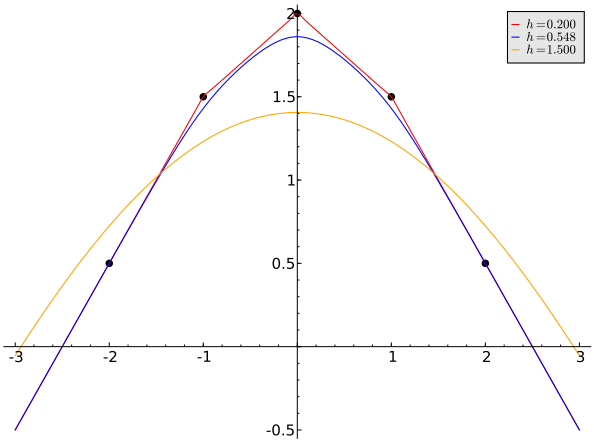

FIGURE 4. Locally linear regression (with Gaussian kernel). Interpolators for three different bandwidths h are shown.

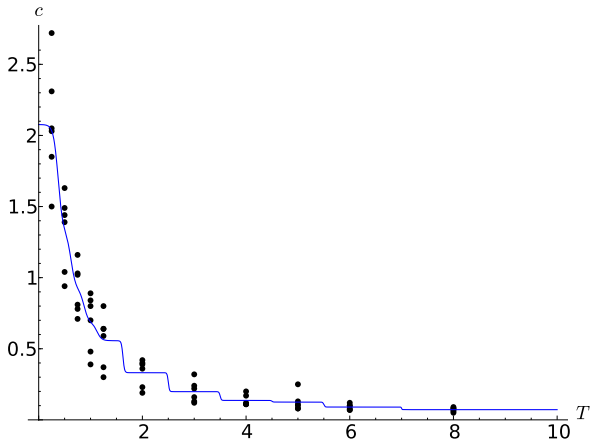

FIGURE 5. Locally constant regression applied to the Indomethacin data of [3]; c is concentration of Indomethacin in μg/ml, and T the time in hours. A Gaussian kernel bandwidth of $h = 0.106$ was chosen by generalized cross-validation.

with $\sqrt{E_{\mathrm{GCV}}} = 0.0199\mu$g/ml. However, the locally constant regressor has a less regular, and clearly unrealistic, step-like shape, as well as indications of boundary bias (look at predictions for small values of T in Figure 5).

From purely theoretical considerations, one can show that boundary bias, as well as *design bias* (sensitivity to the form of the distribution of the input data), are symptomatic of locally constant regression (a.k.a. moving average interpolation). By contrast, locally *linear* regression has no design bias, and the boundary bias is much less; specifically it is asymptotically of order h^2 instead of order h. For details see Fan [1] and Fan and Gijbels [2], or the textbook [4].

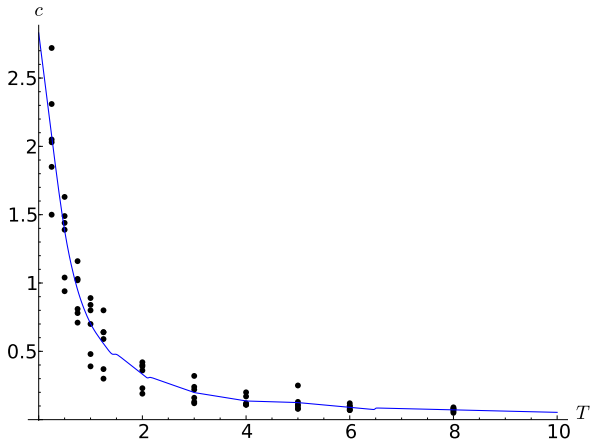

FIGURE 6. Locally linear regression applied to the Indomethacin data of [3]; c is concentration of Indomethacin in μg/ml, and T the time in hours. A Gaussian kernel bandwidth of $h = 0.155$ was chosen by generalized cross-validation.

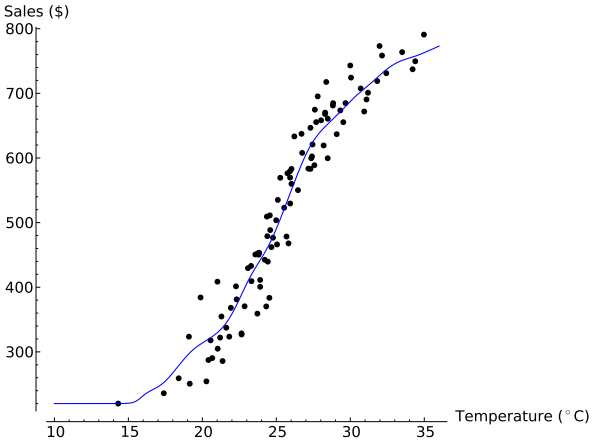

FIGURE 7. Locally constant regression applied to the daily sales of a chain of convenience stores, as a function of maximum daily temperature. Bandwidth chosen by generalized cross-validation.

Figures 7 and 8 show the results of applying locally constant and locally linear regression to the daily icecream sales of a (fictitious) chain of convenience stores, as a function of temperature. Performance of locally linear regression is slightly better with $\sqrt{E_{\text{GCV}}} = \$38.80$, compared with $\sqrt{E_{\text{GCV}}} = \$39.80$ for for locally constant regression.

4. Cross-validation and bandwidth selection

To avoid overfitting in local regression one chooses a bandwidth minimizing a cross-validation error. Fixing a particular interpolation scheme (e.g.,

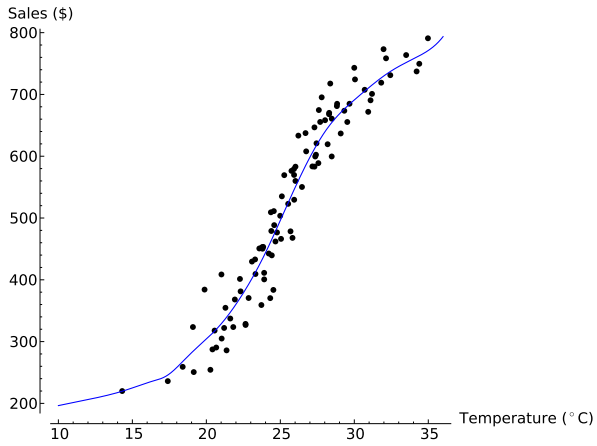

FIGURE 8. Locally linear regression applied to the daily sales of a chain of convenience stores, as a function of maximum daily temperature. Bandwidth chosen by generalized cross-validation.

locally linear regression), the *leave-one-out cross-validation error* is defined by

$$E_{\mathrm{CV}} = \frac{1}{N}\sum_{i=1}^{N}(y^{(i)}(x_i) - y_i)^2,$$

where $y^{(i)}$ is the interpolator obtained by applying the fixed scheme to the data *with the ith data point removed*. In general, the cross-validation error is computationally expensive. However, for many interpolators of the general form (5) (known as *linear smoothers*), including both locally constant and locally linear regression, one has the following simplification in the prediction $y^{(i)}(x_i)$:

$$y^{(i)}(x_i) = \frac{y(x_i) - l_i(x_i)y_i}{1 - l_i(x_i)}.$$

Here $y(x_i)$ is the prediction at the input x_i of the interpolator y obtained by applying the given scheme to the *full* data set. Applying this simplification to the definition of E_{CV}, we obtain,

$$E_{\mathrm{CV}} = \frac{1}{N}\sum_{i=1}^{N}\left(\frac{y(x_i) - y_i}{1 - l_i(x_i)}\right)^2 \tag{7}$$

The sum,

$$\nu = \sum_{i=1}^{N} l_i(x_i)$$

is known as the *effective degrees of freedom*. If, on the right-hand side of (7), we approximate each $l_i(x_i)$ $(i = 1, 2, \ldots, N)$ by its average value ν/N,

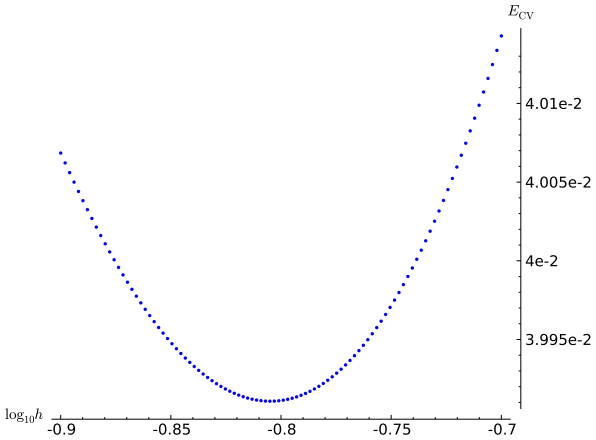

FIGURE 9. A plot of the generalized cross-validation error E_{GVC} against $\log_{10} h$, for locally linear regression applied to the Indomethacin data of [3]. Here h denotes the Gaussian kernel bandwidth.

then we obtain an approximation of E_{CV} known as the *generalized cross-validation error*:

$$E_{\text{GCV}} = \frac{1}{N(1-\nu/N)^2} \sum_{i=1}^{N} (y(x_i) - y_i)^2 = \frac{\hat{E}}{(1-\nu/N)^2},$$

where $\hat{E}$ is the training error for the full data set. For locally constant and locally linear regression, both $\hat{E}$ and ν are functions of the bandwidth, h. Usually, the bandwidth that minimizes E_{GCV} is close enough to the bandwidth minimizing E_{CV} that in common practice the proxy E_{GVC} is used.

Figure 9 shows a plot of E_{GVC} against $\log_{10} h$ for locally linear regression, as applied to the Indomethacin data of the previous section.

5. Beyond locally linear regression

Locally linear regression is a special case of a large and well-studied family of techniques known as *nonparametric regression.* While one can extend locally linear regression to the case of several input variables (multiple regression) the "curse of dimensionality" currently restricts its application to problems of fairly low dimension. For high-dimensional data, tree-based models, such as random forests and gradient boosted trees, are currently popular non-parametric alternatives.

Appendix A. Derivation of the formula for locally linear regression

Locally linear regression has been motivated and defined in §2. Here we furnish a detailed derivation of Equations (5) and (6).

Fix an input x. Then every straight line in the input-output space is described by a function $y'(x') = m(x'-x)+c$, for some c and m independent of x'. The prediction $y(x)$ of locally linear regression is $c = y'(x)$, where c and m have been chosen so as to minimize the weighted sum of squared residuals,

$$\begin{aligned}\hat{E}(x) &= \frac{1}{N}\sum_{i=1}^{N} w_i(x)(y'(x_i) - y_i)^2, \quad \text{where } w_i(x) = k\left(\frac{x_i - x}{h}\right), \\ &= \frac{1}{N}\sum_{i=1}^{N} w_i(x)(m(x_i - x) + c - y_i)^2.\end{aligned}$$

Since $\hat{E}(x)$ is quadratic in c and m it has a global minimum which can be found by setting the partial derivatives $\partial\hat{E}/\partial c$ and $\partial\hat{E}/\partial m$ to zero. This gives us two linear equations in the two unknowns c and m:

$$\begin{aligned}\left(\sum_i w_i\right) c + \left(\sum_i \xi_i w_i\right) m &= \sum_i w_i y_i \\ \left(\sum_i \xi_i w_i\right) c + \left(\sum_i \xi_i^2 w_i\right) m &= \sum_i \xi_i w_i y_i\end{aligned} \quad . \tag{8}$$

Here $\xi_i = (x_i - x)$ and we have suppressed the dependence of w_i on x.

The determinant for the linear system (8) is

$$\begin{aligned}\Delta &= \left(\sum_i w_i\right)\left(\sum_i \xi_i^2 w_i\right) - \left(\sum_i \xi_i w_i\right)^2 = \sum_i\sum_j (\xi_j - \xi_i)\xi_j w_i w_j \\ &= \sum_j\sum_i (\xi_i - \xi_j)\xi_i w_i w_j = \sum_j g_j, \quad \text{where } g_j = \left(\sum_i (\xi_i - \xi_j)\xi_i w_i\right) w_j.\end{aligned}$$

Notice that, unravelling the definitions of ξ_i and $w_i = w_i(x)$, the definition of g_j here coincides with the one given in (6), §2.

From the general form of solutions to two-variable linear systems, we obtain,

$$\begin{aligned} c\Delta &= \left(\sum_i \xi_i^2 w_i\right)\left(\sum_i w_i y_i\right) - \left(\sum_i \xi_i w_i\right)\left(\sum_i \xi_i w_i y_i\right) \\ &= \sum_j \left(\left(\sum_i \xi_i^2 w_i\right) w_j y_j - \left(\sum_i \xi_i w_i\right) \xi_j w_j y_j \right) \\ &= \sum_j \sum_i (\xi_i - \xi_j)\xi_i w_i w_j y_j = \sum_j g_j y_j \\ \implies c &= \sum_j \left(\frac{g_j y_j}{\Delta}\right) = \sum_j l_j y_j, \end{aligned}$$

where $l_j = \dfrac{g_j}{\sum_i g_i}$, Q. E. D.

References

[1] J. Fan. Design-adaptive non-parametric regression. *J. Amer. Stat. Soc.*, 87:998–1004, 1992.

[2] J. Fan and I. Gijbels. *Local Polynomial Modelling and its Applications*. Chapman and Hall, New York, 1992.

[3] C. Kwan, K. O. Breault, G. R. Umbenhauer, E. G. McMahon, F. and E. Duggan, D.˙Kinetics of Indomethacin absorption, elimination, and enterohepatic circulation. *J. Pharmacokinetics and Biopharmaceutics*, 4:255–280, 1976.

[4] L. Wasserman. *All of Nonparametric Statistics*. Springer, New York, 2006.

January, 2013