

# ISYE 7406 - Homework 4

Matthew Bernstein

March 1, 2023

## Introduction

The purpose of this experiment is to compare different methods of the local smoothing over a particularly difficult function. Two sets of datapoints, one equidistant and one pre-set, were generated between  $-2\pi$  and  $2\pi$ . The idea was to smooth over the "Mexican Hat" function, which contains some properties that could be difficult for smoothing algorithms.

## Exploratory Data Analysis

The "Mexican Hat" function is shown in Figure 1

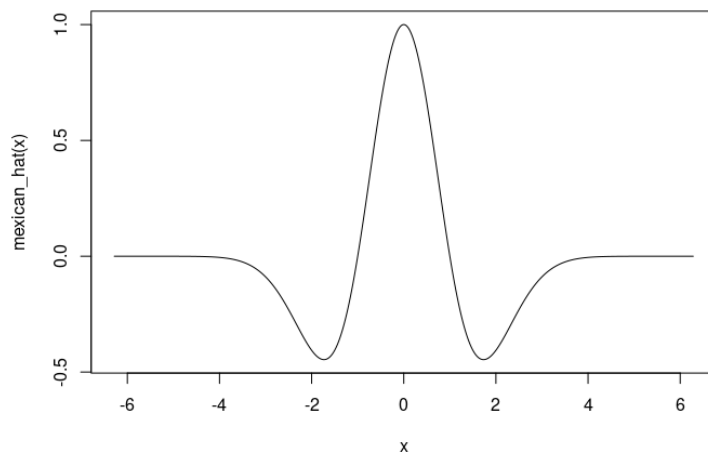


Figure 1: Mexican Hat function

While the center section appears to be a set of smooth curves that should be easily estimatable for a local smoothing algorithm, the tails are essentially flat. This could pose a problem

for smoothing algorithms that try to find local polynomials. It also affects the smoothing parameters. Smoothing closer to the tails is likely going to be less accurate as the surrounding points are going to be highly influenced by the other flat points.

## Methodology

In order to test the various local smoothing methods, two sets of points that fit the "mexican hat function":

$$f(x) = (1 - x^2)\exp(-0.5x^2) \quad -2\pi \leq x \leq 2\pi \quad (1)$$

One of the set of points is equidistant in the range  $[-2\pi, 2\pi]$ . The other set is deterministic. For each set of points, Gaussian jitter of form  $\mathcal{N}(0, 0.2^2)$  is added to each point and then 3 separate local smoothing methods are applied to the new data points:

1. LOESS
2. Nadaraya-Watson Kernel Smoothing with a Gaussian Kernel
3. Splines Smoothing

1000 trials for each sets points which the Gaussian jitter regenerated each time. The results at each  $x_i$  are aggregated and bias, variance, and MSR are calculated.

## Results

### *Equidistant Point Set*

Figure 2 shows the results from the equidistant point set. Kernel and Splines smoothing appear to fit the true data set very closely. Both have extremely low bias and MSE throughout the domain, although kernel smoothing has significantly more variance. LOESS was the most consistent smoothing methods but also the least accurate, this is could be due to an extremely high span parameter of 0.75 meaning of the 101 total points used, around 75 of them are used in the local smoothing and so points from far away affect the models smoothing. Interestingly, all smoothing algorithms have higher variance at the tails of the domain.

### *Deterministic Point Set*

Figure 3 shows the results of the deterministic point set. All three smoothing methods are fairly similar in this point set. Kernel smoothing is again the most accurate but also has the highest variance. Splines smoothing is the least accurate of the three in this point set, this is somewhat to be expected since spline smoothing relies on equidistant points. The lower  $\alpha$  value in the LOESS smoothing contributes to a more accurate model as well.

Figure 2: Results from Part 1, equidistant point set

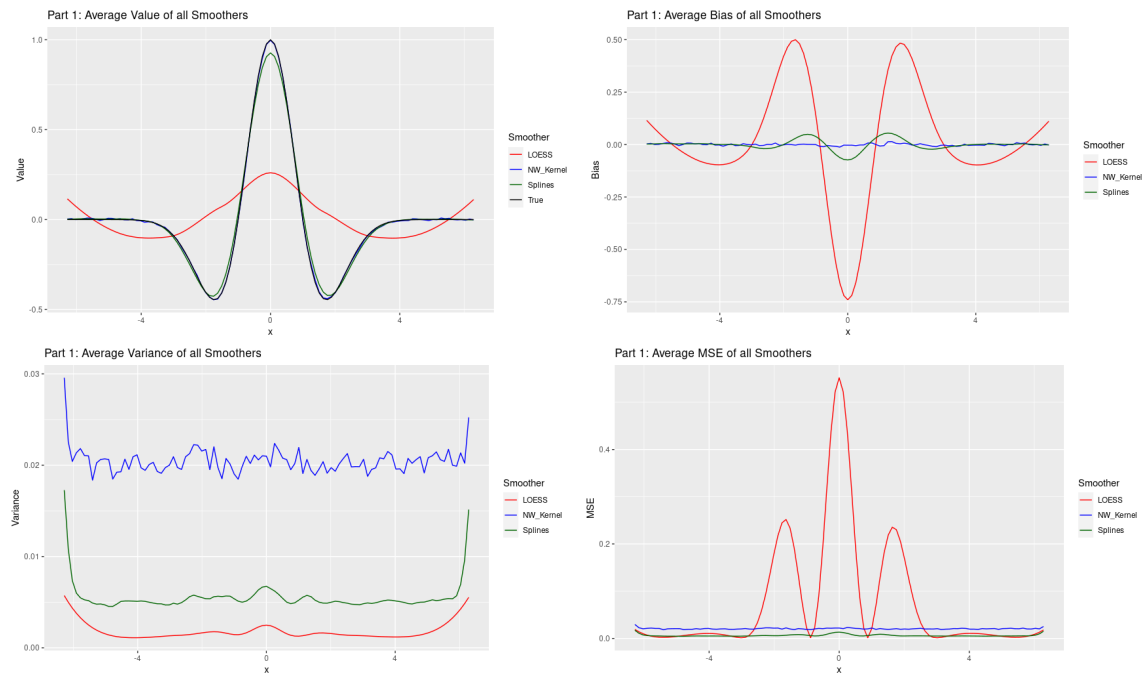
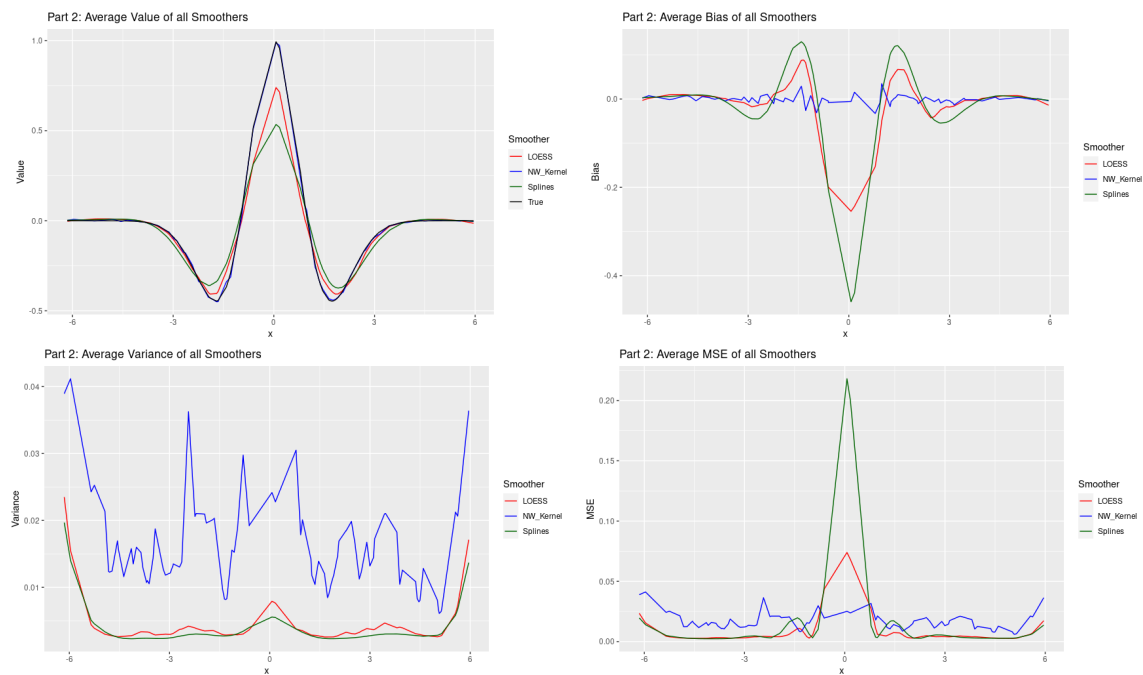


Figure 3: Results from Part 2, deterministic point set



## Conclusion

Each of the three smoothing algorithms can perform well in smoothing a difficult to estimate functions. Spline smoothing performs significantly better given an equidistant point set. LOESS smoothing can be particularly sensitive to the chosen smoothing parameter. Given the two data sets, it appeared that kernel smoothing was the most accurate, however with the highest variance so could be unreliable in lower iteration. In conclusion, each method has its strengths and weaknesses and are all valid methods for local smoothing.