Computational Aspects of Automatic Model Selection in Local Polynomial Regression

Vadim Kutsyy
Department of Statistics
The University of Chicago
5734 University Avenue, Chicago, IL 60637

May, 1996

ABSTRACT

In this paper I am interested in investigating automatic model selection in local polynomial fitting, in particular the method of bandwidth and order selection introduced by Fan and Gijbels. I will discuss computational aspects of these methods. Finally I will introduce a method of automatically selecting the quantity and location of a set of local polynomials.

1 INTRODUCTION

Smoothing techniques are often used in nonparametric regression as a powerful method for finding regression curves from a set of points. There is a wide selection of such methods (for example see Hastie and Tibshirani 1990).

Common regression smoothers are the Nadaraya-Watson and Gasser-Müller kernel estimators (Gasser and Müller 1979, 1984: Nadaraya 1964; Watson 1964), smoothing splines (Reinsch 1967; Wahba 1990), and local polynomial (see Müller 1988). All of them use the idea of kernel smoothers. Fitting local polynomial smoothers, which includes local linear smoothers as a special case of polynomials with order 1, has a number of advantages. For example, the estimator achieves full minimax efficiency (using an Epanechnikov kernel (Fan 1993)) among linear estimators, and adapts automatically to the boundary (Fan and Gijbels 1992). However, many practical issues of the fitting of local polynomials remain to be studied.

The rest of the paper is organized as follows. Section 2 is used to introduce local polynomial regression. In Section 3 I will explore methods of automatic bandwidth and order selection suggested by Fan and Gijbels, while in Section 4 I consider some computational issues. In Section 5 I introduce a new method for automatically choosing the number and location of local polynomial regression center points. Finally, in Section 6 I make a few considering remarks.

2 LOCAL POLYNOMIAL REGRESSION

Consider bivariate data $(X_1,Y_1),...(X_n,Y_n)$ (similar results can be achieve for higher dimensions), which is a random sample from the population (X,Y) whose relationship can be represented as:

$$Y = m(X) + \sigma(X)\varepsilon \qquad E(\varepsilon) = 0, \qquad var(\varepsilon) = 1, \tag{1}$$

where X and ε are independent. We are interested in estimating the regression function $m(x) = E(Y \mid X = x)$ and its derivatives. We can approximate the function $m(x)$, using a Taylor series, by a polynomial of degree p for x in a neighborhood of x_0:

$$m(x) \approx m(x_0) + m^{(1)}(x_0)(x - x_0) + ... + m^{(p)}(x_0)(x - x_0)^p/p! \ . \tag{2}$$

Let h be a bandwidth which controls the size of the neighborhood and be kernel function (symmetric probability density function). Then we can carry out weighted polynomial regression by minimizing

$$\sum_{i=1}^{n} \left(Y_i - \sum_{j=0}^{p} \beta_j (X_i - x_0)^j \right)^2 K\left(\frac{X_i - x_0}{h} \right) \tag{3}$$

where $\beta_j = m^{(j)}(x_0)/j!$. This can be written in the matrix form:

$$(\mathbf{Y} - \mathbf{X}\beta)^t \mathbf{W} (\mathbf{Y} - \mathbf{X}\beta), \tag{4}$$

where $\mathbf{W}$ is an $n \times n$ diagonal matrix with $\mathbf{W}_{i,i} = K((X_i - x_0)/h)$, $(Y) = (Y_1, Y_2..., Y_n)^t$, $\beta = (\beta_1, \beta_2..., \beta_n)^t$ and

$$\mathbf{X} = \begin{pmatrix} 1 & (X_1 - x_0) & \dots & (X_1 - x_0)^p \\ 1 & (X_2 - x_0) & \dots & (X_2 - x_0)^p \\ \vdots & \vdots & \ddots & \vdots \\ 1 & (X_n - x_0) & \dots & (X_n - x_0)^p \end{pmatrix} \ . \tag{5}$$

Then ordinary least square theory gives the solution:

$$\hat{\beta}(x_0) = (\mathbf{X}^t \mathbf{W} \mathbf{X})^{-1} \mathbf{X}^t \mathbf{W} \mathbf{Y} \tag{6}$$

whose conditional mean and variance are:

$$E(\hat{\beta}(x_0)|X_1,X_2,...,X_n) = (\mathbf{X}^t\mathbf{W}\mathbf{X})^{-1}\mathbf{X}^t\mathbf{W}\mathbf{m} = \beta + (\mathbf{X}^t\mathbf{W}\mathbf{X})^{-1}\mathbf{X}^t\mathbf{W}\mathbf{r}, \tag{7}$$

$$var(\hat{\beta}(x_0)|X_1,X_2,...,X_n) = (\mathbf{X}^t\mathbf{W}\mathbf{X})^{-1}(\mathbf{X}^t\blacksquare\mathbf{X})(\mathbf{X}^t\mathbf{W}\mathbf{X})^{-1} \tag{8}$$

where $\mathbf{m} = (m(X_1), m(X_2), ..., m(X_n))^t$, $\mathbf{r} = \mathbf{m} - \mathbf{X}\beta$, the residuals of the local polynomial, and $\blacksquare$ is an $n \times n$ covariance diagonal matrix with $\blacksquare_{i,i} = K^2(\frac{X_i - x_0}{h})\sigma^2(X_i)$.

In practice, before performing local polynomial regression, the number of local polynomials (n_{x_0}), the location of the center points (vector $\mathbf{x_0}$), the bandwidth (vector $\mathbf{h}$) and the order of the polynomials (vector $\mathbf{p}$), and the kernel function have to be specified. Usually, one may want to start with polynomials of order $p = 3$, some bandwidth h, a set of center points, and depending on the result, increase or decrease the bandwidth, and sometimes increase or decrease the order.

The main family of kernel functions used is the symmetric Beta family:

$$K(x) = \frac{\Gamma(2\alpha + \alpha)}{\Gamma(1+\alpha)^2 2^{2\alpha+1}}(1-x^2)_+^\alpha \tag{9}$$

where the subscript + means positive part (which is assumed to be taken before the exponential, so the function support is $[-1,1]$). This family includes the uniform kernel ($\alpha = 0$), the Epanechnikov kernel ($\alpha = 1$), the biweight kernel ($\alpha = 2$) and the triwieght kernel ($\alpha = 3$). Also, it includes the Gaussian kernel $K(x) = \phi(x)$, in the limit as $\alpha \to \infty$ (Marron and Nolan 1988).

3 AUTOMATIC MODEL SELECTION

3.1 Model selection criterion

The above theory of local polynomials works fine for the right choice of order p, and bandwidth h. In practice we do not know the right choice of p and h. Let us look at

regressions, when we use the wrong p and h. I will use a simulation of a Gaussian peak (Seifert and Gasser 1996) (Figure 1) as an example:

$$m(x) = 2 - 5x + 5e^{-\left(\frac{x-.5}{.05}\right)^2} \quad . \tag{10}$$

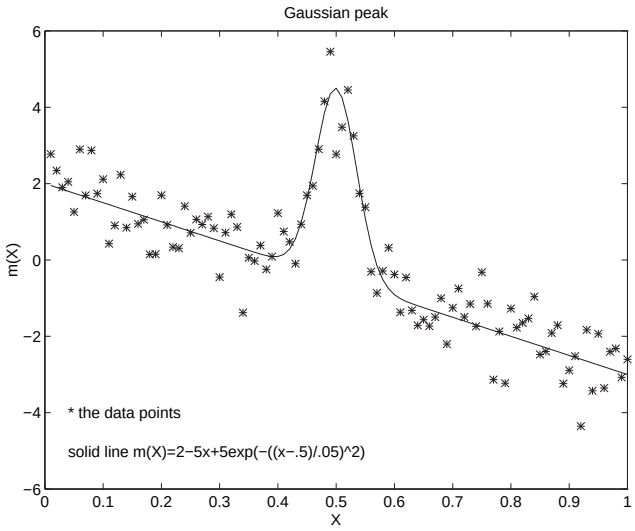

Figure 1: Plot of a Gaussian peak and the same Gaussian peak with added noise in the form of Normal with mean 0 and variance 0.5

Figure 1 shows $m(x)$ and a sample at 100 equally spaced points in the interval $[0,1]$ where normally distributed error with variance $\sigma^2 = 0.5$ was added.

First consider bandwidth selection. When we fit a local polynomial of a degree p with bandwidth h we have to specify the bandwidth. From Figure 2 it can be seen that small bandwidth gives a smaller bias, but higher variance, and larger bandwidth give lower variance, but larger bias. I have computed mean squared error, the lowest $\mathbf{MSE} = 8.5$ has regression with $h = 0.1$.

Now, consider the regression behavior for the different orders of polynomial. For these regressions I chose bandwidth $h = .1$. From Figure 3 it also can be seen that the lowest order of polynomial has the highest bias and the lowest variance. Here we have

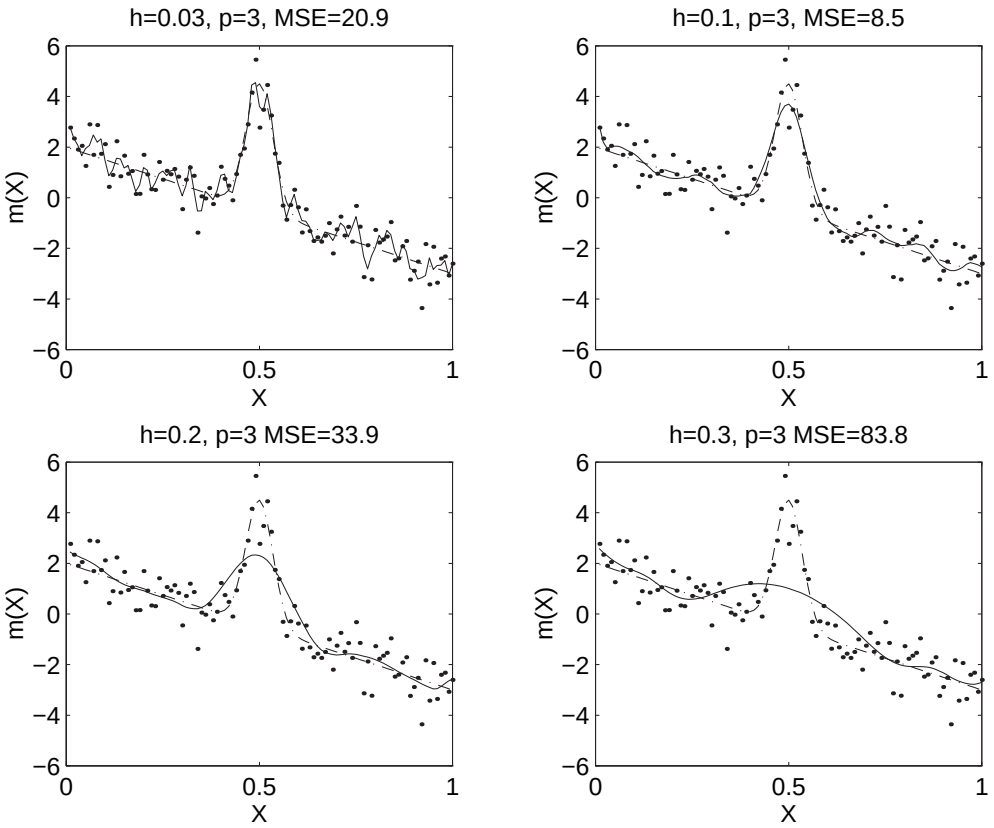

Figure 2: The local polynomial fit based on 100 equally spaced data points. The fit consist of 100 local equally spaced polynomial with bandwidth h given at each plot. The solid line is local polynomial fit, dash-doted line is Gaussian peak. (CPU time less than 4 seconds per regression)

a regression which is a little bit better than the one which we had before (for $p = 5$ and $h = 0.1$, $\mathbf{MSE} = 7.9$).

However, here I used information about the data to calculate **MSE** and chose the model. Usually we do not have that kind of information (otherwise there would be no reason to do estimation).

So, to find the optimal order and bandwidth we need to introduce some quantity which would represent the cost in bias-variance trade off. We will use both the usual MSE and a preliminary quantity called the residual squared criterion (RSC) of Fan and Gijbels (1995b) which estimates the local mean-squared error (MSE) (I will use bold font

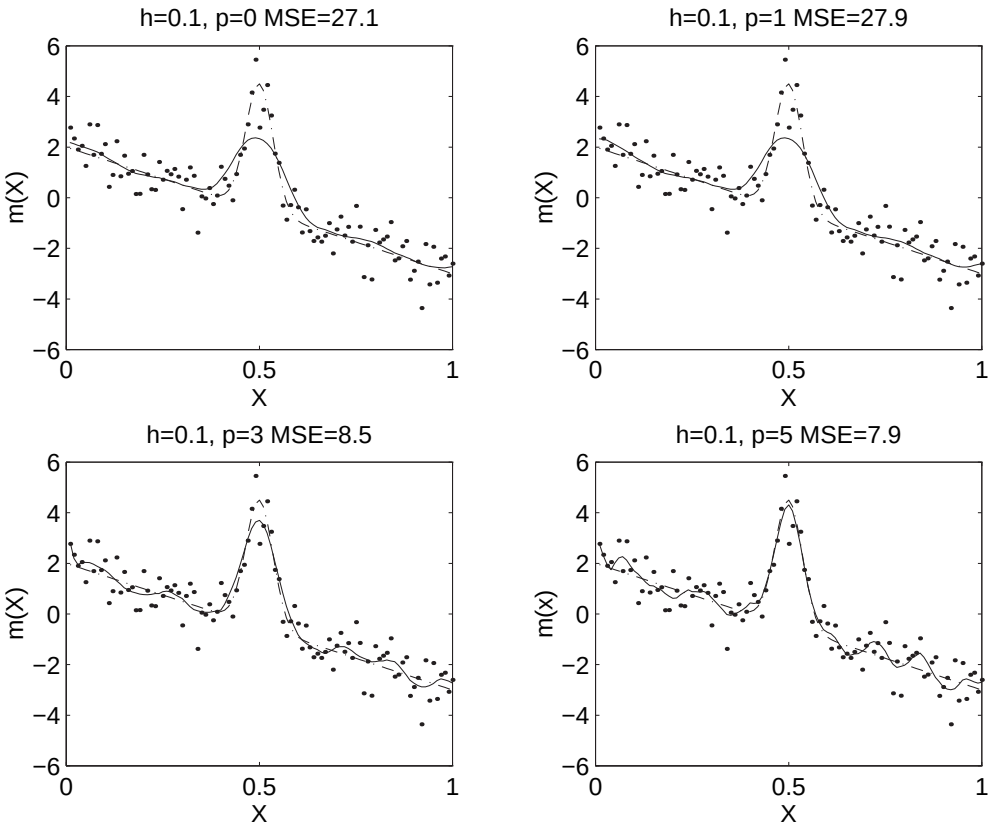

Figure 3: The local polynomial fit based on 100 equally spaced data points. The fit consist of 100 local equally spaced polynomial and the order of polynomial is given at each plot. The solid line is local polynomial fit, dash-doted line is Gaussian peak. (CPU time less than 4 seconds per regression)

for observed value of **MSE** and italic for *MSE* which we want to approximate). This latter quantity is based on the normalized weighted residual sum of squares:

$$\hat{\sigma}^2 = \frac{1}{tr(\mathbf{W}) - tr((\mathbf{X}^t\mathbf{W}\mathbf{X})^{-1}\mathbf{X}^t\mathbf{W}^2\mathbf{X})} \sum_{i+1}^{n} (Y_i - \hat{Y}_i)^2 K\left(\frac{X_i - x_0}{h}\right) \tag{11}$$

with $\hat{y} = (\hat{Y}_1, \hat{Y}_2, ..., \hat{Y}_n)^t = \mathbf{X}\beta$.

The *RSC* is *defined* as :

$$RSC(x_0; h) = \hat{\sigma}(x_0)(1 + (p+1)\mathbf{V}) \tag{12}$$

where $V = (\mathbf{S}^{-1}\mathbf{S}^*\mathbf{S}^{-1})_{1,1}$, $\mathbf{S} = \mathbf{X}^t\mathbf{W}\mathbf{X}$, and $\mathbf{S}^* = \mathbf{X}^t\mathbf{W}^2\mathbf{X}$. We can estimate Σ by $\sigma^2\mathbf{W}^2$,

(Fan and Gijbels 1995b), then the variance in (8) can be estimated as

$$var(\hat{\beta}|X_1, X_2, ..., X_n) \approx \sigma^2 \mathbf{S}^{-1}\mathbf{S}^*\mathbf{S}^{-1}$$

.

A few words about V. The formula for V is kind of complicated, but let us consider for simplicity the case where W is an $n \times n$ identity matrix, then

$$V^{-1} = ((\mathbf{X}^t\mathbf{X})^{-1}(\mathbf{X}^t\mathbf{X})(\mathbf{X}^t\mathbf{X})^{-1})^{-1}_{1,1} = (\mathbf{X}^t\mathbf{X})_{1,1} = \sum_{i=1}^{n} X_i^2 \ .$$

In general, V^{-1} measure the number of active local points, i.e. points which contribute to the regression.

The intuition behind (12) is as follows. If the bandwidth is too large or the order of polynomial is too small, the regression does not fit well, and the residual sum of squares $\hat{\sigma}^2(x_0)$ is large. If the bandwidth is too small or the order of polynomial is too large, the variance term V is large . The theoretical proof of this result can be found in Fan and Gijbels (1995b).

Based on the variance approximation above, we can also find the mean square error of a given fit: $MSE_p = bias_p^2 + var_p^2$. Variance can be approximated by

$$\widehat{var}_p^2 = \mathbf{S}^{-1}\mathbf{S}^*\mathbf{S}^{-1}\hat{\sigma}^2(X_0). \tag{13}$$

To approximate the bias, we can fit a polynomial of larger order (say of order $p+a$). Than $bias_p = (\mathbf{X}^t\mathbf{W}\mathbf{X})^{-1}\mathbf{X}^t\mathbf{W}\mathbf{r}$, with $\mathbf{r} = \mathbf{m} - \mathbf{X}\beta$,can be approximated by

$$\widehat{bias}_p = (\mathbf{X}^t\mathbf{W}\mathbf{X})^{-1}\mathbf{X}^t\mathbf{W}\tau \tag{14}$$

where τ is an $n \times 1$ vector:

$$\tau_i = \beta_{p+1}(X_i - x_0)^{p+1} + ... + \beta_{p+a}(X_i - x_0)^{p+a} \tag{15}$$

The choice $a = 4$ is $\sqrt{n}$ consistent, but $a = 2$ will be close to $\sqrt{n}$ consistent and a lot cheaper computationally. Using the *RSC* at a preliminary stage, we will obtain an estimate of $\hat{\tau}$ of τ. Then the mean squared error is approximated by:

$$\widehat{MSE}_p = \widehat{bias}_p^2 + \widehat{var}_p^2 \ . \tag{16}$$

3.2 Automatic bandwidth selection

From the previous section it is clear that we have to minimize *RSC* in order to find the best bandwidth, we have to begin by minimizing *RSC*. However, *RSC* reflects only residual squared criterion at one given point x_0, but we want to use the resulting regression in some neighborhood around x_0, say in the interval $[c, d]$. So, to find the estimated optimal bandwidth $\hat{h}$ for the polynomial of order p we first have to fit a polynomial of order $p+a$ and minimize the integrated version of *RSC*:

$$IRSC(h) = \int_{[c,d]} RSC(y;h)dy. \tag{17}$$

Then we have to minimize the integrated version of *MSE* for p^{th} order polynomial:

$$IMSE = \int_{[c,d]} \widehat{MSE}_p(y;h)dy. \tag{18}$$

using the bandwidth obtained from minimizing (17) to estimate τ. In both cases we will have a step function for bandwidth which we can smooth by averaging locally. Let us call the bandwidth obtained by minimizing (18) $\hat{h}^R$.

The resulting bandwidth $\hat{h}^R$ is the best estimated bandwidth for a polynomial of order p. Note that we have to do the same for *each* local polynomial.

Returning back to the Gaussian peak, I have found $\hat{h}^R$, based on p=3, and the Epanechnikov kernel, by minimizing *IRSC* and *IMSE* in the set:

$$h = \{0.05, 0.1, 0.15, 0.2, 0.25, 0.3, 0.35, 0.4, 0.45, 0.5\}.$$

The regression is shown in Figure 4. This regression is clearly better that any with the constant bandwidth in Figure 2 ($\mathbf{MSE} = 6.4$)

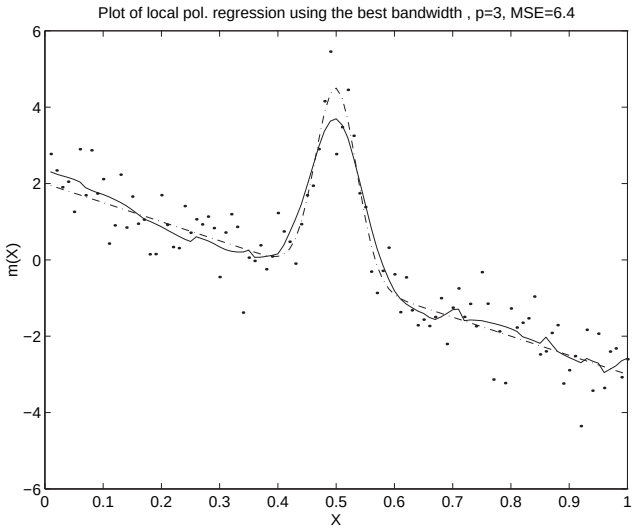

Figure 4: The local polynomial fit based on 100 equally spaced data points. The fit consists of 100 local equally spaced polynomials and the with variable bandwidth h chosen by minimizing *IMSE* for each local polynomial. The solid line is local polynomial fit, dash-doted line is Gaussian peak. (CPU time about 12 of minutes)

3.3 Automatic order selection

Fan and Gijbels (1995a) show that there is no reason to look at even powers of polynomials because we can go up to the next odd power without increasing variance but decreasing bias (free lunch).

Using the same *IMSE* we can choose between different orders of polynomial up to the order of p_{max}. First we have to fit polynomials of order $p_{max} + a$. Then, by using this polynomial, estimate *IMSE* for polynomials of order lower than p_{max}, and then choose the one which has the lowest *IMSE*. The regression for the Gaussian peak is shown in Figure 5. Clearly this regression is the best ($\mathbf{MSE} = 3.9$). Fan and Gijbels (1995a) show that the selected order of regression does not depend highly on bandwidth. Indeed it is highly robust to bandwidth selection.

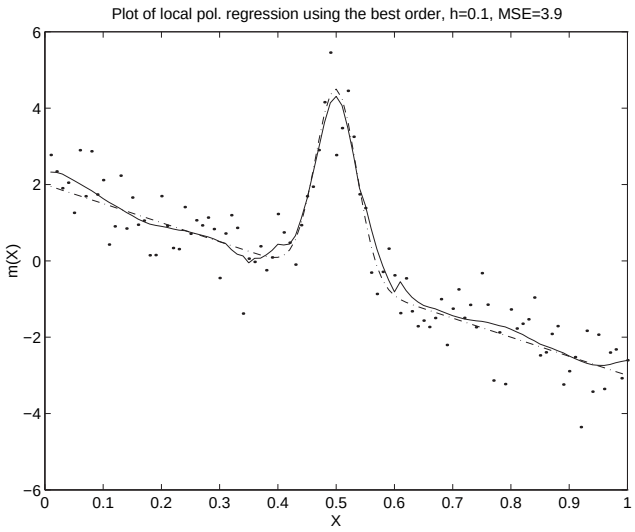

Figure 5: The local polynomial fit based on 100 equally spaced data points. The fit consists of 100 local equally spaced polynomials with variable order p chosen by minimizing *IMSE* for each local polynomial. The solid line is local polynomial fit, dashed -data, dash-doted line is Gaussian peak. (CPU time about 10 minutes)

4 COMPUTATIONAL ASPECTS OF MODEL SELECTION

Above I describe an integrated *RSC* and *MSE* and the model selection procedure by minimizing these two. However, theses are expensive to compute. One of the ways to approximate integrals is to evaluate the function in the middle of the interval and multiply it by the length of the interval. Usually this method works only for small intervals and very smooth functions. Let us look at *RSC* in the Gaussian peak example for center point x_0 on the interval $[0.5, 0.51]$, which is the interval for a single polynomial.

From Figure 6 we can see that *RSC* does not change a lot in the neighborhood of x_0. Figure (7) shows that it is almost flat, and Figure (8) shows that there is almost no difference in the *RSC* for different x_0 and the same h. Indeed we would see similar characteristics for any small enough intervals. Therefore we can minimize *RSC* instead of *IRSC* (length of the interval is just constant). Similarly, we can approximate *IMSE* by *MSE*.

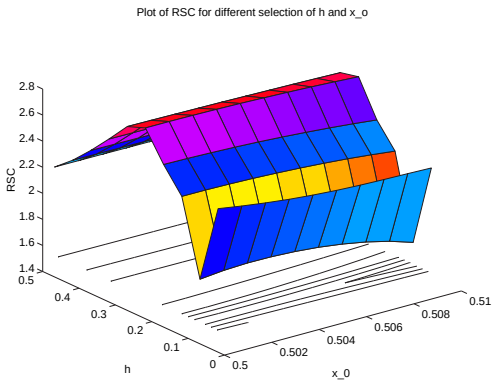

Figure 6: The plot of RSC for x_0 in the interval $[0.5, 0.51]$ and h in interval $[0, 0.5]$. One dimensional versions of this plot are shown in Figures 7,8.

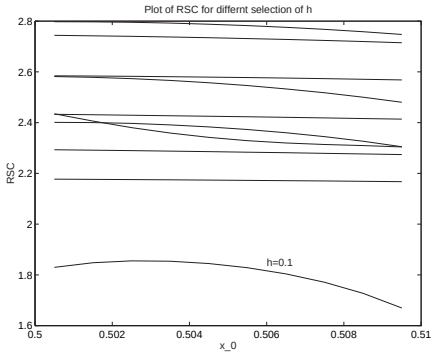

Figure 7: Values of RSC for x_0 in the interval $[0.5, 0.51]$, which has 10 separate lines for h in the interval $[0, 0.5]$.

In both model selection procedures, first we have to minimize *IRSC* for $p+a$ order polynomial to obtain $\hat{\tau}$ and then minimize *IMSE* using that $\hat{\tau}$. However, in the simulation that I have done, *IMSE* (approximated by *MSE*) gives almost the same selection for $\hat{h}$ as *IRSC* does. This happens because I minimize h on a discrete set of 5 or 10 values for h. Increasing the number of points in that set will increase the computational time tremendously. However, if one has the time and a powerful enough computer to minimize it on the set of 100 or more values for h, *IMSE* minimization will give a different result than *IRSC*, which may be noticeably different or not, depending on the data set.

As a result, for many practical cases we can combine the two model selection proce-

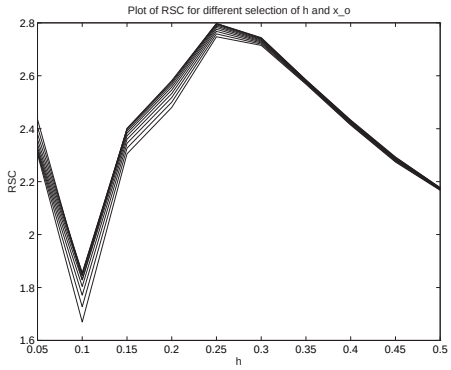

Figure 8: Values of RSC for h in the interval $[0, 0.5]$, which has 10 separate lines for x_0 in the interval $[0.5, 0.51]$

dures above as follows. First we fit $\frac{p+1}{2}$ regressions for each choice of h, then we pick h and p corresponding to the regression with lowest RSC . Note, this procedure has to be done at each point, and at each point we have to select the best bandwidth and order.

Figure 9 shows the regression using this method. This regression is the same, as one in Figure 5 ($\mathbf{MSE} = 3.9$), but computationally a lot faster.

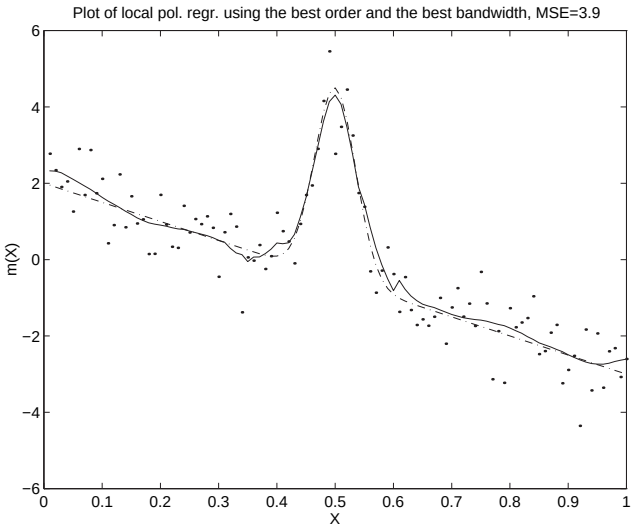

Figure 9: The local polynomial fit based on 100 equally spaced data points. The fit consists of 100 local equally spaced polynomials with variable order p and bandwidth h chosen by minimizing RCS for each local polynomial. The solid line is local polynomial fit, dash-doted line is Gaussian peak. (CPU time about 4 minutes)

5 BINARY TREE INTERVALS SELECTION

Suppose we chose a vector $\mathbf{x_0}$ of length n_{x_0}, and we want to choose between n_h h's, and n_p p's. By the procedure above we would have to make $n_{x_0} \times n_h \times n_p$ local regressions. However, if we choose to have large bandwidths in a region, less center points x_0 should be needed. None of the model selection procedures above look at varying the number of x_0, but it is important. Too many x_0's would increase the cost of computation, too few x_0's would produce singularities. Note that the natural lower bound on h is half of the distance between x_0's. Also, it is clear that in neighboring intervals which are small compared to bandwidth size, the bandwidth should be similar, or even the same. Hence an algorithm which adapts the number of center points with the bandwidth is intuitively appealing. We can do so efficiently by looking at binary tree intervals. Here is a small algorithm of how it works:

(1) First we use n_{x_0} number of x_0's and evaluate *IRSC* using a variable order of polynomials on the whole interval of regression $[a, b]$, using some original bandwidth h.

(2) Then we divide the original interval evenly into two subintervals: $[a, \frac{a+b}{2}]$ and $[\frac{a+b}{2}, b]$.

(3) Repeat steps (1) and (2) for each of the subintervals using n_{x_0} number of x_0's on each of the subintervals and bandwidth $g(h)$ until minimum tolerable bandwidth is reached.

The result of this algorithm is a sequence of local polynomial estimates of $m(x)$, each using twice the number of center points x_0 (and a reduced bandwidth) as its predecessor. A final estimate with an adaptively chosen number of center points may be created by selecting non-overlapping subsections from these various estimates.

I choose to select the model with lowest total *IRCS*. One way would be to look at all

possible combinations, but the number of possible combinations grows up very fast. The table below shows just how quickly this number grows.

# of times n_{x_0} in doubled	# of possible combinations
0	1
1	2
2	5
3	26
4	677
5	458330

Clearly another more efficient method is required. Coifman and Wickerhauser (1992) derived an algorithm which they use for finding the best basis for a given signal, among a library of orthogonal bases. This algorithm, without any changes, can be used to find the best combination.

In this algorithm we compare the *IRCS* of each interval to the sum of the *IRCS*$'s$ of its two children and choose whichever has lower *IRCS*. Figure 10 shows an example of *IRSC* for each interval, and Figure 11 shows intervals which we would choose.

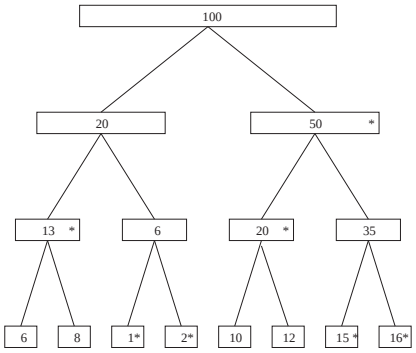

Figure 10: Example of chosing best bandwidth based on binary tree. (Taken from Kolaczyk 1995)

A word about the function $g(h)$ which has to reduce bandwidth h with each interval division. Fan (1993) says that it has to be of the form: $g(h) = c * n_{x_0}^{\alpha}$, where $0 < \alpha \leq 1$,

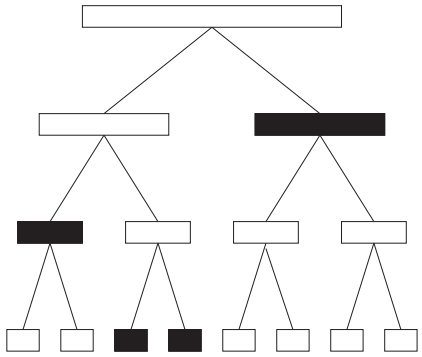

Figure 11: Example of chosing best bandwidth based on binary tree. Dark boxes indicates intervals choosen. For data see Figure 10. (Taken from Kolaczyk 1995)

and the constant c depends on the range of the data, number of points in the data, etc. I have chosen to use $g(h) = c\sqrt{n_{x_0}}$, but different choice of g would lead to a similar result but will change the number of divisions needed.

Figure 12 shows this procedure applied to the Gaussian peak. In this regression I started with $n_{x_0} = 16$ and allowed the algorithm to double it three times. Using the selection procedure described above, 80 central points where chosen. The regression has an $\mathbf{MSE} = 4.8$, which is a little bit higher than in best order selection by minimizing *MSE* (Figure 5), but there I used polynomial of order up to five, and here only of order up to 3, but the computational time is a lot lower. The lower plot shows the selected bandwidths. There are a few areas where bandwidth was selected to be smaller, but the order was select to be 3 in the interval $[0.3822, 0.6897]$ only. There where more point in the intervals with lower bandwidth.

6 DISCUSSION

The simple example presented in this paper, illustrates the computational advantages to be gained by rougher approximates of the minimization criteria, with little loss in quality. Bandwidth selection tools seem to be a promising method in the automa-

tion of model searching, and with this tool we may get improved performance by local polynomials for different data.

It would be interesting to find a criterion so that the number of divisions can be automated. For example, if either of *IRSC* or *IMSE* are convex, we could look for the minimum, and stop once this is accomplished.

7 ACKNOWLEDGMENTS

I wish to thank my advisor Eric Kolaczyk, for his unending support as I completed this paper.

8 REFERENCES

Coifman, R.R. and Wickerhauser, V.W. (1992), Entropy-Based Algorithms for Best Basis Selection. *IEEE Transactions on Information Theory*, **38:2**, 713-718

Fan, J. (1993), Local Linear Regression smoother and there Minimax Efficiencies. *The Annals of Statistics*, **21**, 196-216.

Fan., J., and Gijbels, I. (1992), Variable Bandwidth and Local Linear Regression Smoothers. *The Annals of Statistics*, **20**, 2008-2036.

Fan., J., and Gijbels, I. (1995a), Adaptive Order Polynomial Fitting: Bandwidth Robustification and Bias Reduction. *Journal of Computational and Graphical Statistics*, **4**, 213-227.

Fan., J., and Gijbels, I. (1995b), Data-driven Bandwidth Selection in Local Polynomial Fitting: Variable Bandwidth and Spatial Adaptation. *Journal of Royal Statistical Society*, Ser. B **57**, 371-394.

Gasser, T., and Müller, H.-G. (1979), Kernel Estimation of Regression Functions, in *Smoothing Techniques for Curve Estimation,* Lecture Note in Mathematics **757**, New York: Springer, 23-68.

Gasser, T., and Müller, H.-G. (1984), Estimating Regression Function and There derivatives by the Kernel Method. *Scandinavian Journal of Statistics*, **11**, 171-185.

Hastie, T. J., and Tibshirani, R. J. (1990), *Generalized Additive Models*, Monographs on Statistics and Applied Probability 42, London: Chapman and Hall.

Kolaczyk, E (1995), Statistics 394 Class notes, The University of Chicago.

Marron, J. S., and Nolan, D. (1988), 'Canonical Kernels for Density Estimation. *Statistics and Probability Letters*, **7**, 195-199.

Müller, H.-G. (1988), *Nonparametric Regression Analysis of Longitudinal Data*, Lecture Notes in Statistics 46, Berlin: Springer.

Nadaraya, E. A. (1964), On Estimating Regression. *Theory of Probability and Its Applications*, **9**, 141-142.

Reinsch, C. H. (1967), Smoothing by Spine Functions. *Numerische Mathematik* **9**, 177-183.

Seifert B., Brockmann, M., Engel, J., and Gasser, T. (1994), Fast Algorithms for Nonparametric Curve Estimation. *Journal of Computational and Graphical Statistics*, **3**, 192-213.

Seifert B., and Gasser, T. (1996), Finite-Sample Variance of Local Polynomials: Analysis and Solutions. *Journal of the American Statistical Association*, **91** 267-275.

Wahba, G. (1990), *Spline Models for Observational Data*, CBMS-NSF Regional Conference Series in Applied Mathematics, Vol 59, Philadelphia: Society for Industrial and Applied Mathematics.

Watson, G. S. (1964) Smooth Regression Analysis. *Sankhyā*, Ser. A, **26** 359-372.

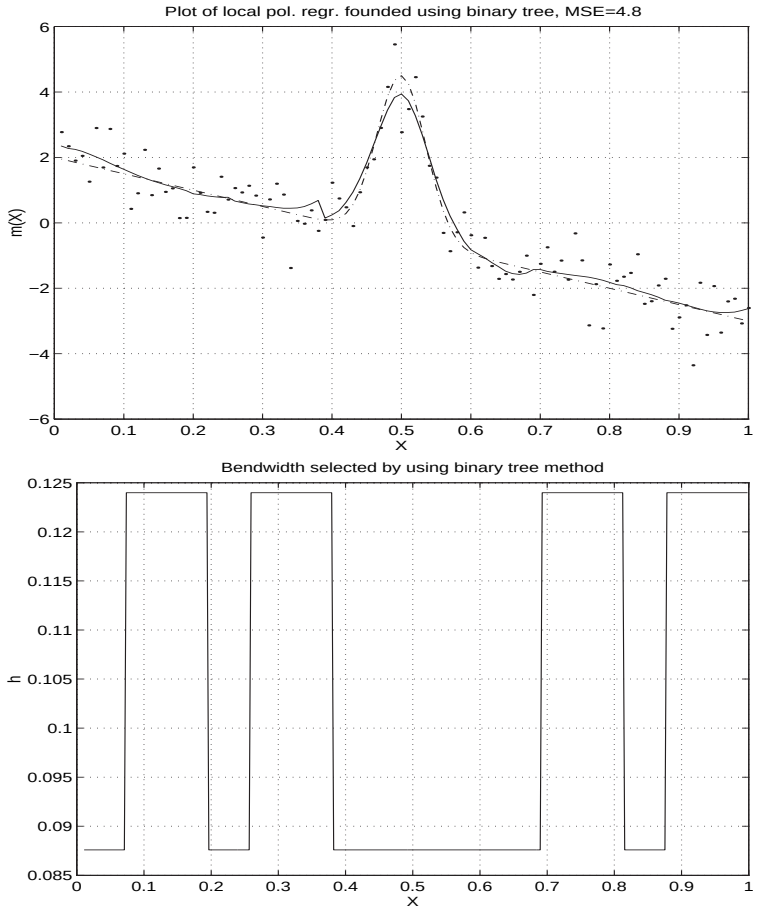

Figure 12: **Upper plot:**The local polynomial fit based on 100 equally spaced data points. The fit consist of 100 local equally spaced polynomials with variable order p and bandwidth h chosen by minimizing RSC using binomial tree. The solid line is local polynomial fit, dashed -data, dash-doted line is Gaussian peak. (CPU time about 2 minutes)
Lower plot: Bandwidth selected by minimizing RSC using binomial tree