Friedman's Super Smoother

Joerg Luedicke
StataCorp
College Station, TX
jluedicke@stata.com

Abstract. This note documents the methods and formulas for the `supsmooth` command which implements a bivariate regression smoother based on local linear regression with adaptive bandwidths. This method is known as Friedman's super smoother. Adaptive bandwidths are especially useful in case of changing degree of curvature in the underlying function and/or non-constant error variance.

Keywords: adaptive bandwidth, local regression, lowess, super smoother

1 Introduction: Friedman's super smoother

Friedman's super smoother is a nonparametric regression estimator based on local linear regression with adaptive bandwidths (Friedman [1984]). The basic idea is to first estimate a number of fixed bandwidth smooths by local linear regression. The leave-one-out cross-validated residuals from each of those initial estimates are then smoothed using a constant bandwidth. Based on the smoothed residuals, the best bandwidths from the initial estimates are selected at each data point over the range of the predictor variable. Those local bandwidths are then smoothed with a constant bandwidth, and the two estimates from the initial estimates with closest bandwidth values to the smoothed bandwidths are selected, and the smoothed outcomes are linearly interpolated. The interpolated points are then smoothed again with a fixed bandwidth, resulting in the final estimate.

2 Adaptive bandwidth local linear regression

2.1 Motivation

Let $x_1...x_n$ and $y_1...y_n$ be n random samples from the joint distribution $P(X,Y)$. The objective is to estimate the conditional expectation $E[Y|X=x]$, such that the expected squared difference $E[Y-f(X)]^2$ is minimized, where $f(X)$ is the true underlying function. Let $P(X,Y)$ be generated from the process

$$Y = f(X) + \epsilon \tag{1}$$

where f is an arbitrary function of X, and ϵ are $i.i.d$ random errors with expectation zero. Then, to estimate $E[Y|X=x]$ we need to find the estimate $\hat{f}(x)$ in

$$y_i = \hat{f}(x_i) + \epsilon_i \tag{2}$$

Local linear regression provides a method for estimating $f(X)$ by locally fitting linear least squares regressions. An optimal estimate for $f(X)$ minimizes the expected squared error of the estimated function and depends greatly on the size of the local window. While local linear regression smoothing is often used with a window size that is constant throughout the range of the predictor variable, situations arise where constant window sizes fail to to produce an optimal estimate. The super smoother provides a method for estimating $\hat{f}(x)$ based on locally adaptive bandwidths, which proves useful in certain situations such as heteroskedastic error variance or a varying degree of curvature in the underlying function, as pointed out in Friedman [1984].

2.2 Local linear regression

One way of estimating $\hat{f}(x_i)$ is to locally fit linear least squares regressions of the form

$$\hat{E}[Y|x_i] = \hat{\alpha} + \hat{\beta}x_j, \quad x_j \in N_i \tag{3}$$

where N defines the local neighborhood around x_i and is the tuning parameter, also known as the bandwidth, which controls the bias-variance trade-off. In the fixed bandwidth case, the size of the window N is constant, while in the adaptive bandwidth case it can vary over the range of the predictor variable. Given a fixed bandwidth J, where J is the number of observations in a window, we can write the local linear estimator as

$$\hat{y}_k = \hat{\alpha} + \hat{\beta}x_k, \quad k = 1, ..., n \tag{4}$$

where $\hat{\alpha}$ and $\hat{\beta}$ are obtained from local fits to data points $i_{-J/2}, ..., i_{+J/2}$, with $x_i \leq x_{i+1}$ for $i = J/2, ..., n - J/2$.

An optimal bandwidth which minimizes the expected squared error

$$e^2(J) = E[Y - f(X|J)]^2 \tag{5}$$

can be obtained by estimating $e^2(J)$ via leave-one-out cross-validation:

$$\hat{e}^2(J) = \frac{1}{n}\sum_{i=1}^{n}[y_i - \hat{f}_{(-i)}(x_i|J)]^2 \tag{6}$$

Minimizing $\hat{e}^2(J)$ then yields the cross-validated optimal bandwidth

$$\hat{e}^2(J_{cv}) = \min_{0<J\leq n} \hat{e}^2(J) \tag{7}$$

The leave-one-out squared residuals can be computed analytically:

$$\hat{e}^2(J) = \frac{1}{n}\sum_{i=1}^{n}[y_i - \hat{f}(x_i|J)]^2 / \left[1 - \frac{1}{J} - \frac{(x_i - \bar{x}_J)^2}{V_J}\right]^2 \tag{8}$$

where $V_J = \sum_{j=i_{-J/2}}^{i_{+J/2}}(x_i - \bar{x}_J)^2$, and $\bar{x}_J = \frac{1}{J}\sum_{j=i_{-J/2}}^{i_{+J/2}} x_i$.

2.3 The super smoother

In order to obtain an estimator with locally adaptive bandwidths, Friedman [1984] proposed to minimize the estimate for

$$e^2(f, J) = E\big[Y - f(X|J(X))\big]^2 \tag{9}$$

with respect to both $f(x)$ and $J(x)$. In order to minimize (9), we first estimate (4) using local linear regression over a grid of values for J. While Friedman [1984] originally proposed to use $J = 0.05n$, $J = 0.2n$, and $J = 0.5n$, `supsmooth` allows for specification of any number of bandwidths in the range $0 < J < n$ to allow for a finer grained bandwidth space. We then compute the cross-validated residuals for each of these initial constant bandwidth estimates by:

$$r_{(i)(J)} = [y_i - \hat{f}(x_i|J)] / \left(1 - \frac{1}{J} - \frac{(x_i - \bar{x}_J)^2}{V_J}\right) \tag{10}$$

and smooth $|r_{(i)(J)}|$ against x_i with bandwidth $J = 0.2n$ to estimate $\hat{e}(f, J|x_i)$ which we use to find the optimal bandwidth at each point:

$$\hat{e}(f, J_{cv}(x_i)|x_i) = \min_J \hat{e}(f, J|x_i) \tag{11}$$

The optimal bandwidths $J_{cv}(x_i)$ are then smoothed again ($J = 0.2n$) against x_i and the two initial estimates with closest bandwidths are selected, subject to

$$J_1 \leq J_{cv}(x_i) \leq J_2 \tag{12}$$

The penultimate smooth is then computed by linearly interpolating between these two initial estimates with respect to $J_{cv}(x_i)$. Finally, the result of the interpolation is then smoothed again with bandwidth $J = 0.05n$.

2.4 Implementation

`supsmooth` is implemented as ado file, with computations being performed in Mata. The parameters of the local linear regressions are estimated by either using the updating algorithm proposed by Friedman [1984], or by actually fitting a least squares model in each window. Using the updating algorithm, intercept $\hat{\alpha}$ and slope $\hat{\beta}$ from (4) are computed as

$$\hat{\alpha} = \bar{y}_J - \hat{\beta}\bar{x}_J \tag{13}$$

$$\hat{\beta} = \frac{C_J}{V_J} \tag{14}$$

with

$$\bar{x}_J = \frac{1}{J} \sum_{j=i_{-J/2}}^{i_{+J/2}} x_i \tag{15}$$

$$\bar{y}_J = \frac{1}{J} \sum_{j=i_{-J/2}}^{i_{+J/2}} y_i \tag{16}$$

$$C_J = \sum_{j=i_{-J/2}}^{i_{+J/2}} (x_i - \bar{x}_J)(y_i - \bar{y}_J) \tag{17}$$

$$V_J = \sum_{j=i_{-J/2}}^{i_{+J/2}} (x_i - \bar{x}_J)^2 \tag{18}$$

Since each window of the local regression is formed by adding and removing a single observation, the results can be updated at each point of the interior space $J/2, ..., n-J/2$. When adding an observation, we calculate:

$$\bar{x}_{J+1} = (J\bar{x}_J + x_{J+1})/(J+1) \tag{19}$$

$$\bar{y}_{J+1} = (J\bar{y}_J + y_{J+1})/(J+1) \tag{20}$$

$$C_{J+1} = C_J + \frac{J+1}{J}(x_{J+1} - \bar{x}_{J+1})(y_{J+1} - \bar{y}_{J+1}) \tag{21}$$

$$V_{J+1} = V_J + \frac{J+1}{J}(x_{J+1} - \bar{x}_{J+1})^2 \tag{22}$$

When removing an observation, we calculate:

$$\bar{x}_{J-1} = ((J+1)\bar{x}_J - x_{J-1})/J \tag{23}$$

$$\bar{y}_{J-1} = ((J+1)\bar{y}_J - y_{J-1})/J \tag{24}$$

$$C_{J-1} = C_J - \frac{J}{J+1}(x_{J-1} - \bar{x}_{J-1})(y_{J-1} - \bar{y}_{J-1}) \tag{25}$$

$$V_{J-1} = V_J - \frac{J}{J+1}(x_{J-1} - \bar{x}_{J-1})^2 \tag{26}$$

Using the `algorithm(wfit)` option results in fitting least squares models at each point instead of updating the results from the previous fit. While this can be considerably slower for larger samples, this method can be expected to be more numerically stable. The (weighted) least squares estimator of $\hat{\alpha}$ and $\hat{\beta}$ from (4) is:

$$\hat{b} = (X'WX)^{-1}X'WY \tag{27}$$

where Y is an $n \times 1$ vector of response variables, X is a $n \times 2$ vector of predictor variables and a constant, and W is a $n \times n$ diagonal weight matrix. $\hat{b}$ is the resulting 2×1 coefficient vector.

In summary, to estimate $f(X)$ using Friedman's super smoother, we first estimate (4) over a grid of bandwidths J, with $0 < J < n$. For each of these initial smooths we calculate (10) and smooth the result of (10) against x_i using (4) with $J = 0.2n$. The resulting smooths are used to estimate (11), i.e., the optimal bandwidths. The optimal bandwidths are then smoothed again using (4) with $J = 0.2n$ and the result of this smooth is used for linearly interpolating between the two initial estimates with closest bandwidth values at each point x_i, subject to (12). Let $y^*_{i_1}$ and $y^*_{i_2}$ be the two smoothed points at x_i with closest bandwidths, then the penultimate estimate is obtained by

$$y_i^* = (y_{i_1}^* - y_{i_2}^*)/(J_{i_1} - J_{i_2})(J_{cv}^*(x_i) - J_{i_2}) + y_{i_2}^* \tag{28}$$

where J_{i_k} are the bandwidths that correspond to the initial estimates $y_{i_k}^*$. The final estimate is then obtained by smoothing y_i^* against x_i using (4) with bandwidth $J = 0.05n$.

Oversmoothing

Friedman (Friedman [1984]) proposed an oversmoothing parameter which biases the smooth towards the largest bandwidth from the grid of bandwidths over which the cross-validation was performed. The parameter is defined in the range [0,10] where zero corresponds to no oversmoothing, and 10 to the maximum oversmooth resulting in a fixed bandwidth estimator with $J = J_{max}$, where J_{max} is the largest bandwidth from the specified grid. Oversmoothing is applied to the results of (11), the optimal bandwidths $J_{cv}(x_i)$, prior to smoothing them:

$$J(x_i) = J_{cv}(x_i) + (J_{max} - J_{cv}(x_i))R_i^{10-\alpha} \tag{29}$$

where α is the parameter that can be specified in the `alpha(#)` option of the `supsmooth` command, and

$$R_i = \left[\frac{\hat{e}(J_{cv}(x_i)|x_i)}{\hat{e}(J_{max}|x_i)}\right] \tag{30}$$

Local weighting

Fitting least squares models locally using the `wfit` algorithm also allows for using locally varying weights, effectively allowing for locally weighted linear regression smoothing with adaptive bandwidths. The local tricube weights are:

$$w_j = \left[1 - \left(\frac{|x_j - xi|}{\Delta}\right)^3\right]^3 \tag{31}$$

where $\Delta = 1.001max(x_{i_{+J/2}} - x_i, x_i - x_{i_{-J/2}})$.

3 Reference

Friedman, J. H. 1984. A variable span smoother. *Laboratory for Computational Statistics*, Department of Statistics, Stanford University: Technical Report(5).