

# Local Quantile Smoothing in Locfit

Catherine Loader  
The University of Auckland

October 29, 2006

Locfit 2.0 contains an algorithm for local quantile smoothing. That is, we fit a local regression model,

$$Y_i = \mu(x_i) + \epsilon_i.$$

The usual local regression assumption is  $E(\epsilon_i) = 0$ , or equivalently,  $E(Y_i) = \mu(x_i)$ . For a local quantile smoother, this assumption is replaced by

$$\begin{aligned} P(\epsilon_i < 0) &= p \\ P(\epsilon_i \geq 0) &= 1 - p \end{aligned}$$

where  $p$  is a specified quantile. If  $p = 0.5$ , then the smoother is a local median. If  $p = 0$  or  $p = 1$ , the smoother is a local minimum or local maximum, respectively.

## 1 Commands

Local quantile smoothing is specified with `family=quant()`. A typical call is

```
> fit <- locfit(NOx~E,family=quant(0.5),data="ethanol")
> plot(fit)
```

for local median smoothing. The argument to `quant()` gives the required quantile  $p$ ; the default  $p = 0.5$  corresponds to a running median.

Note that the local quantile smooth is, in general, discontinuous, even when all components are nicely behaved. The plot command above uses locfit's standard interpolation from a coarse set of points, so a smoother curve results.

## 2 Theory

Formally, the quantile smoother uses local likelihood estimation, based on the family of densities

$$f_p(y; \mu) = \begin{cases} e^{-(y-\mu)/(1-p)} & y > \mu \\ e^{(y-\mu)/p} & y \leq \mu \end{cases}.$$

When  $p = 1/2$ , this is the familiar Laplace (or double exponential) density,  $f_p(y; \mu) = e^{-|y-\mu|/2}$ . For simplicity, I'll write the density with general  $p$  as

$$f_p(y; \mu) = e^{-|y-\mu|_p}$$

where  $|v|_p$  is defined as

$$|v|_p = \begin{cases} \frac{v}{1-p} & v > 0 \\ \frac{-v}{p} & v \leq 0 \end{cases} .$$

**Theorem 1** Suppose  $Y_1, \dots, Y_n$  are an i.i.d. sample from  $f_\mu(y)$ , and let  $\hat{F}(y)$  denote the empirical distribution function. Then any maximum likelihood estimate (MLE)  $\hat{\mu}$  of  $\mu$  must satisfy

$$\hat{F}(y) \begin{cases} \leq p & \text{for } y < \hat{\mu} \\ \geq p & \text{for } y > \hat{\mu} \end{cases} .$$

In plain english, this theorem says that  $\hat{\mu}$  is the  $p$ th sample quantile -  $\hat{F}^{-1}(p)$ , if you consider  $\hat{F}$  to be a continuous staircase, rather than a step function.

**Proof:** The log-likelihood function is

$$l(\mu) = - \sum_{i=1}^n |Y_i - \mu|_p,$$

and any maximizer  $\hat{\mu}$  is an MLE. The (right-continuous) derivative is

$$\begin{aligned} \dot{l}(\mu) &= -\frac{1}{p} \sum I(Y_i \leq \mu) + \frac{1}{1-p} \sum I(Y_i > \mu) \\ &= -\frac{n}{p} \hat{F}(\mu) + \frac{n}{1-p} (1 - \hat{F}(\mu)) \\ &= \frac{n(p - \hat{F}(\mu))}{p(1-p)} . \end{aligned}$$

Therefore, the log-likelihood  $l(\mu)$  is increasing whenever  $\hat{F}(\mu) < p$ , and decreasing whenever  $\hat{F}(\mu) > p$ .

This result motivates the use of  $f_\mu(y)$  in finding the local quantile estimator. Formally, define the local log-likelihood as

$$\begin{aligned} \mathcal{L}_x(a) &= \sum_{i=1}^n w_i(x) \log f(y; \mu_i) \\ &= \sum_{i=1}^n w_i(x) |Y_i - \mu_i|_p, \end{aligned}$$

with  $\mu_i = \langle a, A(x_i - x) \rangle$  (see Chapter 4 of my book for the notation). Then the local quantile estimate is

$$\hat{\mu} = \hat{a}_0,$$

where  $\hat{a}_0$  is the first component of the local MLE  $\hat{a}_x$ .

### 3 Computational Algorithm

We want to find  $\hat{a} = \hat{a}_x$  to minimize

$$\mathcal{L}_x(a) = \sum_{i=1}^n w_i(x) |Y_i - \langle a, A(x_i - x) \rangle|_p.$$

As this is piecewise linear, rather than continuously differentiable, Newton-Raphson and similar algorithms cannot be employed. The basic optimization strategy is as follows:

1. Begin with an initial value  $a_1$ .
2. At the  $j$ th step, choose a ‘move’ direction  $\rho_j$ .
3. Find  $\delta$  to minimize  $\mathcal{L}_x(a_j + \delta\rho_j)$ .
4. Set  $a_{j+1} = a_j + \delta\rho_j$ .
5. Repeat 2, 3 and 4 until convergence.

The two main problems are to choose the move directions  $\rho_j$ , and distance  $\delta$ . The second problem is easier, so we solve that first.

### 3.1 Choosing distance $\delta$

The problem is to minimize

$$S(\delta) = \sum_{i=1}^n w_i |e_i - \delta\lambda_i|_p$$

where  $e_i = Y_i - \langle a, A(x_i - x) \rangle$  is the residual from the current parameter, and  $\lambda_i = \langle \rho, A(x_i - x) \rangle$ .

Split the sum as

$$S(\delta) = \sum_{\lambda_i > 0} w_i \lambda_i \left| \frac{e_i}{\lambda_i} - \delta \right|_p + \sum_{\lambda_i < 0} w_i (-\lambda_i) \left| \frac{e_i}{\lambda_i} - \delta \right|_{1-p} + \sum_{\lambda_i = 0} w_i |e_i|.$$

To differentiate, these sums are further split into  $e_i/\lambda_i < \delta$  and  $e_i/\lambda_i > \delta$ . Note that  $|v|_p$  has slope  $-1/p$  for  $v < 0$  and  $1/(1-p)$  for  $v > 0$ . The derivative (at differentiable points) is therefore

$$S'(\delta) = - \sum_{\lambda_i > 0, e_i/\lambda_i > \delta} \frac{w_i \lambda_i}{1-p} + \sum_{\lambda_i > 0, e_i/\lambda_i < \delta} \frac{w_i \lambda_i}{p} \quad (1)$$

$$\begin{aligned} & - \sum_{\lambda_i < 0, e_i/\lambda_i > \delta} \frac{w_i (-\lambda_i)}{p} + \sum_{\lambda_i < 0, e_i/\lambda_i < \delta} \frac{w_i (-\lambda_i)}{1-p} \\ & = - \sum_{e_i/\lambda_i > \delta} w_i |\lambda_i|_p + \sum_{e_i/\lambda_i < \delta} w_i |\lambda_i|_{1-p}. \end{aligned} \quad (2)$$

As  $\delta$  increases, indices  $i$  move from the first sum (negative) to the second sum (positive). We want to find the value of  $\delta$  where the overall derivative changes sign. This can be achieved using modifications of standard ranking algorithms, for example,

1. Choose a pivot,  $\delta_0 = e_i/\lambda_i$  for some arbitrary  $i$ .
2. Sort observations into  $\{i : e_i/\lambda_i < \delta_0\}$ ;  $\{i : e_i/\lambda_i = \delta_0\}$ ; and  $\{i : e_i/\lambda_i > \delta_0\}$ .
3. Compute the sum (2), first with the ‘=’ group included in the first sum, then with the ‘=’ group included in the second sum.
4. If these have different signs, then we’re done. If both are positive, then  $\delta_0$  is too large - choose another pivot from the low group. If both are negative, choose another pivot from the high group.

Of course, there’s non-uniqueness if (2) is ever exactly equal to 0. We deal with this problem later.

### 3.2 Choosing direction $\rho_j$

We suppose that the  $n \times p$  design matrix  $\mathbf{X}$  has linearly independent columns, even when restricted to rows with  $w_i(x) > 0$ . In this case, the fitted local polynomial will be constrained pass through  $p$  observations, in a linearly independent configuration. The main idea in determining the move directions is to cycle through a set of  $p$  linear constraints on the coefficients. In turn, each constraint is removed, and we choose  $\rho_j$  orthogonal to the remaining  $p - 1$  constraints.

Then, choose  $\delta$  as described above. The chosen  $\delta$  will equal  $e_i/\lambda_i$  for some  $i$  - the corresponding row  $X_i$  of the design matrix is added as the new constraint.

### 3.3 Non-uniqueness

There are two ways in which non-uniqueness can influence the algorithm described here. First, in choosing  $\delta$ , the sum (2) may be exactly 0, resulting in  $\delta$  not being fully defined. This corresponds to the non-uniqueness of the median for  $n$  even. However, proper resolution is critical here - making the wrong choice may lead to the algorithm stopping short of the true optimum.

The second non-uniqueness is in the choice of new constraint, when two (or more) observations, with different design points, have the same critical value of  $e_i/\lambda_i$ . This corresponds, for example, to fitting a straight line to data, when three (or more) points lie on a line. Which points should be used as constraints? If the line is almost, but not quite, optimal, then choosing the wrong points may again result in the algorithm getting stuck before reaching the true optimum.

In either case, we must try to avoid getting stuck with the wrong constraints. Locfit uses a scheme based on cycling through the set of possible points in an attempt to solve this.