Ch 3 Local Polynomial Regression

Friday, March 1, 2019 12:30 AM

A) A natural generalization of locally constant regression is local polynomial regression. For points u in a neighborhood of the target point x, define the polynomial

$$g_x(u:a) = a_0 + \sum_{k=1}^{D} a_j (u-x)^k$$

for some vector of coefficients $a = (a_0, \ldots, a_D)$. We will estimate the coefficients a in $g_x(u;a)$ at some target point x using weighted least squares.

$$\hat{a} = \underset{R^{D+1}}{\arg\min} \sum_{i=1}^{n} w_i \{ y_i - g_x(x_i; a) \}^2$$

where $w_i \equiv w(x_i, x)$ are kernel weights normalized to sum 1. Derive a matrix form of $\hat{a}$, and by extension, the local function estimate $\hat{f}(x)$ at the target value x. Define R_x whose (i,j) entry is $(x_i - x)^{j-1}$ and remember weighted polynomial regression is the same thing as weighted linear regression with a polynomial basis.

What we are trying to do is approximate the true function $f(x)$ by a series of local polynomials $g_x(u:a) = a_0 + \sum_{i=1}^{D} (u-x)^j$ where x is the target point where we want to estimate $f(x)$ at and u is a neighbohood about x. To estimate the values of $a_0, \ldots, a_D$, we minimize the weighted residual sum of squares

$$\sum_{i=1}^{n} w_i \{ y_i - g_x(x_i, a) \}^2$$

To do this minimization, I will first define a matrix R and a vector a:

$$R = \begin{bmatrix} 1 & (x_i - x) & (x_i - x)^2 \cdots & (x_i - x)^D \\ 1 & \vdots & \vdots & \vdots \\ \vdots & & & \\ 1 & (x_n - x) & (x_n - x)^2 \cdots & (x_n - x)^D \end{bmatrix}_{n \times D} \quad a = \begin{bmatrix} a_0 \\ \vdots \\ a_D \end{bmatrix}_{D \times 1}$$

Thus, we can rewrite the polynomial g as $g_x(x;a) = Ra$. Thus, $\hat{a}$ becomes

$$\hat{a} = \underset{R^{D+1}}{\arg\min} (y - Ra)^T w (y - Ra) \quad \text{where } w \text{ is } \operatorname{diag}(w_1, \ldots, w_n) \text{ and } w_i = \frac{1}{h} K\left(\frac{x_i - x}{h}\right)$$

$$\frac{\partial}{\partial a} (y - Ra)^T w (y - Ra) = \frac{\partial}{\partial a} (y^T w - a^T R^T w)(y - Ra) = 0$$

$$= \frac{\partial}{\partial a} (y^T w y - y^T w R a - a^T R^T w y + a^T R^T w R a = 0$$

$$= 0 - 2R^T w y + 2R^T w R a = 0$$

$$\Rightarrow \hat{a} = (R^T w R)^{-1} R^T w y$$

Because $g(u, \hat{a})$ is the estimate of $f(u)$, at x, $g(u, \hat{a})$ simplifies to just the intercept $\hat{a}_0$. Since the estimate of the intercept depends on x, we can write this as $a_0(x)$. Another way to express this is by saying

$$\hat{f}(x) = \sum_{i=1}^{n} l_i y_i \quad \text{where } l = e_1^T (R^T w R)^{-1} R^T w \quad e_1 = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

$$l = (l_1, \ldots, l_n)$$

* note: $\hat{f}(x)$ is a single point estimate at x.

B) From this, conclude that for the special case of the local linear estimator $(D=1)$, we can write $\hat{f}(x)$ as a linear smoother of the form

$$\hat{f}(x) = \frac{\sum_{i=1}^{n} w_i(x) y_i}{\sum_{i=1}^{n} w_i(x)}$$

where the unnormalized weights are

$$w_i(x) = K\left(\frac{x - x_i}{h}\right) \{ s_2(x) - (x_i - x) s_1(x) \}$$

$$s_j(x) = \sum_{i=1}^{n} K\left(\frac{x - x_i}{h}\right)(x_i - x)^j$$

For this case, we have that $R = \begin{bmatrix} 1 & (x_1 - x) \\ \vdots & \vdots \\ 1 & (x_n - x) \end{bmatrix}$

Therefore, to obtain the l_i from the previous portion, I can calculate

$$l_i = \underset{1 \times 2}{e_1^T} \underset{2 \times n}{(R^T w R)^{-1}_{2 \times 2} R^T w_{2n}}$$

$$R^T w R = \begin{bmatrix} 1 & \cdots & \cdots & 1 \\ (x_1 - x) & \cdots & & (x_n - x) \end{bmatrix} \begin{bmatrix} w_1 & & 0 \\ & \ddots & \\ 0 & & w_n \end{bmatrix} \begin{bmatrix} 1 & (x_1 - x) \\ \vdots & \vdots \\ 1 & (x_n - x) \end{bmatrix}$$

$$= \begin{bmatrix} w_1 & \cdots & w_n \\ w_1(x_1 - x) & \cdots & w_n(x_n - x) \end{bmatrix} \begin{bmatrix} 1 & (x_1 - x) \\ \vdots & \vdots \\ 1 & (x_n - x) \end{bmatrix}$$

$$= \begin{bmatrix} \sum_{i=1}^{n} w_i & \sum_{i=1}^{n} w_i (x_i - x) \\ \sum_{i=1}^{n} w_i (x_i - x) & \sum_{i=1}^{n} w_i (x_i - x)^2 \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{n} w_i & s_1 \\ s_1 & s_2 \end{bmatrix}$$

$$(R^T w R)^{-1} = \frac{1}{\det(R^T w R)} \begin{bmatrix} s_2 & -s_1 \\ -s_1 & \sum_{i=1}^{n} w_i \end{bmatrix}$$

$$l = e_1^T (R^T w R)^{-1} R^T w = \frac{1}{\det(R^T w R)} \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} s_2 & -s_1 \\ -s_1 & \sum_{i=1}^{n} w_i \end{bmatrix} \begin{bmatrix} w_1 & \cdots & w_n \\ w_1(x_1 - x) & \cdots & w_n(x_n - x) \end{bmatrix}$$

$$= \frac{1}{s_1^2 - s_2 \sum_{i=1}^{n} w_i} \begin{bmatrix} s_2 & -s_1 \end{bmatrix} \begin{bmatrix} w_1 & \cdots & w_n \\ w_1(x_1 - x) & \cdots & w_n(x_n - x) \end{bmatrix}$$

$$= \frac{1}{s_1^2 - s_2 \sum_{i=1}^{n} w_i} \begin{bmatrix} s_2 w_1 - s_1 w_1 (x_1 - x) & \cdots & s_2 w_n - s_1 w_1 (x_n - x) \end{bmatrix}$$

We also have $\sum_{i=1}^{n} s_2 w_i - s_1 w_i (x_i - x) = s_2 \sum_{i=1}^{n} w_i - s_1 \sum_{i=1}^{n} w_i (x_i - x)$

$\lfloor w_1(x_1-x) \cdots w_n(x_n-x) \rfloor$

$$= \frac{1}{S_1^2 - S_2\sum_{i=1}^n w_i}\left[S_2 w_1 - S_1 w_1(x_1-x) \ \ldots \ S_2 w_n - S_1 w_1(x_n-x)\right]$$

We also have $\sum_{i=1}^n S_2 w_i - S_1 w_i (x_i - x) = S_2 \sum_{i=1}^n w_i - S_1 \sum_{i=1}^n w_i(x_i - x)$

$$= S_2 \sum_{i=1}^n w_i - S_1^2$$

Putting it all together, we have $\hat{f}(x) = \dfrac{\sum_{i=1}^n l_i y_i}{\sum_{i=1}^n l_i}$ where $l_i = \dfrac{K\left(\frac{x-x_i}{h}\right)\{S_2(x) - (x_i - x)S_1(x)\}}{S_2\sum_{i=1}^n w_i - S_1^2}$

I chose to change the notation for this problem because I do not like using w_i to denote both the kernel weights and the linear weights.

C) Suppose that the residuals have constant variance σ^2 (that is the $\overset{\text{is this the errors?}}{\text{residual}}$ does not depend on x) Derive the mean and variance of the sampling distribution for the local polynomial estimate $\hat{f}(x)$ at some arbitrary point x. Note: the random variable $\hat{f}(x)$ is just a scalar quantity at x, not the whole function.

We have $\hat{f}(x) = \sum_{i=1}^n l_i y_i$ where $l_i = e_1^T (R^T w R)^{-1} R^T W$ $\quad e_1 = \begin{bmatrix}1\\0\\\vdots\\0\end{bmatrix}$. Therefore, the mean of $\hat{f}(x)$ is

$$E(\hat{f}(x)) = E\left[\sum_{i=1}^n l_i y_i\right]$$

$$= E\left[e_1^T (R^T w R)^{-1} R^T w) y\right]$$

$$= e_1^T (R^T w R)^{-1} R^T w (f(\underset{\sim}{x}))$$ where $\underset{\sim}{x} = (x_1, \ldots, x_n)$ since $y = f(\underset{\sim}{x}) + \varepsilon$ where $E(\varepsilon) = 0$

$$= \sum_{i=1}^n l_i f(x_i)$$

$$Var(\hat{f}(x)) = Var(e_1^T (R^T w R)^{-1} R^T w y)$$

$$= e_1^T (R^T w R)^{-1} R^T w \ Var(y) \ w^T R (R^T w R)^{-1} e_1$$

$$= \sigma^2 e_1^T (R^T w R^{-1}) R^T w I w^T R (R^T w R)^{-1} e_1$$

$$= \sigma^2 \|l_i\|_2^2$$

$\hat{f}(x) \sim N(l_i^T f(x_i), \sigma^2 \|l_i\|_2^2)$, so the bias comes in because the mean of $\hat{f}(x)$ is not $f(x)$ but rather, $f(x)$ with a smoother applied to it.

D) To estimate σ^2, note that if x is a random vector with mean μ and covariance matrix Σ, then for any symmetric matrix Q of appropriate dimension, then $x^T Q x$ has expectation

$$E(x^T Q x) = tr(Q\Sigma) + \mu^T Q \mu$$

Write the vector of residuals as $r = y - \hat{y} = y - Hy$, where H is the smoothing (or hat) matrix. Compute the expected value of the estimator

$$\hat{\sigma}^2 = \frac{\|r\|_2^2}{n - 2tr(H) + tr(H^T H)}$$

and simplify things as much as possible. Roughly under what circumstances will this estimator be nearly unbiased for large n?

Let $r = y - \hat{y} = y - Hy$.

$$E(\hat{\sigma}^2) = E\left[\frac{\|r\|_2^2}{n - 2tr(H) + tr(H^T H)}\right]$$

$$= \underbrace{\frac{1}{n - 2Tr(H) + tr(H^T H)}}_{\text{call this } C} E\left[\|r\|_2^2\right]$$

$$= C\, E[(y - Hy)^T (y - Hy)]$$

$$= C\, E[(y^T - y^T H^T)(y - Hy)]$$

$$= C\, E[y^T y - y^T H y - y^T H^T y + y^T H^T H y]$$

$$= C\left[E[y^T y] - 2E[tr(y^T H y)] + E[y^T H^T H y]\right]$$

$$= C\left[E\left[\sum_{i=1}^n y_i^2\right] - 2E[tr(Hyy^T)] + E[tr(H^T H y y^T)]\right]$$

$$= C\left[\sum_{i=1}^n (\sigma^2 + f(x_i)^2) - 2Tr(E(Hyy^T)) + Tr(E(H^T H yy^T))\right]$$

$$= C\left[n\sigma^2 + \sum_{i=1}^n f(x_i)^2 - 2Tr(H \underbrace{E(yy^T)}_{\substack{= Cov(y) + E^2(y) \\ = \sigma^2 I + f(\underset{\sim}{x})^T f(\underset{\sim}{x})}}) + Tr(H^T H E(yy^T))\right]$$

$$= C\left[n\sigma^2 + f(\underset{\sim}{x})^T f(\underset{\sim}{x}) - 2Tr(H(\sigma^2 I + f(\underset{\sim}{x})^T f(\underset{\sim}{x})) + Tr(H^T H(\sigma^2 I + f(\underset{\sim}{x})^T f(\underset{\sim}{x})))\right]$$

$$= C\left[n\sigma^2 + f(\underset{\sim}{x})^T I f(\underset{\sim}{x}) - 2\sigma^2 Tr(H) - 2Tr(Hf(\underset{\sim}{x})^T f(\underset{\sim}{x})) + \sigma^2 Tr(H^T H) + Tr(H^T H f(\underset{\sim}{x})^T f(\underset{\sim}{x}))\right]$$

$$= C\left[\sigma^2(n - 2Tr(H) + Tr(H^T H)) + f(\underset{\sim}{x})^T I f(\underset{\sim}{x}) - 2f(x)^T H f(x) + f(x)^T H^T H f(x)\right] \qquad Tr(A^T B) = Tr(AB^T)$$

$$= C\left[\sigma^2(n - 2Tr(H) + Tr(H^T H)) + (f(x) - Hf(x))^T (f(x) - Hf(x))\right]$$

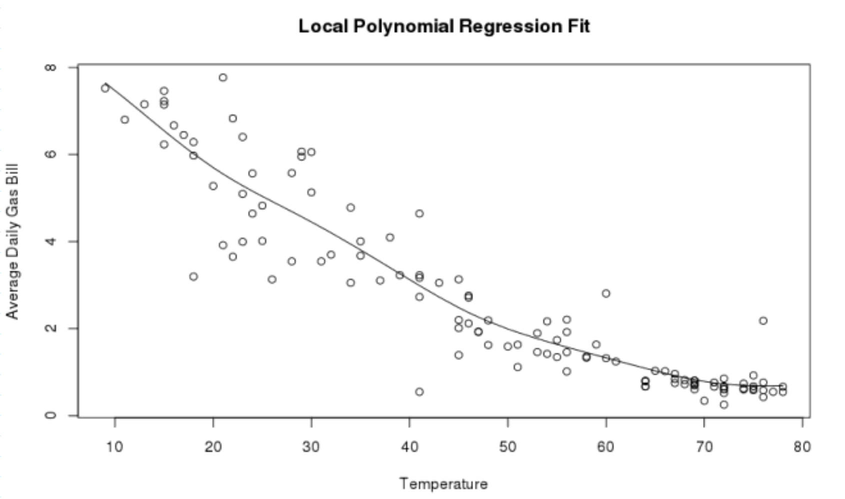
$$= \frac{\sigma^2 \cancel{(n - 2Tr(H) + Tr(H^T H))}}{\cancel{(n - 2Tr(H) + Tr(H^T H))}} + \frac{(f(x) - Hf(x))^T (f(x) - Hf(x))}{(n - 2Tr(H) + Tr(H^T H))}$$

$$= \sigma^2 + \frac{(f(x) - Hf(x))^T (f(x) - Hf(x))}{\underbrace{(n - 2Tr(H) + Tr(H^T H))}_{\text{effective degrees of freedom}}}$$

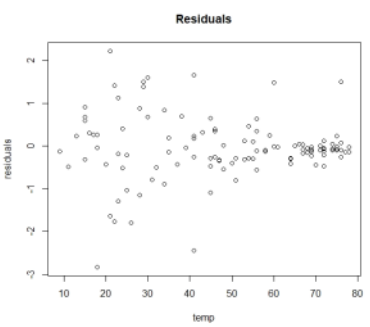
This estimator $\hat{\sigma}^2$ will be unbiased if $f(x)$ is close to $Hf(x)$. This can be interpreted as the true function $f(x)$ is close to its projection onto the polynomial basis. That is, if $f(x)$ is close to its polynomial estimate. Also, if n is large, ie we have a large sample, the bias will also be smaller. Therefore, the bias is smaller if we don't do a lot of smoothing, or we have a large sample.

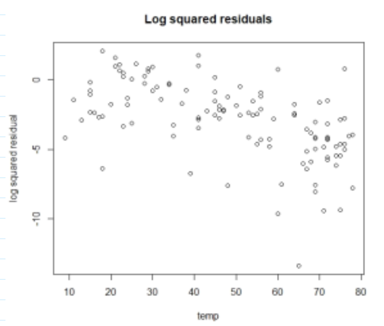
E) Fit a local linear estimator using a Gaussian Kernel. Pick the bandwidth h using leave one out cross validation. Let Y be the average daily gas bill and X be the temperature.

The h value my LOOCV chose was 6.88. Below is the resulting plot

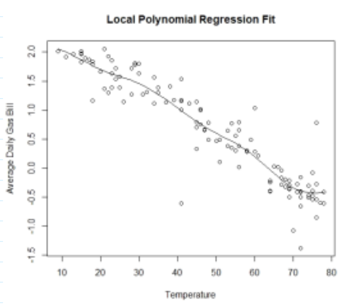


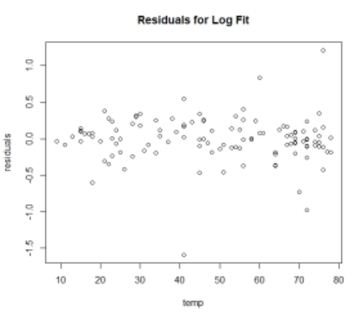
F) Inspect the residuals from the fitted model. Does the assumption of constant variance (homoscedasticity) look reasonable? If not, do you have any suggestions for fixing it?





The residuals show clear signs of heteroskedascity. To remedy the heteroskedascity, there are several options. One option is to take the log of the y variable and refit the model using the transformed response. Any fitted values would be exponentiated to reverse the transformation.





Looking at the new residuals, we see that there is not as much heteroskedascity, though the residuals do still look a little tighter towards the left than towards the right.

Another option we can take is to fit a regression on the log squared residuals and use the fitted residuals to transform our original model into a homoskedastic model. The theory is:

$$Var(\varepsilon_i) = \sigma_i^2 = \sigma^2 x_i^{\gamma} \quad \text{where } \gamma \text{ is some unknown parameter}$$

$$\begin{aligned} \ln(\sigma_i^2) &= \ln(\sigma^2) + \gamma \ln(x_i) \\ &= \alpha_1 + \alpha_2 z_i \end{aligned}$$

To perform this regression, we can use the residuals squared as the observed σ_i^2 values:

$$\ln(\hat{\varepsilon}_i) = \ln(\sigma_i^2) + \underset{\text{some new error}}{v_i} = \alpha_1 + \alpha_2 z_i + v_i$$

The log squared residuals have a vague linear looking relationship so we can find $\hat{\alpha}_1$ and $\hat{\alpha}_2$ via OLS. After that, we can obtain fitted values for σ_i^2, $\hat{\sigma}_i^2$.

$$\hat{\sigma}_i^2 = \exp(\hat{\alpha}_1 + \hat{\alpha}_2 z_i)$$

From here, we can use $\hat{\sigma}_i^2$ to reweight our original regression problem.

$$y_i^* = \frac{y_i}{\hat{\sigma}_i} = \frac{f(x_i)}{\hat{\sigma}_i} + \frac{\varepsilon_i}{\hat{\sigma}_i}$$
$$= f^*(x_i) + \varepsilon_i^*$$

Now, ε_i^* should have variance 1 since $Var(\frac{\varepsilon_i}{\sigma}) = \frac{1}{\sigma_i^2}\sigma_i^2 = 1$. I'll try to implement this later.

G) Put everything together to construct an approximate pointwise 95% confidence interval for the local linear model. Plot the confidence bands together with the estimated function on a scatterplot of the data.

To make the 95% confidence interval, I calculated $\hat{\sigma}^2$ according to part D. Then for each point, I calculated a 95% confidence interval

$$\left[\hat{f}(x_i) - 1.96\sqrt{\hat{\sigma}^2},\ \hat{f}(x_i) + 1.96\sqrt{\hat{\sigma}^2}\right]$$

Below is the plot.