Clemson University
TigerPrints

All Theses | Theses

8-2012

Local Polynomial Regression with Application to Sea Surface Temperatures

Michael Finney
Clemson University, finney2@clemson.edu

Follow this and additional works at: https://tigerprints.clemson.edu/all_theses

 Part of the Applied Mathematics Commons

Recommended Citation
Finney, Michael, "Local Polynomial Regression with Application to Sea Surface Temperatures" (2012). *All Theses*. 1473.
https://tigerprints.clemson.edu/all_theses/1473

This Thesis is brought to you for free and open access by the Theses at TigerPrints. It has been accepted for inclusion in All Theses by an authorized administrator of TigerPrints. For more information, please contact kokeefe@clemson.edu.

LOCAL POLYNOMIAL REGRESSION WITH APPLICATION TO SEA SURFACE TEMPERATURES

A Masters Thesis
Presented to
the Graduate School of
Clemson University

In Partial Fulfillment
of the Requirements for the Degree
Masters of Science
Mathematics

---

by
Michael Thomas Finney
August 2012

Accepted by:
Dr. Robert Lund, Committee Chair
Dr. Karunarathna B. Kulasekera
Dr. Colin Gallagher
Dr. Peter Kiessler

Abstract

Our problem involves methods for determining the times of a maximum or minimum for a general mean function in time series data. The methods explored here involve polynomial smoothing. In theory, the methods calculate a general number of derivatives of the estimated polynomial. Using these techniques, we wish to find a balance between error, variance, and complexity and apply it to a time series of sea surface temperatures. We will first explore the theory behind the method and then find a way to optimally apply it to our data.

Table of Contents

List of Figures

Chapter 1

Introduction

The sea surface temperature data (shown in Figure 1) was given to us by Julien Emile-Geay at the University of Southern California and during the summer of 2011. Our goal is to find the concise times of maximum and minimum of the mean value function in hopes that our mathematics would help close the gap on the estimation of when the last ice age began/ended. I was commissioned to research advanced density estimation methods that would give mathematically precise estimations of a derivative. In this paper, we will explore the foundations behind our method, an explanation and analysis of local polynomial smoothing, and our methods' application to our data.

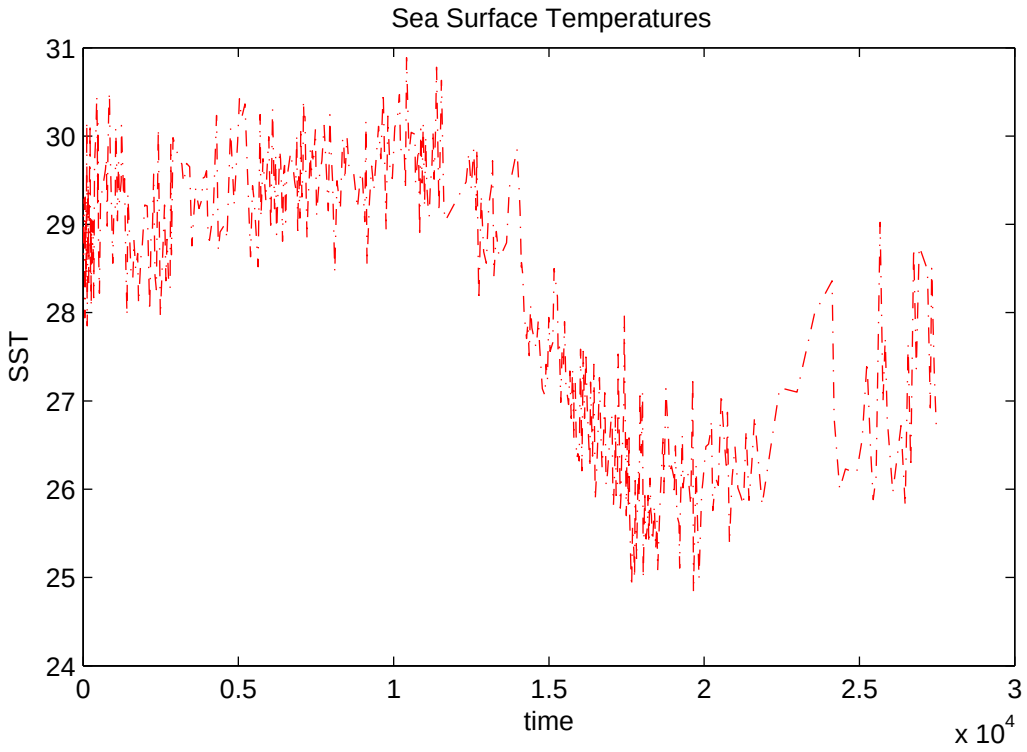

Figure 1.1: Sea surface temperatures as a function of time. Note that larger times are furthur from the present, and the temperatues are in degrees Centigrade

Chapter 2

Smoothing and Kernel Functions

To understand local polynomial regression, we first review smoothing parameters and kernel estimation.

1. Smoothing

The idea behind smoothing permeates many regression problems. Given noisy data, we desire to extract general structure in the observations without following local fluctuations due to noise. Good smoothing methods strike a balance between error and bias.

In smoothing, we typically use the variables h or λ to quantify the amount of "smoothing". Smoothing here is done via a kernel function, which we will explain below. The selection of h, often called the "bandwidth," is complicated and has no optimal answer outside of theoretical situations. Depending on how one measures error, different ideal h-values emerge. Regardless, each ideal value for h depends upon the sample size n and specific aspects of the problem. For instance, suppose one measures accuracy via the mean integrated squared error, or

$$E\left[\int_{-\infty}^{\infty}(\hat{m}_h(x) - m(x))^2 dx\right]. \tag{2.1}$$

Here, $m(t) = E[X(t)]$, $\{X(t)\}$ is a time series, and $\hat{m}_h(t)$ is an estimate of $m(t)$ with bandwidth h. Evaluating this integral and performing a detailed asymptotic anaysis shows that as $n \to \infty$, the

best bandwidth satisfies

$$h = K_2^{-2/5} \left(\int_{-\infty}^{\infty} K(t)^2 dt \right)^{\frac{1}{5}} \left(\int_{-\infty}^{\infty} m''(x)^2 dx \right)^{-\frac{1}{5}} n^{-\frac{1}{5}}, \tag{2.2}$$

where K_2 is some constant, and $K(\cdot)$ is our kernel function (more below). We usually simplify the optimal value to $cn^{-\frac{1}{5}}$ where c is a constant. Other optimal values of h are of a similar fashion (Silverman 1986).

Since the optimal value of h depends on the true mean function m, we can never, in truth, know the best value of h. It is not hard, however, to determine bad values for h. Letting $h \downarrow 0$ brings us closer to an interpolation of the data, and as $h \to \infty$, our fit slowly becomes a constant function (or polynomial of degree n if using local polynomial regression) (Fan and Gijbels 1996).

Firgure 2-4 demonstrate how different values of h influence smoothing. The data is simulated from the equation $m(x) = x^3 - 7x^2 + 10x$ over $x \in [0, 5.5]$. Here, random white noise with zero mean and variance 10 was added at each point.

Firgure 2 shows a smoothing parameter that is relatively close to zero ($h = .05$).

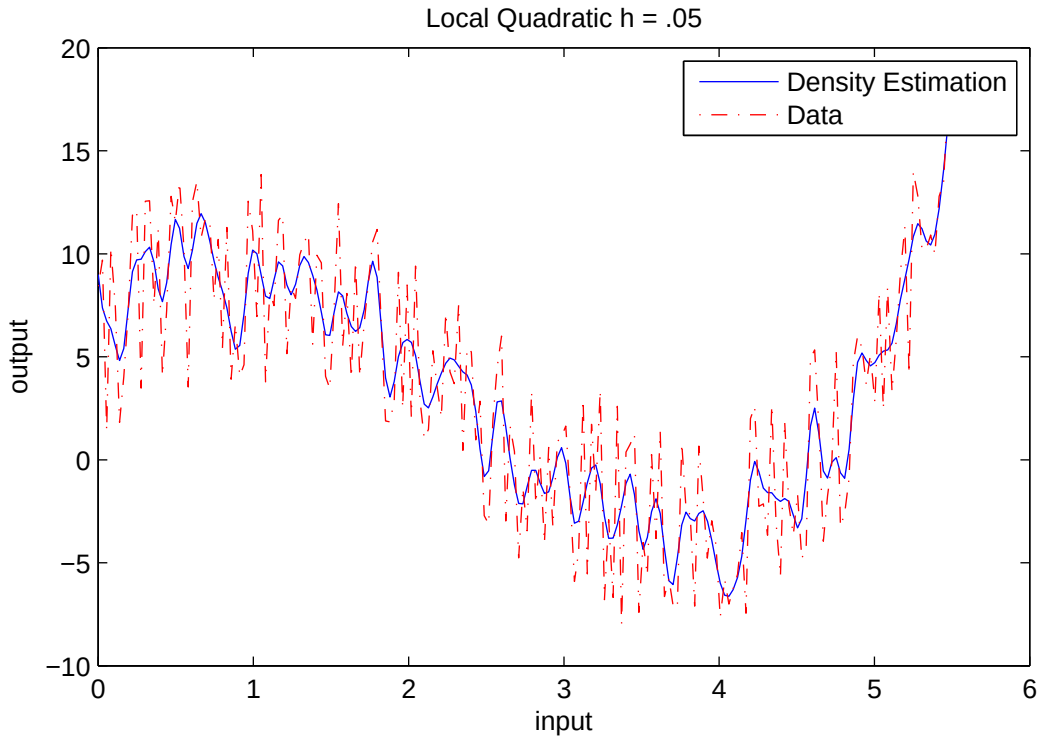

Figure 2.1: $m(x) = x^3 - 7x^2 + 10x$ with random noise smoothed with h=.05

Figure 3 shows a smoothing parameter that is far too big for the data ($h = 500$), and our regression appears to be the best quadratic fit for the data (local quadratic regression was used, so this is appropriate).

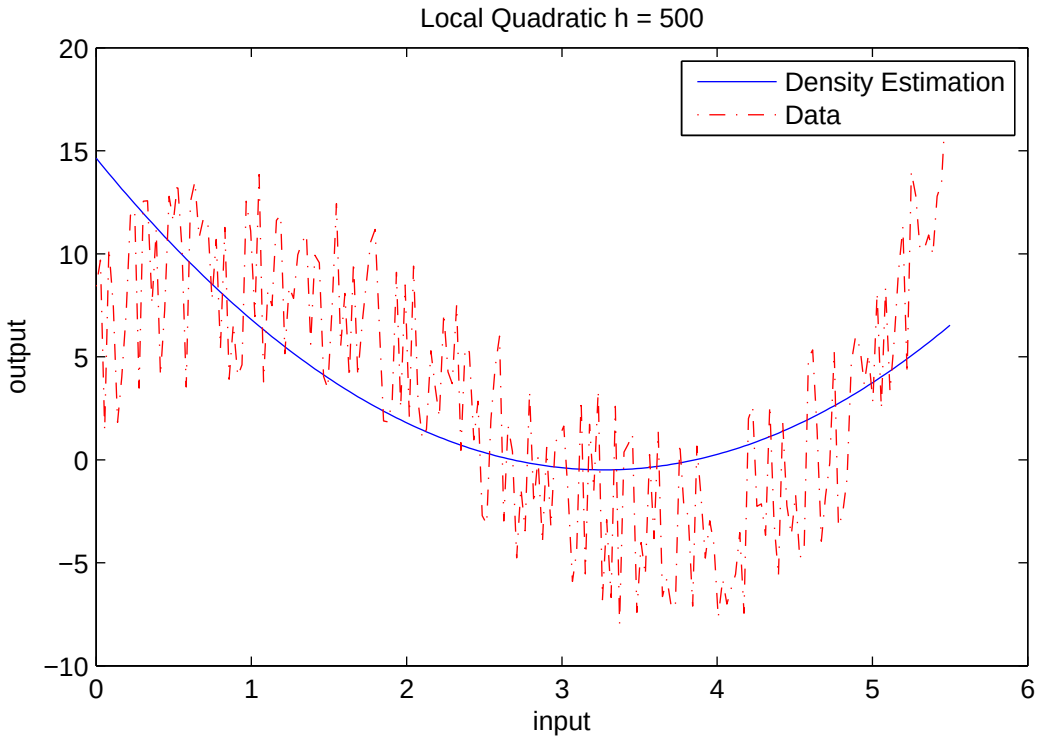

Figure 2.2: $m(x) = x^3 - 7x^2 + 10x$ with random noise smoothed with h=500

Figure 4 shows a density estimator for h=1, which gives visually appealing results.

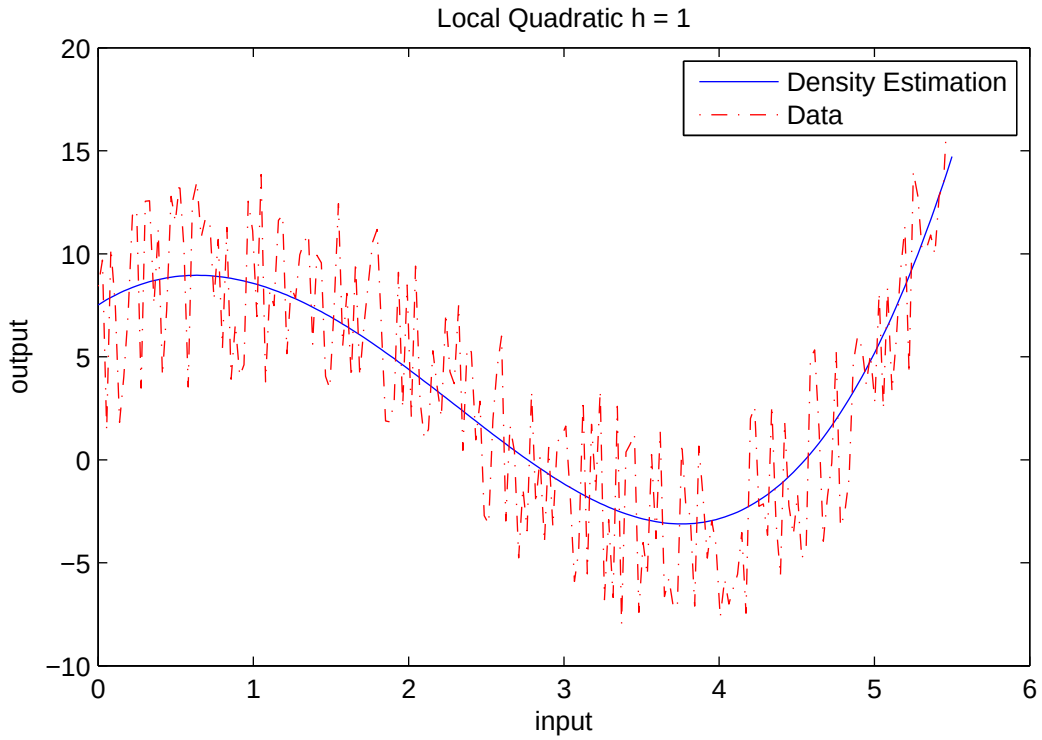

Figure 2.3: $m(x) = x^3 - 7x^2 + 10x$ with random noise smoothed with h=1

2. Kernel functions

An assumption posited in smoothing problems is that $E[X(t)]$ depends on t. We also assume a finite error variance, so there is natural variation in the data. To make calculations easier, we assume that all observations are spaced equally. In other words, we say that for the observation at time t_j, our estimate has form

$$\hat{m}(t_j) = \sum_{i=1}^{n} w_i y(t_i), \tag{2.3}$$

where $y(t_i)$ is the observation at time t_i and w_i is a weight function which quantifies how dependent $\hat{m}(t_j)$ is on $y(t_i)$.

It is natural to consider that as the space between observations increases, the dendence between the observations decrease as well. We wish for our weights to reflect this. This is where

we select our weights according to a *kernel* function. A kernel function is a zero mean symmetric probability density function that concentrates most of its weight around its center and whose mass either is negated or decays exponentially as the distance from its center increases (Ramsay and Silverman 2006). Examples of Kernel functions include:

· Uniform: $K(t) = .5, |t| \leq 1$ (0 otherwise)

· Quadtratic: $K(t) = .75(1 - u^2), |t| \leq 1$ (0 otherwise)

· Gaussian: $K(t) = (2\pi)^{-1/2} \exp(-t^2/2)$

For the problems considered here, the influence of our kernel function for a given value t will be $(t - t_j)/h$, where h is our smoothing parameter. One can see how h affects our estimate: the value of h determines how many times t_j belong to a ball of radius 1 around our estimate time, t.

Now with our knowledge of the kernel function and smoothing parameters, we rewrite our estimate as

$$\hat{m}(t) = \sum_{j=1}^{n} K\left(\frac{t - t_j}{h}\right) y_j. \tag{2.4}$$

Chapter 3

Local n-th degree polynomial regression

To find the derivatives of our estimate, we examine the Taylor expansion of $m(t)$, say

$$m(t) = \sum_{j=0}^{\infty} \frac{m^{(j)}(x)}{j!}(t-x)^j$$

for some x. If we truncate this expansion, then we have that, for a neighborhood around t and some positive integer p,

$$m(t) \approx \sum_{j=0}^{p} \frac{m^{(j)}(x)}{j!}(t-x)^j.$$

So if we create an estimate of the form

$$\hat{m}(t) = \sum_{j=0}^{p} \theta_j (t-x)^j, \tag{3.1}$$

then

$$\hat{m}^{(k)}(x) = k!\theta_k, \tag{3.2}$$

where $m^{(k)}(x)$ is the estimate of the kth derivative of the mean function. To find the best estimate of θ_j, we look to minimize the local sum of squares

$$\sum_{i=1}^{n} K_i(t)[y_i - \sum_{j=0}^{p} \theta_j (t-t_i)^j]^2 \tag{3.3}$$

where $K_i(t) = K\left(\frac{t-t_i}{h}\right)$. Taking derivatives with respect to θ_j for all j and equating to zero yields a system of $p+1$ equations with $p+1$ variables. A solution to this system of equations is found by solving

$$A\theta = b, \tag{3.4}$$

where

$$A = \left[\frac{\sum_i K_i(t)(t-t_i)^{k+l-2}}{\sum_i K_i(t)(t-t_i)^{2(k-1)}}\right]_{k,l\in\{1,\ldots,p+1\}}, \quad \theta = [\theta_j]_{j\in\{0,\ldots,p\}}, \tag{3.5}$$

and

$$b = \left[\frac{\sum_i K_i(t)(t-t_i)^{l-1} y_i}{\sum_i K_i(t)(t-t_i)^{2(l-1)}}\right]_{l\in\{1,\ldots,p+1\}}. \tag{3.6}$$

1. Bias and Variance

To calculate the bias of θ, set

$$T = \begin{bmatrix} \frac{1}{\sqrt{\sum_i K_i(t)}} & \frac{1}{\sqrt{\sum_i K_i(t)}} & \cdots & \frac{1}{\sqrt{\sum_i K_i(t)}} \\ \frac{(t-t_1)}{\sqrt{\sum_i K_i(t)(t-t_i)^2}} & \frac{(t-t_2)}{\sqrt{\sum_i K_i(t)(t-t_i)^2}} & \cdots & \frac{(t-t_n)}{\sqrt{\sum_i K_i(t)(t-t_i)^2}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{(t-t_1)^p}{\sqrt{\sum_i K_i(t)(t-t_i)^{2p}}} & \frac{(t-t_2)^p}{\sqrt{\sum_i K_i(t)(t-t_i)^{2p}}} & \cdots & \frac{(t-t_n)^p}{\sqrt{\sum_i K_i(t)(t-t_i)^{2p}}} \end{bmatrix}_{p+1\times n} \tag{3.7}$$

and

$$K = \begin{bmatrix} K_1(t) & 0 & \cdots & 0 \\ 0 & K_2(t) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & K_n(t) \end{bmatrix}_{n\times n}. \tag{3.8}$$

Then note that $A = TKT'$ and $b = TKy$ where $y = [y_1, y_2, \ldots, y_n]'$

Given this, the expectation for our estimator $\hat{\theta}$ given our true values θ is

$$E[\hat{\theta}] = \theta + A^{-1}TKr, \tag{3.9}$$

where $r = E[y] - T\theta$. The variance is

$$Var(\hat{\theta}) = A^{-1}TK\Sigma KT'(A^{-1})', \tag{3.10}$$

where $\Sigma = \text{diag}\{\sigma^2(y_i)\}$. A proof for these is given in the appendix.

Chapter 4

Our data

We now look at the application of local polynomial regression to a set of sea surface temperature data. The main complication is that our data points are not equally spaced. This causes some problems which we will address as they arise.

Our series has 453 data points. For our density estimate, 500 equally spaced input points were chosen between the lowest input value and our highest. A smoothing parameter of $h = 900$ was chosen, rather arbitrarily (the maximum spacing between input values was 678, while a standard deviation between the spacing was 68.9583, so I added the maximum and three standard deviations to get roughly 900). For computations, we choose a Gaussian kernel; specifically,

$$K_i(t) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}\left(\frac{t - t_i}{h}\right)^2\right). \tag{4.1}$$

An important issue now surfaces: since we do not know a theoretical polynomial order, we must choose a legitimate value.

1. Choosing the polynomial order

Choosing the number of derivatives to calculate is a balancing act. With a higher degree polynomial, our estimated function has more information. Of course, when calculating the estimates of θ, error is involved. Indeed, the denominator for every element of A is $\sum_i K_i(t)(t - t_i)^{2(k-1)}$, which could cause a number of problems.

We fix the smoothing parameter at $h = 900$. First, as a control, we look at the data smoothing using a zero-degree polynomial. This special case is known as the Nadaraya-Watson kernel estimate.

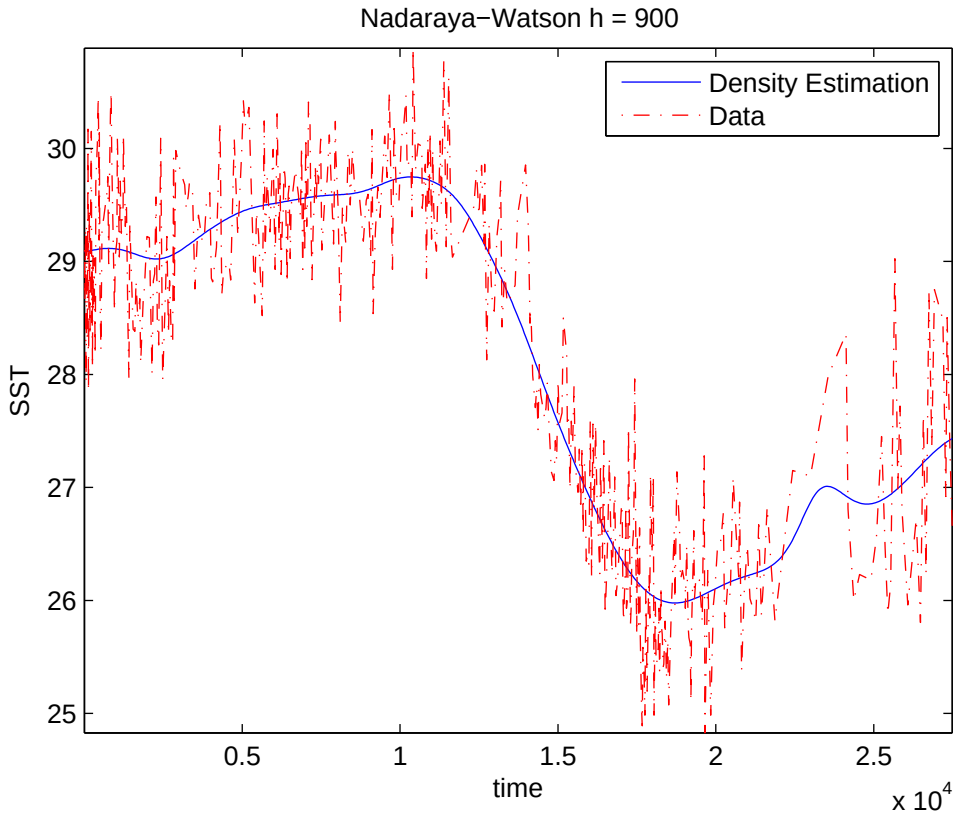

Figure 4.1: Sea surface temperatures using 0-degree polynomial regression

Since we are working with a one by one matrix, there is no condition number for our esimate in Figure 5. Therefore, we can assume (contingent upon a good selection of h) that the computations are stable. However, the estimated regression has no derivatives, which defeats the purpose of the excercise.

Next, we look at a local linear regression.

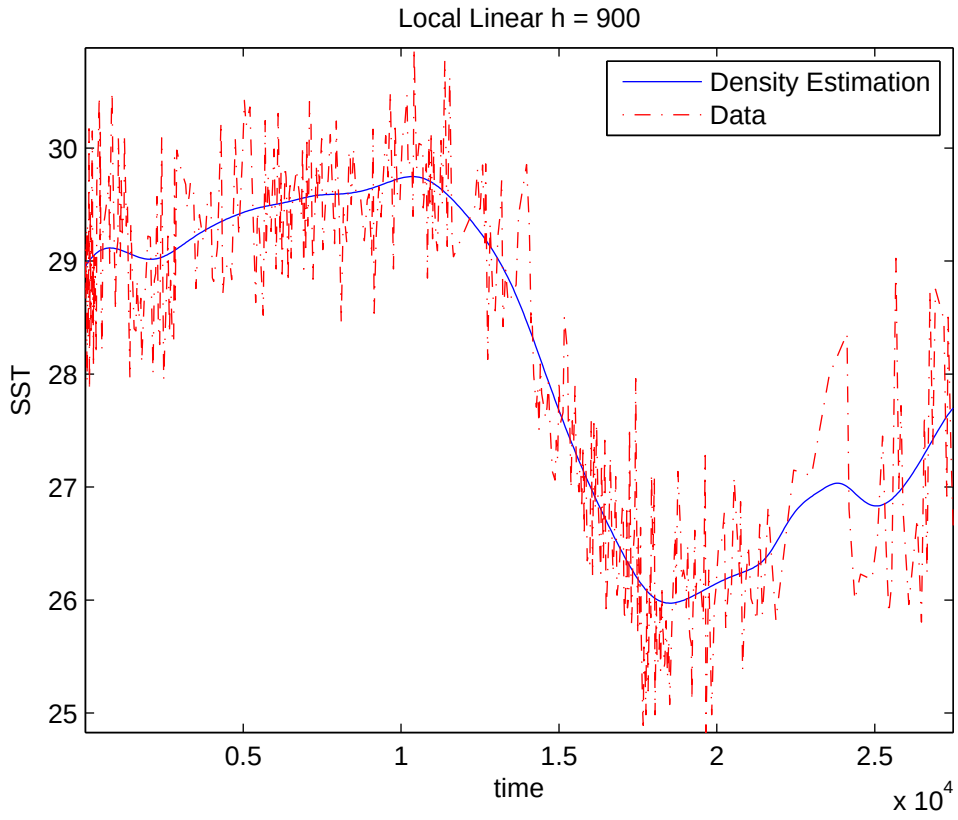

Figure 4.2: Sea surface temperatures using 1-degree polynomial regression

In Figure 6, our maximum condition number is 8.4596e+005. During calculation of the parameter $\hat{\theta}_0$, a parameter estimating the derivative at every point is calculated as a by-product. Though we are able to calculate derivatives now, we may produce a better fit with a local quadratic regression.

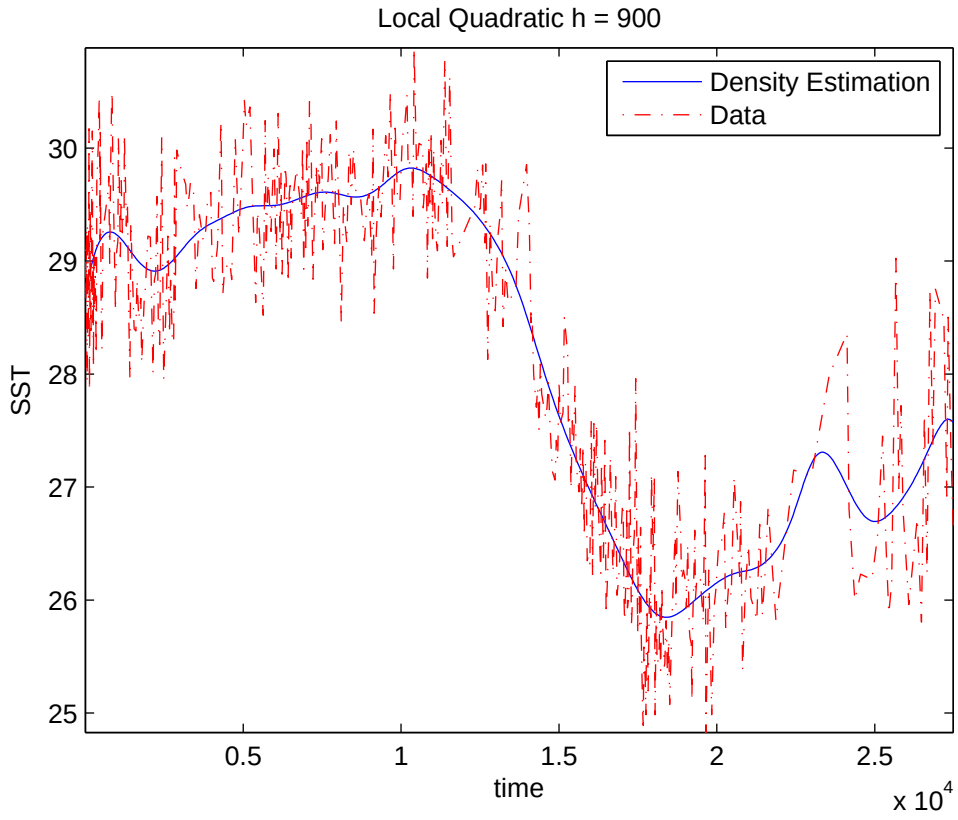

Figure 4.3: Sea surface temperatures using 2-degree polynomial regression

In Figure 7, our condition number is 5.3778e+012, so the method is becoming less reliable. For a cubic regression, the methods break down as evidenced in Figure 8.

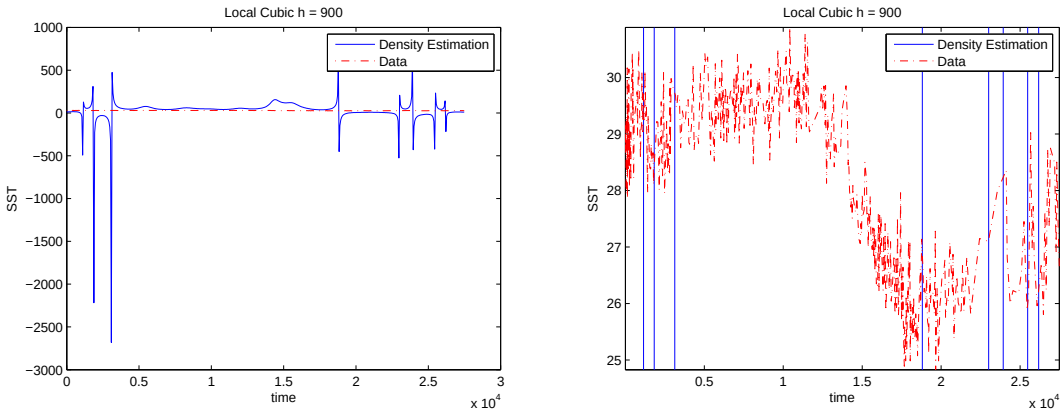

Figure 4.4: Sea surface temperatures using 3-degree polynomial regression scaled to fit and then rescaled to previous scales

Our maximum condition number from Figure 8 is 1.4308e+018, meaning that all of our calculations are completely unreliable.

Clearly, calculations for any degree above three will be even more unreliable. Our decision now is between a local linear or a local quadratic estimator. Given the condition numbers presented, a local linear estimator seems preferable. However, the solutions were solved using MATLAB's backslash operator, which is famous for being able to work with ill-conditioned matricies. Indeed, the maximum entry of the residual taken in the Euclidan norm, $\mathbf{r} = ||b - A\hat{\theta}||_2$, for our local linear estimation is 3.552e-015 while the maximum entry for the quadratic estimation is virtually the same. Both of these can be attributed to machine error. Luckily, Fan and Gijbels (1996) have a theorem to help us decide the best degree (1996).

If p is the degree of our polynomial, and we wish to calculate the v-th derivative, then the asymptotic conditional bias when p-v is even is greater than the asymptotic conditional bias when p-v is odd.

For the details of this theorem, please see Fan and Gijbels (1996, pg 61-62). Given Theorem 4.1, it becomes clear why our graph of the local linear regression is prettier than the graph for local quadratic regression. This also means, though, that for calculating derivatives, we want to use local quadratic regression.

2. Calculating derivatives

First we look at a control to see how well our method can calculate derivatives. Looking at our equation from earlier, $m(x) = x^3 - 7x^2 + 10x$, we have a derivative of $m'(x) = 3x^2 - 14x + 20$ whose roots are $(14 \pm \sqrt{76})/6 \approx 3.7863, 0.8804$. In the graph below, the calculated points where the derivative changes signs are circled and linked together.

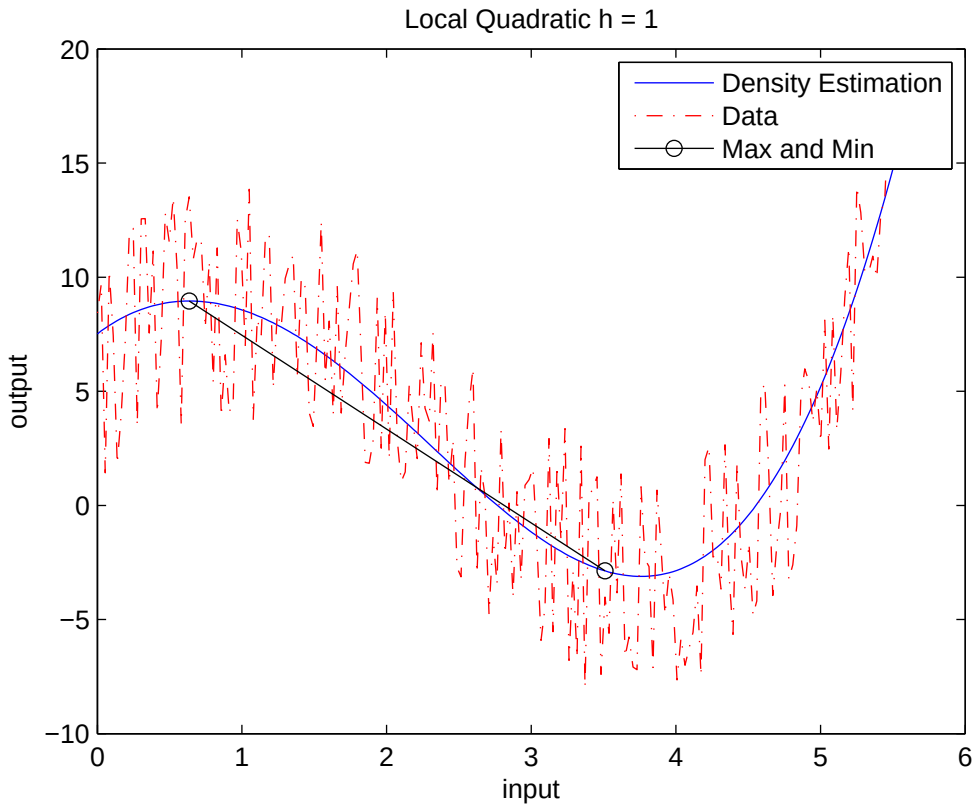

Figure 4.5: $m(x) = x^3 - 7x^2 + 10x$ with random noise smoothed with $h = 1$ and extreme points circled

The values given in this graph are $x = 3.5101, 0.6357$ leaving an absolute error of 0.2762 and 0.2447, respectively. We can attribute this to the natural bias in the method and/or computational error.

Now, running the program on our sea surface temperature data, produces Figure 10:

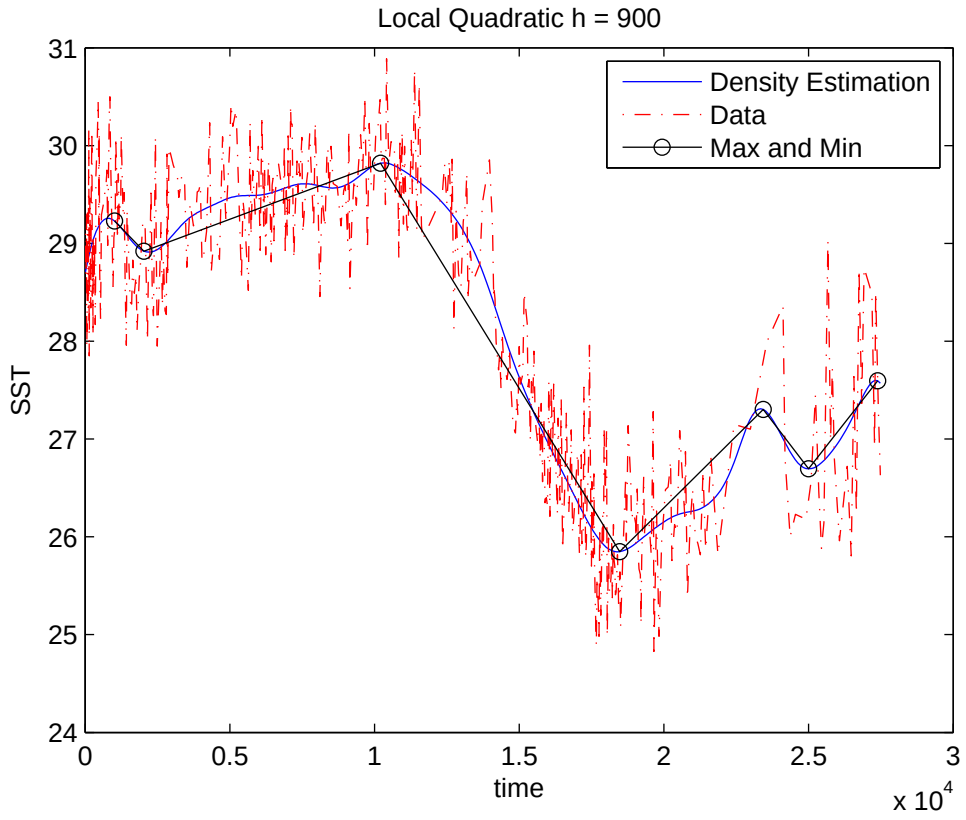

Figure 4.6: Sea surface temperatures using two-degree polynomial regression and extreme points circled

Figure 10 gives us our absolute maximum at $t = 4.2978e + 003$ and an absolute minimum at $t = 9.7555e + 003$. We have uncovered our desired result.

Chapter 5

Discussion

At the beginning of the paper, we established that our goal was to find a 'precise' maximum and minimum for the mean value function. One could argue that since we found two exact numbers that our goal was reached. However, the *accuracy* of these estimation leaves something to debate. There are many issues involving how reliable our esimates are.

1. Errors in calculating zeros of the derivative

The most glaring source of error is that our method approximates continuous data with discrete data. We require finding the zeros of our derivative, but the best we can find is the interval where our derivative changes signs. This means that any number in the space of the interval could be zero. However, the error in calculating the maximum and minimum of our cubic equation was around .25 while the length of the interval between each point was only .0275. Of course, we are dealing with bias and computational error as well. This means that we have closed in on where the maximums and minimums are located, but we still do not have 'precise' estimates. When dealing with statistics, this is not uncommon. Furthur research would find confidence intervals for the times of maximums and minimums.

2. Problems with non-equally spaced data

Recalling Figures 5-7 (the Nadara-Watson, local linear, and local quadratic regression), the interval starting at time 22,000 and ending at time 25,000 only contains 8 data entries. On average, an interval of length 3,000 contains 49.4570 entries with this data. There are other occurences like this in the data, but for this interval, the temperatures are exceptionally high given the surrounding data. This gives a local maximum which may not accurately reflect the true mean function. Though this is not relavent towards our goal of finding the absolute maximum and minimum of this data set, it could lead to errors in similar problems or if this data is used for other reasons. A simple solution would be to have h vary with the data based on the spacing of points, making it so the kernel's cutoff is based off of the number entries as opposed to the length of the interval.

Appendix A

Proofs

Proof of equation 3.4: $\hat{\theta} = A^{-1}b$ is the solution to of the minimum of $\hat{m}(t) = \sum_{i=1}^{n} K_i(t)[y_i - \sum_{j=0}^{p} \theta_j (t-t_i)^j]^2$, where

$$A = \left[\frac{\sum_i K_i(t)(t-t_i)^{h+l-2}}{\sum_i K_i(t)(t-t_i)^{2(h-1)}} \right]_{h,l \in \{1,\ldots,p+1\}}, \quad \theta = [\theta_j]_{j \in \{0,\ldots,p\}}$$

$$b = \left[\frac{\sum_i K_i(t)(t-t_i)^{l-1} y_i}{\sum_i K_i(t)(t-t_i)^{2(l-1)}} \right]_{l \in \{1,\ldots,p+1\}}.$$

Taking a derivative of $\hat{m}(t) = \sum_{i=1}^{n} K_i(t)[y_i - \sum_{j=0}^{p} \theta_j (t-t_i)^j]^2$ with respect to θ_k and setting the result to zero gives

$$0 = \sum_{i=1}^{n} K_i(t) 2[y_i - \sum_{j=0}^{p} \theta_j (t-t_i)^j](t-t_i)^k,$$

which implies that

$$\sum_{i=1}^{n} K_i(t)\theta_k (t-t_i)^{2k} = \sum_{i=1}^{n} K_i(t)(t-t_i)^k [y_i - \sum_{j \neq k}^{p} \theta_j (t-t_i)^j].$$

Solving for θ_k gives

$$\theta_k = \frac{\sum_{i=1}^{n} K_i(t)(t-t_i)^k [y_i - \sum_{j \neq k}^{n} \theta_j (t-t_i)^j]}{\sum_{i=1}^{n} K_i(t)(t-t_i)^{2k}}.$$

We can rewrite this equation as

$$\theta_k + \frac{\sum_{i=1}^n K_i(t)(t-t_i)^k \sum_{j\neq k}^p \theta_j (t-t_i)^j}{\sum_{i=1}^n K_i(t)(t-t_i)^{2k}} = \frac{\sum_{i=1}^n K_i(t)(t-t_i)^k y_i}{\sum_{i=1}^n K_i(t)(t-t_i)^{2k}}.$$

Note that the right hand side is the $k+1$st element of b above. The left hand side is equal to

$$\frac{\sum_i K_i(t)(t-t_i)^k \theta_0}{\sum_i K_i(t)(t-t_i)^{2k}} + \frac{\sum_i K_i(t)(t-t_i)^{k+1} \theta_1}{\sum_i K_i(t)(t-t_i)^{2k}} + \cdots + \theta_k + \cdots + \frac{\sum_i K_i(t)(t-t_i)^{k+n} \theta_n}{\sum_i K_i(t)(t-t_i)^{2k}}$$

which is equal to $A_{k+1,1}\theta_0 + A_{k+1,2}\theta_1 + \cdots + A_{k+1,n+1}\theta_n$, which is the $k+1$st row of A multiplied by θ. Since this was for arbitrary k, this proves our result.

Proof of equation 3.9:

$$\begin{aligned}
E[\hat{\theta}] &= E[A^{-1}b] \\
&= E[A^{-1}TKy] \\
&= A^{-1}TKE[y] - \theta + \theta \\
&= A^{-1}TKE[y] - A^{-1}A\theta + \theta \\
&= A^{-1}TKE[y] - A^{-1}TKT'\theta + \theta \\
&= \theta + A^{-1}TK(E[y] - T'\theta) \\
&= \theta + A^{-1}TKr,
\end{aligned}$$

where $r = (E[y] - T'\theta)$.

Proof of equation 3.10:

$$\begin{aligned}
Var(\hat{\theta}) &= Var(A^{-1}b) \\
&= A^{-1}Var(b)(A^{-1})' \\
&= A^{-1}Var(TKy)(A^{-1})' \\
&= A^{-1}TKVar(y)KT'(A^{-1})' \\
&= A^{-1}TK\Sigma KT'(A^{-1})',
\end{aligned}$$

where $\Sigma = \text{diag}\{\sigma^2(y_i)\}$

Bibliography

[1] Fan, J. and I. Gijbels. "Local Polynomial Modeling and Its Applications." 1st Ed. CRC Press. Boca Raton: 1996.

[2] Ramsay, J.O. and B. W. Silverman. "Functional Data Analysis" 2nd Ed. Springer Science+Business Media. New York: 2006.

[3] Silverman, B.W. "Density Estimation for Statistics and Data Analysis." 1st Ed. CRC Press. Boca Raton: 1986.