Motion Picture Analysis of Smoothing

J. S. Marron, D. Ruppert, E. K. Smith and G. Conley

Abstract

This paper presents an innovative teaching tool, which demonstrates local polynomial smoothing via a movie, and several new graphical devices. The movie emphasizes the "data fitting inside a moving window" interpretation of local polynomial smoothing. A number of lessons are highlighted by showing three smooths simultaneously in the movie. An additional view of the smoothing process is the locally fit polynomials, called "snakes", because of their interesting motion. Additional insight comes from a Pythagorean Theorem based visualization of the Mean Squared Error, and its components, the standard deviation and bias.

1 Introduction

Local polynomial smoothing is a useful way to find structure in regression data. Early references in the modern literature are Stone (1975, 1977), but see Cleveland and Loader (1996) for earlier references dating back to the last century. Practical application started in a serious way with the introduction of LO(W)ESS by Cleveland (1979) and Cleveland and Devlin (1988). This area bloomed in terms of theoretical work following the papers of Fan (1992, 1993). For additional historical discussion, for many interesting real data examples, and for discussion of other aspects of local polynomial smoothing, see Wand and Jones (1995), Fan and Gijbels (1996), Simonoff (1996), and Bowman and Azzalini (1997).

As seen in Figure 1, good visual insight into the performance of local polynomial smoothing, at a single point, comes from thinking about doing least squares fitting of a polynomial "inside a window". This style of presentation can be found for example in Hastie and Loader (1993), Wand and

Jones (1995), Seifert and Gasser (1996) and Fan and Gijbels (1996). However, in real data applications, it is important to study the full curve, which comes from moving the window over the range of estimation. This cannot be done in a static plot, but is conveniently displayed in a dynamic plot, i.e. a movie, where "time" is the location of the smoothing window.

This type of movie is presented in this paper, as a tool for teaching how local polynomial smoothing works. We present both a set of illustrative examples that are easily WWW accessible in the MPEG format, and Matlab software for constructing other possible examples. In addition to being engaging, and thus a useful teaching tool, the examples also highlight some surprising aspects, as seen in Sections 4.2, 4.4 and 4.5.

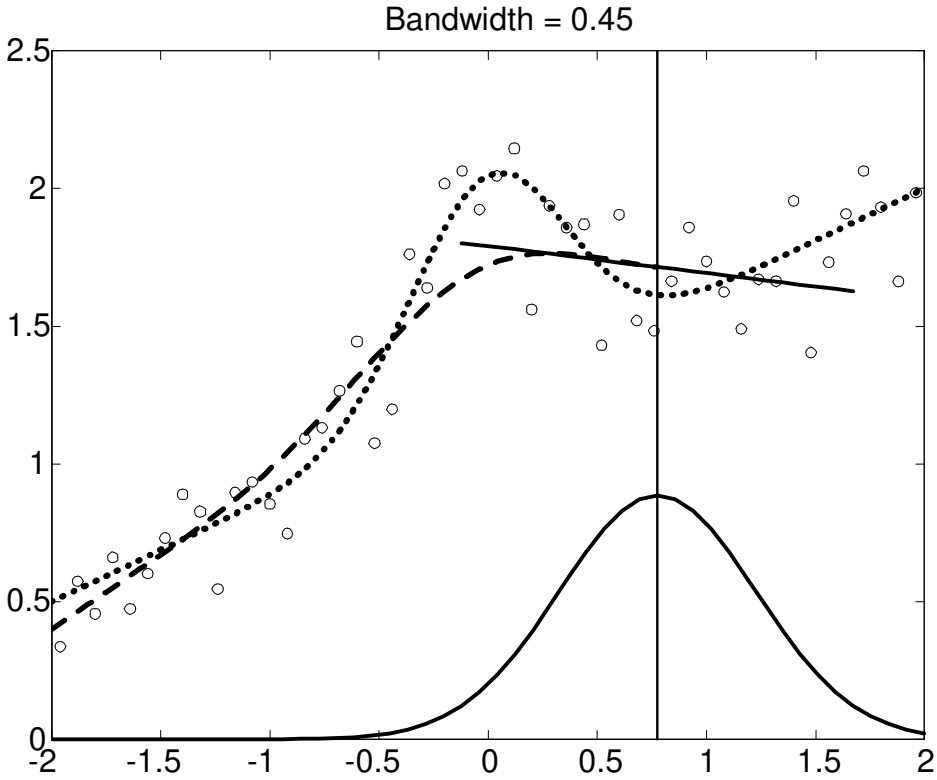

FIGURE 1: *One part of one movie frame, illustrating local linear regression.*

The format of our examples is discussed in Section 3. In addition to side by side movie frames, which allows comparison of important aspects of smoothing, we propose additional graphical devices which highlight various

important concepts. One of these is a "snake window" allowing direct comparison of the local fits. Another is a new visualization of the Mean Squared Error and its components of variance and squared bias, represented via the Pythagorean Theorem.

Figure 1 shows one part of a single frame of such a movie. This is a simulated example, where the underlying regression curve is essentially Gasser's blip (a line plus a scaled Gaussian density), shown with a dotted line type. Mean $\mu = 0$ Gaussian errors, with standard deviation $\sigma = 0.2$, are added to the underlying curve at 50 equally spaced points, to result in data shown as circles. These data are smoothed via the local linear method, using Gaussian kernel weights, and the bandwidth $h = 0.45$. These kernel weights are represented by the solid curve near the bottom, where the vertical line represents the current point of estimation. In the movie, this estimation point (and thus the whole kernel window) moves from right to left, so the estimation process is viewed "on line". The solid line, called a "snake" because it often writhes for higher degree local polynomials, shows the locally weighted least squares fit at this location. As the window slides along the smooth is created as the thick dashed curve shown in Figure 1.

The basic concepts and notation for local polynomial smoothing are given in Section 2. Specific examples, with the focus on insights to be explained to students, are discussed in detail in Section 4. Specifics include movies that highlight:

- the importance of the bandwidth, i.e. the width of the smoothing window, in Section 4.1.
- the impact of polynomial degree, in Section 4.2.
- the effect of the kernel window shape, and also the concept of canonical kernels, in Section 4.3.
- issues about "signal to noise ratio", through varying the noise variance and the sample size, in Section 4.4.
- how changing the design points, i.e. the location of the x points, affects the estimation, in Section 4.5.
- the dependence of performance on the scedasticity, i.e. the local level of noise variance, in Section 4.6.

Details about the Matlab implementation are given in Section 5.

2 Basics of local polynomial smoothing

Nonparametric regression, or scatterplot smoothing, can be formulated as using data $(X_1, Y_1), ..., (X_n, Y_n)$, shown as circles in Figure 1, generated by a model of the form

$$Y_i = m(X_i) + \varepsilon_i, \ i = 1, ..., n,$$

to estimate the regression curve $m(x)$, shown as the dotted curve in Figure 1. The X_i are called the design points, and can be either random or fixed, e.g. equally spaced. The curve $m(x)$ is called the underlying target curve, or the signal, and is assumed to be "smooth", although a few jump discontinuities are allowed. The errors, i.e. noise terms, ε_i are assumed to be independent identically distributed mean zero random variables, with variance σ_i^2. The error distribution is said to be homoscedastic when the σ_i^2 are all the same, and then σ^2 is used to denote the common value. When the error variances σ_i^2 are different, the errors are said to be heteroscedastic. In the simulated examples considered in this paper, the error distribution is always Gaussian.

In the examples in Section 4, various targets, designs and variance structures are considered.

As noted in the introduction, the local polynomial smoothing method is a useful way to recover the signal $m(x)$, from the noisy data $(X_1, Y_1), ..., (X_n, Y_n)$. Let

$$f_p(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_p x^p$$

denote a polynomial of degree p. The local linear fit chooses the coefficients $a_0, ..., a_p$, by weighted least squares. The weights are local in character, as determined by the kernel function. For a kernel function K that is symmetric about 0, e.g. the Gaussian density, or some other choices as given in Section 4.3, the kernel weight assigned to the data point Y_i is $K_h(x - X_i)$, as shown by the lower solid curve in Figure 1. The subscript of h denotes a rescaling of the kernel function, by the bandwidth h, in particular

$$K_h(x) = \frac{1}{h} K\left(\frac{x}{h}\right).$$

The local least squares criterion, for estimating m at the location x, shown as the vertical line in Figure 1, is

$$LS\left(a_0,...,a_p\right)=\sum_{i=1}^{n}\left[Y_i-f_p(X_i-x)\right]^2 K_h\left(x-X_i\right).$$

Minimization is performed over the parameters $a_0,...,a_p$, and the estimate is taken to be the intercept term, i.e. $\widehat{m}_h\left(x\right)=a_0$, since a_0 is the value of the centered polynomial $f_p(\cdot-x)$, at the point of estimation, x. As the location x moves from left to right, the curve $\widehat{m}_h\left(x\right)$ is traced as shown with the dashed line type in Figure1.

3 Description of view

While it is informative to study a movie version of Figure 1, effective learning comes from comparison of movies. This is difficult to accomplish with separate movies, so our standard movie simultaneously shows 3 different versions of Figure 1, which allow direct visual comparison of various aspects of the local polynomial smoothing process.

In addition, we have developed other graphical devices, which highlight other aspects of smoothing.

3.1 Snake window

Useful insights come from visual study of the "snakes", i. e. the local fits. The snake in Figure 1 is shown by the solid line. Direct comparison of snake performance comes from overlaying them in a separate window. One frame of a typical snake window is shown in Figure 2.

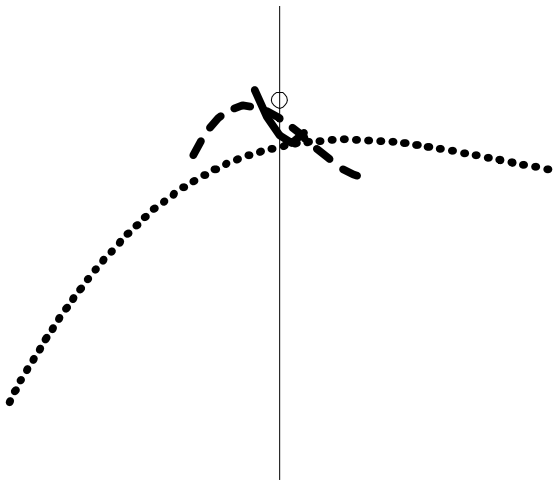

FIGURE 2: *The snake window, showing three local cubic fits, from one frame of a movie.*

The three snakes are the local cubics that were fit to the data, at the same location, one from each of the three versions of Figure 1. In Figure 2, the solid snake is for bandwidth $h = 0.1$, the dashed snake is for bandwidth $h = 0.3$. and the dotted snake is for bandwidth $h = 0.9$. The length of each snake represents the "effective window width" taken to be two standard deviations of the Gaussian kernel function. The location of estimation is represented by the vertical line (which is fixed in this display), and the height of the true underlying regression curve at this location is represented by the circle (which moves vertically as the kernel window moves.

The dashed snake ($h = 0.3$) is closest to the circle, i.e. to the target curve. This is consistent with it being an intermediate amount of smoothing, i.e. a good trade-off between under and over-smoothing. The shape of this snake is relatively consistent with the shape of the underlying curve in this region. The solid snake ($h = 0.1$) is undersmoothed, and thus follows the noise in the data more than the underlying curve. Thus its shape is actually curved in the wrong direction in Figure 2, and the value at the estimation point is farther from the target than for $h = 0.3$. This reflects the increased variance which quantifies this type of performance. In the movie version, it is clearly

seen that undersmoothed snakes oscillate very rapidly. The dotted snake ($h = 0.9$) is oversmoothed, because the kernel window includes data from quite far away. This oversmoothing reduces the flexibility of the cubic snake so that it does a poor job of representing the local structure of the underlying target, and the value at the estimation point is even further from the target than the others. This effect is quantified by the bias, which is quite large in this case.

A graphical display of the impact of variance and bias is developed in the next section.

3.2 Root Mean Squared Error windows

The Root Mean Squared Error and its components, variance and squared bias, play an important role in understanding smoothing methods. Further description and deep analysis can be found in several of the above monographs. Intuitive insight into these measures comes from another graphical device, which is part of our movie.

The (conditional) Mean Square Error is defined here as

$$MSE = E\left[\widehat{m}_h(x) - m(x) | X_1, ..., X_n\right]^2.$$

Conditioning on the X_i is natural, since the regression function $m(x)$ is a conditional quantity. For the same reason, this practice is usual in the mathematical analysis of linear models. Another viewpoint, is that corresponds to using the X_i Empirical Distribution Function that is "closest to the data". This issue is studied further in Section 4.5.2. The variance and squared bias components are effectively displayed via the Pythagorean relationship

$$MSE = var + bias^2$$

which is equivalent to

$$RMSE = \sqrt{MSE} = \sqrt{sd^2 + bias^2}.$$

The simultaneous relationship between sd and $bias$ is usefully represented as a point in the $(sd, bias)$ plane. The position of the point, and the Pythagorean Theorem, are highlighted by drawing in a right triangle whose hypotenuse is the line connecting $(0, 0)$ and $(sd, bias)$. There are two such triangles, that

are both shown, resulting in a rectangle, to give equal emphasis to *sd* and *bias*. The length of the diagonal of this rectangle, i.e. box, is the $RMSE$. Finally, for easy comparison of $RMSE$, even when the *sd* and *bias* trade-offs are quite different, the $RMSE$ is highlighted by drawing the semicircle centered at the origin, whose radius is the $RMSE$. Thus the semicircle meets the box at its corner opposite the origin. The semicircles allow quick visual comparison of the $RMSE$ across the smooths being compared. An additional interpretation of this semicircle is that its area is proportional to the $MSE = RMSE^2$. The box corresponding to each circle adds the additional information of how the $RMSE$ is divided between *sd* and *bias*.

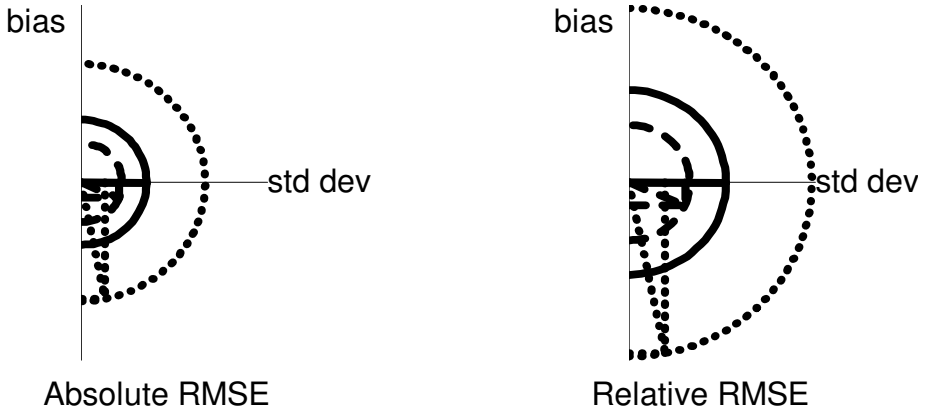

FIGURE 3: *Graphical display of standard deviation, bias and Root Mean Squared Error, via the Pythagorean Theorem.*

This graphical device is demonstrated in Figure 3, where the movie frame, and the estimation setting are the same as for Figure 2. The small solid snake in Figure 2 came from a small bandwidth smooth, and had large variance and small bias, thus the $RMSE$ box lies close to (essentially on top of) the *sd* axis. The dashed snake came from the intermediate bandwidth smooth, and its box is not close to either axis. The dotted large bandwidth smooth had higher bias, and its box is closer to the *bias* axis. The semicircles make it clear that the intermediate bandwidth dashed estimate is best, in terms of having the smallest $RMSE$. The large bandwidth dotted snake has the worst $RMSE$ of the three in this frame. These relationships typically change rapidly during the course of a movie.

Precise definitions of the *sd* and *bias* that are used here are given in Section 5.1.

Two such plots are shown, that are the same except for the scaling. The left plot is on an "absolute scale", determined by the largest $RMSE$ at the start of the movie (usually the largest in the whole movie, since errors are typically largest at the boundary). The right plot is on a "relative scale", where the largest of the three semicircles always has radius one. Both scales are useful and convey different types of information.

In the figures in this paper, the three different estimates and snakes are represented with different line types. But in the movies, colors are instead used to differentiate these. Common colors are used for the title of the main window, for the corresponding snakes and for the corresponding $RMSE$ graphics. The movie colors (not the present line types) will be referred to in the discussion of the examples in Section 4.

Best continuous viewing of the movies is done in "back and forth mode", where the movie is run in reverse, after the end is reached.

4 Discussion of examples

The examples discussed here should be viewed as movies. Easily viewable MPEG files, referred to at the beginning of each subsection below, can be found at the web address:

`http://www.stat.unc.edu/faculty/marron/Movies/locpoly_movies.html`,

together with the Matlab software that generated them, and a more detailed version of this paper. For more information on MPEG players follow the link "Back to Movies Table of Contents" at the bottom of that page. Each case is indexed by the number of the button, which calls it in our Matlab software.

4.1 Comparison of bandwidths

These movies show the importance of the bandwidth in smoothing methods.

4.1.1 Movie 1a, blip target, local linear case

This context is the same as for Figure 1. The three bandwidths, $h = 0.05$, $h = 0.15$ and $h = 0.45$ were carefully chosen to represent "undersmoothing", "effective smoothing" and "oversmoothing" respectively.

The short yellow snake in the small bandwidth ($h = 0.05$) window oscillates wildly as the estimated curve traces a very jagged path. This occurs because the window is so small that the estimate feels the noise in the data too strongly, which results in the slope of the yellow snake being often very different from the underlying target. In this case, the variance of the estimate is larger than for the other bandwidths, but the bias is smaller. This bias appears as the light blue estimated curve being "centered correctly". In the $RMSE$ plots, the low bias and high standard deviation appear in terms of the yellow rectangle lying close to the x axis. The large standard deviation dominates, and so the yellow semicircle is often the largest, i.e. $h = 0.05$ has the largest $RMSE$.

The medium green snake in the intermediate bandwidth ($h = 0.15$) window is much more steady, and follows the true curve more closely. The resulting curve estimate is much more accurate. There is some small bias near the central peak, and the estimate wobbles on the right side, but this is an overall good trade-off in the bandwidth. The snake window also shows the increased stability, and that the estimate is usually closer to the target. The $RMSE$ windows show that the $h = 0.15$ smooth has more bias than for $h = 0.05$, but the decreased standard deviation makes this worth while in terms of $RMSE$, since the green semicircle is always inside the yellow semicircle.

The long red snake in the large bandwidth ($h = 0.45$) window is even steadier, but now loses the flexibility to follow the shape of the true underlying target curve. The resulting estimate suffers from too much bias in regions of high target curvature, because a line cannot capture the features of the underlying curve at this scale. However, in regions where the target is nearly linear, the reduction in bias means the smooth fits the best. This behavior is clearly visible in the snake window, since in regions of high curvature, the long red snake is far from the target circle. This behavior also appears clearly in the $RMSE$ window. When the underlying curvature is high, the red rectangle lies close to the y axis, since bias is dominant and the $RMSE$ is very large. But when the target is nearly linear, bias is small which results in smaller $RMSE$ than for the smaller bandwidths. These effects happen quickly as the movie runs, so it is helpful to stop the movie at appropriate points to observe them.

It is useful to summarize the above observations in terms of aspects of the smoothing problem. The $RMSE$ plots show that the standard deviations

are always ordered as red $<$ green $<$ yellow, which is sensible, because the local fits are more stable when there are more points in the kernel window. The biases are usually ordered as red $\gg$ green $>$ yellow ≈ 0. These tend to move up and down together, and have sign changes at inflection points of the underlying target curve $m(x)$. The green $RMSE$ semicircle is usually smallest, except in regions where the target is nearly linear, where the red is best. The relative order of the semicircles changes from location to location, depending on whether standard deviation or bias are dominant.

4.1.2 Movie 1b, blip target, local cubic case

This movie uses a cubic as the local fit. When comparing the linear and cubic fits many similarities surface. As in Movie 1a, it is useful to observe the dependence of the performance of the estimate on the length of the bandwidths, $h = 0.1$, $h = 0.3$, and $h = 0.9$, which correspond to "undersmoothing", "effective smoothing", and "oversmoothing." Larger bandwidths are used than in Movie 1a, since in general larger bandwidths are required for similar performance for higher polynomial degree. These were chosen so that the central green estimate gives an overall good trade off between sd and $bias$.

There are strong similarities between this movie and Movie 1a. As in the linear fit examples, the cubic yellow snake traces a very jagged path because the estimate feels the noise in the data. The cubic green snake also follows the same pattern as the linear green snake by following the true curve more closely, and does better because curvature in the snake is allowed. The cubic red snake also shows a strong similarity to the linear fit. The wide window reduces flexibility, resulting in an inaccurate approximation to the underlying target curve. As a result of the similarities between the plots in both movies, the standard deviations and biases of the three windows share many of the same characteristics. Both the standard deviation and the bias of the snakes follow the same order as the local linear case. In particular, the yellow snake has the smallest bias and the largest standard deviation, etc.

Many differences can also be found between the cubic and linear graphs. Because the local cubic fits use larger bandwidths, the changes in the bias and variance are more stable. The cubic snakes have a greater amount of curvature than the linear snakes. Therefore, the bias is more complicated in the cubic case than in the linear case. As a result of this, the large bandwidth

bias in Movie 1b is larger at the right boundary than in the first movie, as seen by stopping the movie there. The reason is the local cubic fit snake doesn't have the flexibility to follow the true curve well.

4.1.3 Movie 1c, blip target, local quintic case

As noted in Section 5.4 of Wand and Jones (1995), the quintic is a smaller step up from the cubic, than the cubic is from the linear. Hence, this movie bears a closer resemblance to Movie 1b, than Movie 1b does to Movie 1a. The bandwidths used in this movie are larger, which is appropriate for a higher degree polynomial. This is because the decreased bias for larger p is accompanied by increased variance, as made clear in Section 3.3.1 of Fan and Gijbels (1996). One source of great interest is the increased flexibility caused by $p = 5$. Because of this flexibility, the yellow $h = 0.13$ snake dances even more, and the green $h = 0.4$ snake follows the true curve more closely. Although the bandwidth is larger, the increased amount of flexibility allows the red $h = 1.2$ snake to come close to the peak at $x = 0$. This is an improvement over the red snake in Movie 1b. It is in Movie 1c that the snakes look most like snakes.

The $RMSE$ displays teach us many of the same lessons as discussed for Movies 1a and 1b. The exception occurs for the yellow snake. In this case, the error at the boundaries is very large, which results in very small $RMSE$, as shown in the Absolute $RMSE$ window, in the interior. This is caused by "too few degrees of freedom" of the data near the edges, for fitting a polynomial of degree 5.

4.1.4 Movies 2a, 2b, 2c, piecewise target,

The buttons on the second row of our menu, also are for direct comparison of bandwidths, in the three parts of our movies. But the underlying signal is now a piecewise polynomial

$$m(x) = \begin{cases} 1.5x + 3.5 & \text{if } 2 \le x \le -1 \\ (1.3x)^2 + 0.5 & \text{if } -1 < x \le 1 \\ 0.5 & \text{if } 1 < x \le 2 \end{cases},$$

which has a cusp at $x = -1$ and a jump discontinuity at $x = 1$.

Because the target used in these movies has a more complex structure, they move at a faster pace. Therefore, it is important to stop the MPEG files at various points to see rapidly disappearing structure. The under and over smoothing issues are quite similar to Movies 1a, 1b, and 1c. However, which bandwidth is "best" depends on x the point of estimation.

Movie 2a features the degree $p = 0$ local polynomial smoother, which is a kernel weighted local average, called the Nadaraya - Watson estimate, studied by Nadaraya (1964) and Watson (1964). This is also local constant fitting, as is made clear from the always horizontal snakes. This local constant character causes inferior fitting, in regions of change of $m(x)$, but better fitting near $x = 1$.

Movie 2b uses the degree $p = 1$ local linear smoother. The added flexibility allows for better fits where $m(x)$ changes, especially at the boundaries, but performance could be improved where $m(x)$ is more curved, near $x = \pm 1$. The extent to which this improvement is available is seen in Movie 2c, where the degree $p = 3$ local cubic is shown.

The $RMSE$ displays for Movies 2a-2c emphasize the observations made above. In Movie 2a, the errors are generally larger at the left boundary than at right (seen clearly in the Absolute $RMSE$ window). This occurs because the target $m(x)$ slopes up steeply at the left, and thus is not locally fit well by a constant function. But on the right, $m(x)$ is flat and is fit well. The $RMSE$ displays for Movies 2a and 2b also show that the error for red, at the $x = -1$ peak, is even worse than at the left boundary. This is easily observed in the Absolute $RMSE$ window, since the red semicircle extends partly outside the viewing range. This effect is even stronger, at the peak where $x = 1$, in Movies 2a and 2c, and is largest in Movie 2b, where all three $RMSE$ semicircles extend outside. Larger error was expected, because the peak is thinner, so the best fit local average will be lower.

4.2 Comparison of polynomial degree

Choice of degree of local polynomial has been a controversial subject, and there is still no consensus on the "best" degree. For example, Cleveland, Grosse and Shyu (1992) frequently prefer degree 2. However, in Section 5.4 of Wand and Jones (1995) and in Section 3.3.2 of Fan and Gijbels (1996) the asymptotic case for odd degrees being preferable is given. But in a simulation study, Ruppert (1997) found that degree $p = 3$, did not improve over $p = 2$

at interior points, and near the boundary higher variability of $p = 3$ made it inferior to $p = 2$. In Sections 3 and 7.2 of Cleveland and Loader (1996) an overall viewpoint shows that the "right degree" depends on the context, with every degree being "best" in some situations. All of these points are clearly demonstrated using our movies.

Movies 2a-2c showed that each of the degrees $p = 0$, 1 or 3 were best in particular situations, and at some locations. But the comparison there was complicated by differing polynomial degrees being in separate movies. Putting different degrees in the simultaneous windows allows more direct comparison of degree.

Our movies both demonstrate the expected ideas, and also contained some surprises. In particular, some of the asymptotic lessons did not hold, for reasons that are made clear in the following discussion.

The lessons in this section are effectively demonstrated with a different underlying target curve, which is a shift and scale of a sin wave with three cycles in the region of interest,

$$m(x) = .75 \sin(3\pi x/2) + 1.25.$$

The buttons 3a, 3b, 3c compare a low range of degrees, $p = 0, 1, 2$. The buttons 4a, 4b, 4c compare a higher range of degrees, $p = 2, 3, 4$.

4.2.1 Movies 3a, 3b, 3c, low degree

Movies 3a, 3b and 3c use the relatively small bandwidth $h = 0.09$, the intermediate bandwidth $h = 0.27$, and the larger bandwidth of $h = 0.8$, respectively. Asymptotic theory, e.g. as discussed in Chapter 5 of Wand and Jones (1995) and Chapter 3 of Fan and Gijbels (1996), suggests that the yellow local constant $p = 0$ and the green local linear $p = 1$ should be different at the boundary, and should be similar in the interior. The anticipated boundary difference shows up in Movies 3a and 3c. But the yellow and the green are surprisingly similar in Movie 3b, since the bandwidth $h = 0.27$ yields a green local linear fit with essentially 0 slope. The anticipated similarity in the interior shows up in all three movies, but note the "interior" region is very small for Movie 3c. The yellow and green snakes are only similar in terms of their intercepts (i.e. in terms of $\widehat{m}_h(x)$, which is where the snakes cross the light blue vertical line in the snake window) and have rather different slopes, i.e. the snakes are visually different. Furthermore,

the yellow and green $RMSE$ displays are all virtually the same (with only the green visible, since it is overplotted) for x in the interior, but separate for x at the boundaries.

The asymptotic rate of convergence suggests that the green local linear $p = 1$ and the red local quadratic $p = 2$ should be similar at the boundary, but the red should have better local curvature adaptation in the interior. The boundary similarity does not hold up well. In Movie 3a, the small window width gives an unstable local quadratic fit, which is different from the linear on the left, but similar on the right. In Movie 3b, the large window width includes the neighboring peak or valley, so the local fit has substantial curvature, i.e. the usual asymptotic lesson doesn't apply. This effect is even stronger for Movie 3c. Because the high noise drowns out the benefits of the adaptability to curvature, the anticipated superior performance of the local quadratic in regions of curvature does not show up well for the small bandwidths in Movie 3a. This appears in all of the light blue estimates $\widehat{m}_h(x)$, the snakes, and the $RMSE$. However, the improvement does show up dramatically, in all ways in Movie 3b, where the green smooth only recovers about half the height of the peaks and valleys. The one exception is near the inflection points, where the red $RMSE$ is slightly bigger than the green (the movie should be stopped to examine this) where the usually dominant bias of the green crosses zero. The bandwidth used in Movie 3c, is so large that none of the polynomial degrees has the flexibility to adapt to the periodic sine wave structure, so the green and the red results are very similar in all ways, in particular completely smoothing away the peaks and valleys, which is reflected in the expected very large *bias*.

4.2.2 Movies 4a, 4b, 4c, high degree

Movies 4a, 4b and 4c use the relatively small bandwidth $h = 0.2$, the intermediate bandwidth $h = 0.4$, and the larger bandwidth of $h = 0.8$, respectively. These bandwidths are larger than used for Movies 3a-3c, which is appropriate to get the same amount of smoothing for higher polynomial degrees.

The above referenced asymptotic theory suggests that the yellow quadratic, $p = 2$, should be similar to the green cubic, $p = 3$ in the interior regions. They are close, especially for the smaller bandwidth as shown in Movie 4a. But a noticeable difference shows up in the $RMSE$ displays for Movies 4b and 4c. This perhaps reflects the idea that larger sample sizes are needed

for the asymptotic to take effect, as suggested by Marron and Wand (1992).

The asymptotic theory suggests more differences between the green cubic, $p = 3$, and the red quartic, $p = 4$. These differences are not dramatic in terms of the estimates in Movie 4a, although they are quite noticeable in Movies 4b and 4c at the peaks and valleys. The red and green snakes look rather similar, but the differences in performance show up clearly in the $RMSE$ plots.

In Movie 4c the Absolute $RMSE$ tells a very different story from the Relative $RMSE$. This difference is a result of the *bias*, which crosses 0 when the target curve crosses the center, at its inflection points. At these points, the bias decreases and thus so does the $RMSE$. Thus the Absolute $RMSE$ semicircles have large size changes which follow the oscillations of the target semicircle in the snake window, while the Relative $RMSE$ keeps a roughly constant size.

4.3 Comparison of kernels

In this section we discuss the impact of the shape of the kernel. Theoretical results, dating back at least to Epanechnikov (1969) have shown that kernel shape has a secondary effect on the performance of smoothing methods. But when kernels are implemented using their usual definitions, the same bandwidth can represent rather different amounts of smoothing, as shown in Section 4.3.1.

A recipe for kernel rescaling, which allows use of the same bandwidth to represent the same amount of smoothing, was named the "canonical rescaling" by Marron and Nolan (1989). The effect of this is shown in Section 4.3.2. The canonical kernel viewpoint also provides a clear and simple solution to the problem of "best" kernel shape.

The three kernels considered in our movies are the Triweight:

$$K(x) = \frac{35}{32}\left(1 - x^2\right)^3 1_{[-1,1]}(x),$$

where $1_{[-1,1]}$ denotes the indicator function of the interval $[-1, 1]$, the Uniform:

$$K(x) = \frac{1}{2} 1_{[-1,1]}(x),$$

and the Gaussian (Standard Normal):

$$K(x) = (2\pi)^{-1/2} e^{-x^2/2}.$$

Exactly these version are used in Section 4.3.1.

4.3.1 Movie 5a, ordinary scaling

In order to show an overall moderate amount of smoothing, the same bandwidth $h = 0.4$ is used in each window. Because of the different kernels used in each window, each snake traces a unique path. The snake in the Triweight kernel window traces a more variable path. The bends are highly visible on the right side of the window because the of increased variability in the data there. However, the Uniform kernel has "fine scale variability." The magnitude of the bends of the Uniform kernel is less than the magnitude of the high variability bends of the Triweight. This occurs because of "kernel edge effects," caused by points being either completely inside, or completely outside the uniform window, while the change is more gradual for the other kernels. The bends in the smooth for the Uniform kernel are of smaller magnitude than for the Triweight kernel. The Gaussian kernel differs from both the Triweight and the Uniform kernels in its increased amount of smoothness. This smoothness is the result of the larger effective bandwidth

These differences in the kernels cause similar changes in the snake motions. The yellow Triweight snake does large scale dancing, and the green Uniform snake does small scale dancing. However, the red Gaussian snake is very stable.

These differences also drive the order of the $RMSE$ plots. The bias is largest for red and smallest for yellow, with green in between the two. The sd has the opposite order, with the yellow as the largest. Based on the evidence from Movie 5a the same numerical value of the bandwidth results in a different amount of smoothing for ordinary kernels.

4.3.2 Movie 5b, canonical scaling

Using Table 1 of Nolan and Marron (1989), the canonical rescalings of these three kernels are all of the form

$$K_\delta(x) = \frac{1}{\delta} K\left(\frac{x}{\delta}\right),$$

where $\delta = \left(\frac{9450}{143}\right)^{1/5} \approx 2.3122$ for the Triweight, $\delta = (9/2)^{1/5} \approx 1.3510$ for the Uniform, and $\left(\frac{1}{4\pi}\right)^{1/10} \approx 0.7764$ for the Gaussian.

Except for the kernel rescalings, the setting of this movie is the same as Movie 5b, with even the same realization of the data. Many of the same observations can be made about this movie and Movie 5a. The three estimates represent the same amount of smoothing in both movies. Because of the bin edge effect, the snake in the uniform window of this movie wobbles in the same manner as the snake in Movie 5b. As expected, the Triweight and Gaussian kernel windows in this movie share many of the same characteristics.

The snakes in this movie also reflect great similarities. The red Gaussian and yellow Triweight snakes are very nearly the same lines, and the green snake crosses the centerline very near the red and yellow. However, the green Uniform snake has more variability, mostly in slope, due to the kernel edge effect

As anticipated from the theory, the $RMSE$ behavior for the three kernels is now very similar. In particular, the *bias* and *sd* trade-offs are the same. The only substantial difference occurs at the edges, where the green snake is slightly outside, as predicted theoretically. Based on the evidence above, canonical kernels, with the same numerical value of h, result in the same amount of smoothing.

4.4 Comparison of sample sizes and sigmas

This section explores the relationship between the underlying signal and the noise that is added. In Section 4.4.1 everything is the same, except the sample size n is different in the three comparison windows, to illustrate how decreasing difficulty of the estimation process impacts the various aspects of smoothing. Section 4.4.2 shows similar lessons where the setting is the same, but there is increasing difficulty of estimation because the noise level σ increases. In Section 4.4.3 the noise level σ is increased at the same time as the sample size n is decreased. This is done so that the signal to noise ratio, i.e. the difficulty of estimation, stays constant.

4.4.1 Movie 6a, different n

Although improved performance comes from decreasing h as n increases, the same bandwidth $h = .2$ is used in each window for presentation purposes. The snakes and the estimates for each window are rather similar, but the $n = 25$ yellow wobbles more than the other two sample sizes and does not get as close to the peak of the target as the $n = 400$ red snake, which is more stable.

In this movie, the $RMSE$ boxes are very informative. The biases are the same across n, which means the $RMSE$ boxes all share a common horizontal edge. This was expected since bias feels only the kernel, the bandwidth, and the true curve, and in particular is independent of n. The total $RMSE$ is thus determined by the sd, which is constant in the interior, and grows larger at each boundary. The smaller sample size $n = 25$ yellow has the larger value of sd, and thus of $RMSE$. Both sd and $RMSE$ are smaller for the $n = 100$ green and even smaller for the $n = 400$ yellow.

4.4.2 Movie 6b, different sigma

Again improved performance follows from increasing h as σ increases, but the same bandwidth $h = .2$ is used in each window for a more clear presentation. Movie 6a and this movie share many characteristics. As in the previous movie, the snakes are quite similar between the windows. However, the red $\sigma = 0.4$ snake, and its corresponding estimator $\widehat{m}_h(x)$ wiggles the most, especially on the right side of the window.

The $RMSE$ boxes are very similar to those in Movie 6a. In particular, the biases are again all the same. However, the sd is smaller for smaller values of σ. Therefore, the yellow $\sigma = 0.1$ snake reflects the best fit, and the red snake reflects the worst fit because of the increased amount of noise.

4.4.3 Movie 6c, signal to noise ratio fixed

The same bandwidth $h = 0.2$ is used in each window, which is sensible because the signal to noise ratio is the same. As expected, all three estimates appear extremely similar. However, upon careful observation, the red $n = 400$, $\sigma = 0.4$ estimate is slightly better, especially at the $x = 0$ peak. This can be explained as a discretization effect, as the data are quite sparse for the yellow $n = 25$, $\sigma = 0.1$ estimate.

The $RMSE$ boxes are also quite similar, so similar, that often only the red semicircle is observed because the yellow and green lie underneath. The exception to this phenomenon is the end points. Here the yellow estimate seems slightly worse than the others do. This is perhaps due also to discretization effects. See Jones (1989), Hall and Wand (1996) and Gonzalez - Manteiga, Sanchez - Sellero, Wand (1996) for mathematical quantification of discretization effects in smoothing.

4.5 Comparison of X designs

In this section, we study the effect of different designs, i.e. different configurations of the X_i.

4.5.1 Movies 7a, 7b, random designs

The left window uses a decreasing design density, where the X_i are random, and independent, identically distributed as Beta($\frac{3}{2}$,1), i.e. they have density $f(x) = (3/2)x^{1/2}1_{(0,1)}(x)$. The central window uses a Uniform design, i.e. the X_i are drawn from the density $1_{(0,1)}(x)$. The right window uses design points from the Beta(1,$\frac{3}{2}$) distribution, i.e. they have density $f(x) = (3/2)(1-x)^{1/2}1_{(0,1)}(x)$, which increases in x.

In Movie 7a a bandwidth of $h = 0.3$ is used, and the snake is a polynomial of degree $p = 2$. Many of the features of this movie reflect the lessons learned above, after realizing that there are more data in some locations, and less in others. The important new lesson is about "design bias", i.e. bias that can arise from nonuniform design points $X_1, ..., X_n$, that arises for even values of p in interior regions. This has been asymptotically analyzed, e.g. in Section 5.4 of Wand and Jones. This can be seen in Movie 7a, by stopping the movie near $x = -0.9$. Note that the yellow and green estimates have positive bias (see the $RMSE$ box), which is expected since the target is convex here. However, the red curve has negative bias. This is caused by the red estimate using more data to the right than to the left. The same unusual effect happens in the opposite direction for the yellow estimate, near $x = 0.9$.

The asymptotic theory predicts this problem will disappear for odd values of p. This is investigated in Movie 7b. Occasional sign differences in the

bias are still visible, but they are much smaller in magnitude, and happen when all of the biases are smaller.

4.5.2 Movie 7c, fixed vs. random

The left window has $n = 50$ uniform random X_i, and error standard deviation $\sigma = 0.2$. The center window has $n = 50$ equally spaced X_i, and the same error standard deviation σ. The right window is a random design with many more data points, $n = 450$, but the estimation setting has the same signal to noise ratio, since $\sigma = 0.6$.

In Movie 7c a bandwidth of $h = 0.3$ is used, and the snake is linear, i.e. degree $p = 1$. The three curve estimates are qualitatively similar, i.e. they represent the same amount of smoothness. However, the three estimates and snakes are not directly comparable because they come from different realizations of the data.

Some of the observations concerning the $RMSE$ displays were unexpected. For instance, because a random design is less efficient than equally spaced, the $RMSE$ for the yellow snake is expected to be larger than for the green, and this usually occurs in this $RMSE$ display, except near $x = -1.2$ and $x = 0.8$. However, because there are "clusters in the random X_i", i.e. the local empirical density is high, the yellow $RMSE$ is smaller near those points since there is more information in the data. This occasional superior performance comes from the fact that $RMSE$ is calculated conditionally on $X_1, ..., X_n$. The unconditional $RMSE$ has not been computed here, but we believe the green will be always smaller than the yellow. This example highlights the difference between the conditional and unconditional $RMSE$.

Although there were some unexpected observations, other observations occurred as expected. For instance, usually for larger n, there should be less difference between equally spaced and random uniform designs. In Movie 7c this appears as the $RMSE$ for the green equally spaced design being very similar to the red large n random design. A fine point is that the red semicircle is usually just outside the green because there is still some small inefficiency of the random design compared to the equally spaced.

4.6 Comparison of scedasticity

In this section we study how changing noise levels affects the smoothing process.

4.6.1 Movie 8a, monotone

In this section the noise level σ_i is different at different locations. The variance curves used here are shown in Figure 4.

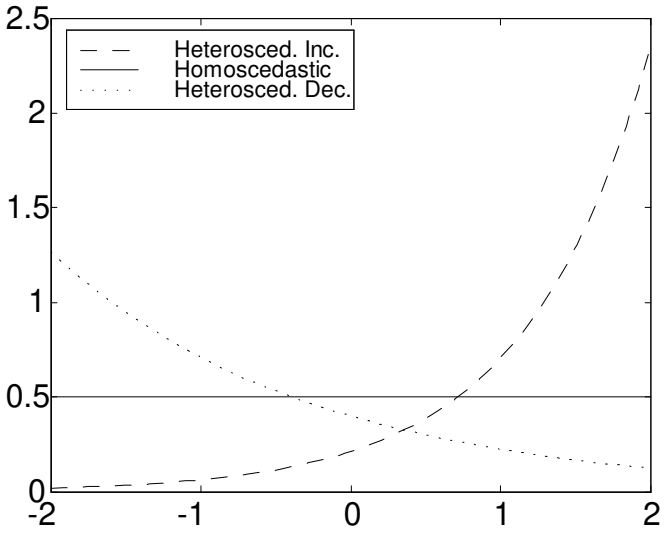

FIGURE 4: *Standard deviation functions used in Movie 8a.*

Movie 8a uses a bandwidth of $h = 0.35$, and the snake is a polynomial of degree $p = 3$. The $RMSE$ displays in Movie 8a are driven by the differences in the *sd*. The *bias* remains the same, but the left and right sides of the window show great differences among the *sd* values. On the left the *sd* falls in the following order

$$\text{yellow} < \text{green} < \text{red},$$

and the exact opposite ordering of the *sd* values occurs on the right side of the window. In the center all the snakes are similar, but the green snake is slightly worse because its variance is higher, as shown in Figure 4. The red snake does not appear in the same way on both ends because as seen in Figure 4, the increasing heteroscedasticity goes higher on the right than the decreasing one does on the left.

5 Description of programs

The Matlab software used to generate these examples is available from the link near the bottom of the web page listed in Section 4. All files in the directory should be downloaded, probably to a separate directory. From Matlab, a menu screen can be started using the command

```
» nprmov1
```

This menu has push-buttons to start each movie described here, and some control parameters as well. For custom movie choice, the Matlab function `mainfig1` can be used for a much wider array of movies. The command

```
» help mainfig1
```

gives details on the use of this function. Most of the remaining files at that web address are sub function files, which allow choice from an array of kernels, scedasticities, target curves and x - designs. Other options can be simply developed by copying those available and editing them.

5.1 Details of calculations

In the j-th movie frame, estimation is performed at a point x_j, shown as the vertical thin cyan line. For smooth viewing, the x_j are taken to be equally spaced, and thus can be different from the data locations, X_i.

At each centerpoint x_j, the bias is calculated as:

$$bias(x_j) = E\widehat{m}_h(x_j) - m(x_j),$$

where $E\widehat{m}_h(x_j)$ is the expected value of $\widehat{m}_h(x_j)$. As shown for example in Section 5.2 of Wand and Jones (1995), the local polynomial smoother is a linear estimator, $\widehat{m}_h(x_j) = \sum_{i=1}^n W_i(x_j)Y_i$, for suitable weights $W_i(x_j)$. Thus $E\widehat{m}_h(x_j)$ is calculated as $E\widehat{m}_h(x_j) = \sum_{i=1}^n W_i(x_j)m(x_i)$..

Also at each centerpoint x_j, using the formula for variance of a linear operator, the standard deviation is :

$$sd(x_j) = \sqrt{\sum_{i=1}^{n} W_i(x_j)^2 \sigma_i},$$

where $\sigma_i = var(Y_i)^{1/2}$, $i = 1, ..., n$.

References

[1] Bowman, A. W. and Azzalini, A. (1997) *Applied Smoothing Techniques for Data Analysis, the Kernel Approach with S-plus Illustrations*, Clarendon Press, Oxford.

[2] Cleveland, W. S. (1979) Robust locally weighted regression and smoothing scatterplots, *Journal of the American Statistical Association*, 74, 829-836.

[3] Cleveland, W. S. and Devlin, S. J. (1988) Locally weighted regression: an approach to regression analysis by local fitting, *Journal of the American Statistical Association*, 83, 596-610.

[4] Cleveland, W. S., Grosse, E. and Shyu, W. M. (1992) Local regression models, *Statistical Models in S*, (J. M. Chambers, T. J. Hastie, eds.), Chapman and Hall, New York, 309-376.

[5] Cleveland, W. S. and Loader, C. (1996) Smoothing by local regression: principles and methods, in *Statistical Theory and Computational Aspects of Smoothing*, Härdle, W. and Schimek, M. G. eds., Physica Verlag, Heidelberg, 10-50.

[6] Epanechnikov, V. A. (1969) Nonparametric estimation of a multivariate probability density, *Theory of Probability and Its Applications*, 14, 153-158.

[7] Fan, J. (1992) Design adaptive nonparameteric regression, *Journal of the American Statistical Association*, 87, 998-1004.

[8] Fan, J. (1993) Local linear regression smoothers and their minimax efficiencies, *Annals of Statistics*, 21, 196-216.

[9] Fan, J. and Gijbels, I. (1996) *Local Polynomial Modelling and Its Applications*, Chapman and Hall, London.

[10] Gonzalez - Manteiga W., Sanchez - Sellero C., Wand, M. P. (1996) Accuracy of binned kernel functional approximations, *Computational Statistics and Data Analysis*, 22, 1-16.

[11] Hall, P. and Wand, M. P. (1996) On the accuracy of binned kernel density estimators, *Journal of Multivariate Analysis*, 56, 165-184.

[12] Hastie, T. and Loader, C. (1993) Local Regression: Automatic Kernel Carpentry, with discussion, *Statistical Science*, 8, 120-143.

[13] Jones, M. C. (1989) Discretized and interpolated kernel density estimates, *Journal of the American Statistical Association*, 84, 733-741.

[14] Marron, J. S. and Nolan, D. (1989) Canonical kernels for density estimation, *Statistics and Probability Letters*, 7, 195-199.

[15] Marron, J. S. and Wand, M. P. (1992) Exact mean integrated squared error, *Annals of Statistics*, 20, 712-736.

[16] Nadaraya, E. A. (1964) On estimating regression, *Theory of Probability and Its Applications*, 9, 141-142.

[17] Ruppert, D. (1997) Empirical-bias bandwidths for local polynomial nonparametric regression and density estimation, *Journal of the American Statistical Association*, 92, 1049-1062.

[18] Seifert, B. and Gasser, T. (1996) Variance properties of local polynomials and ensuing modifications, in *Statistical Theory and Computational Aspects of Smoothing*, Härdle, W. and Schimek, M. G. eds., Physica Verlag, Heidelberg, 51-80..

[19] Simonoff, J. S. (1996) *Smoothing Methods in Statistics*, Springer Verlag, New York.

[20] Stone, C. J. (1975) Nearest neighbor estimators of a nonlinear regression function, *Proc. Computer Science and Statistics, 8th Annual Symposium on the Interface*, 413-418.

[21] Stone, C. J. (1977) Consistent nonparmetric regression, with discussion, *Annals of Statistics*, 5, 549-645.

[22] Wand, M. P. and Jones, M. C. (1995) *Kernel Smoothing*, Chapman and Hall, London.

[23] Watson, G. S. (1964) Smooth regression analysis, *Sankhyā, Ser. A*, 26, 359-372.