Optimizing Knot Placement in Regression Splines

Joseph Holey   Michael O'Malley
STOR-i, Lancaster University

Introduction

- In this work we simulated a function $q(x)$ given by:

$$\frac{\sin(x)}{x} + 0.5\cos(2x + \frac{\pi}{4}). \tag{1}$$

- We then added some random normally distributed noise with a mean of zero and a standard deviation of 0.1 to our grid of values.
- Next, we used a variety of methods to obtain an estimate $\hat{f}(x)$ for the true function $q(x)$ based on simulated data. These methods required the use of the `mgcv` package and the `freeknotspline` package. We made use of bootstrapping in conjunction with both of these.

What is a Spline?

- Regressions splines are used to construct a model $\hat{f}(x)$ to fit a set of data. Many of them require knots to do this.
- Regression splines are a linear combination of basis function which depend on a set of knot points. The basis functions are constructed such that the resulting linear combination will be continuous and have a certain number of continuous derivatives. The basis functions highly depend on the placement of knot points.
- In this work we mainly used penalised splines (or P-splines) which punish models which vary too rapidly. To fit this model we sought to minimise the following equation:

$$||\boldsymbol{y} - \mathbf{X}\beta||^2 + \lambda \int_0^1 [f''(x)]^2 \, \mathrm{d}x \tag{2}$$

where $\mathbf{X}$ is our model matrix, β is a vector of unknown parameters that we are trying to find and λ is a tuning parameter [1].

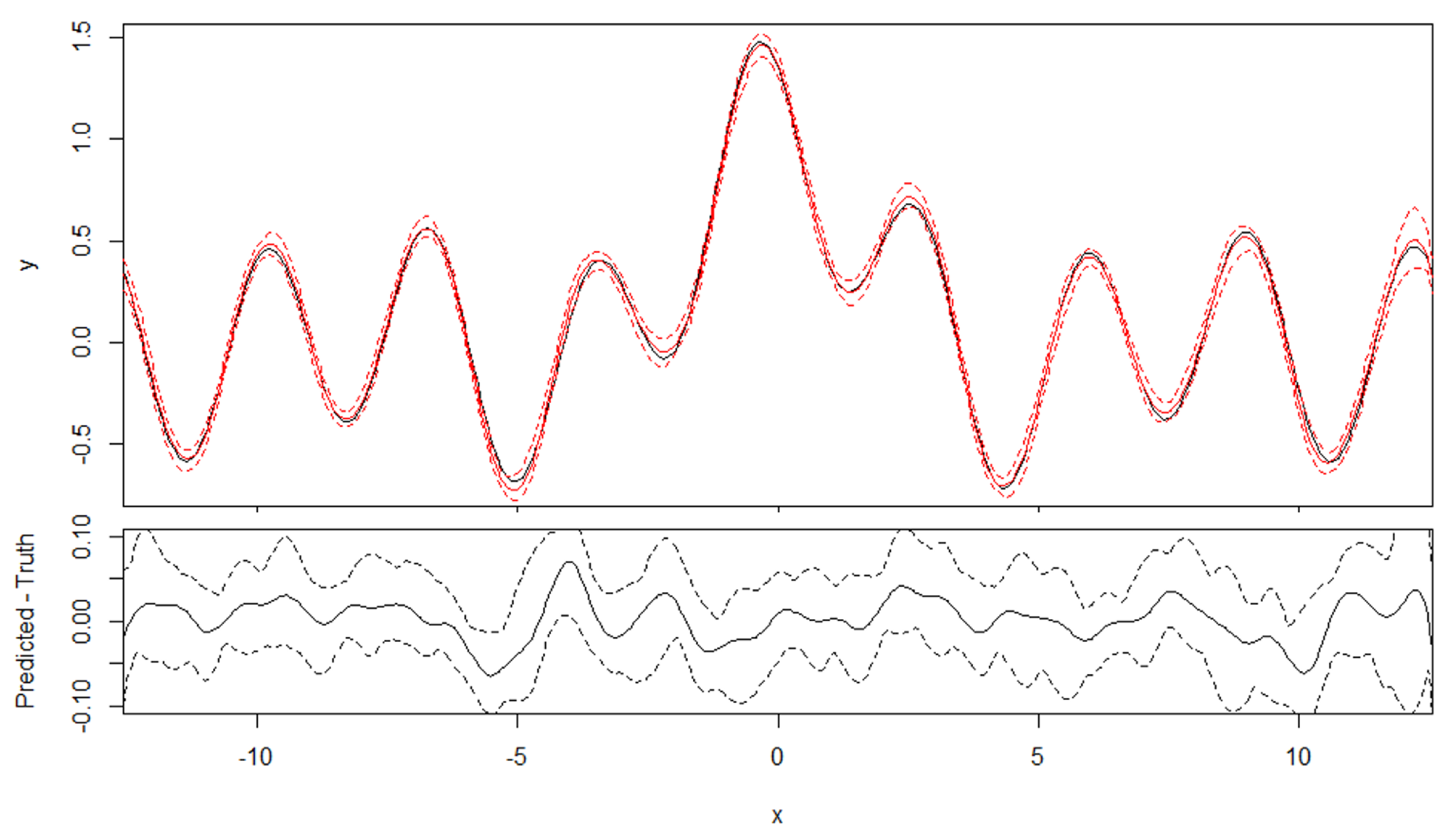

Figure 1: The upper panel shows the true function in black with the fit using the `mgcv` package and its 99% confidence intervals shown in red. The lower panel shows the fit and the confidence intervals with the true data values subtracted.

The Bootstrap

- The bootstrap is an example of a resampling method used for uncertainty estimation.
- If you had a set of n data points $(x_1, x_2, ..., x_n)$ then the bootstrap method would take a sample of these of size n with replacement so some of the data points may be repeated multiple times and others may not get selected at all.
- Measuring the statistic of interest (e.g. the mean) for each of these bootstrapped samples then allows you to determine the variance of that statistic [2].

Knot Positioning

- The positions of the knots for a spline can have a very large impact on how good a fit it provides to the data points.
- Conventionally the knots are placed at regular intervals over the range of the x values or at the quantiles of the data. These do not generally provide the best possible fit for a given data set.
- A poor selection of the location of knots can lead to splines that are not even competitive with a simple polynomial regression.
- We looked into some other algorithms for selecting the optimum knot locations.

The Genetic Algorithm

- The genetic algorithm begins by first randomly generating a large number of sets of k knots. There are then 3 main steps to the algorithm - selection, crossover and mutation [3].
- Selection consists of choosing the sets of knots which provide the smallest residual sum of squares.
- In the crossover step an integer, ℓ is randomly chosen between 1 and k and two parent sets are also chosen at random. A child set is then produced consisting of the first $\ell - 1$ knots of one of the parent sets and the last $k - \ell + 1$ knots of the other parent set.
- Mutation consists of randomly choosing one of the surviving knot sets and then selecting one of the knots ξ_ℓ. This knot is then replaced with one randomly selected in $(\xi_{\ell-1}, \xi_{\ell+1})$.
- The steps of selection, crossover and mutation are then iterated over multiple times to find the best set of knots.

Method	Mean Squared Error
Fixed Knots	5.896×10^{-4}
Genetic algorithm	5.762×10^{-4}

Table 1: This table shows the Mean Squared Error for the two methods between the truth and the fit.

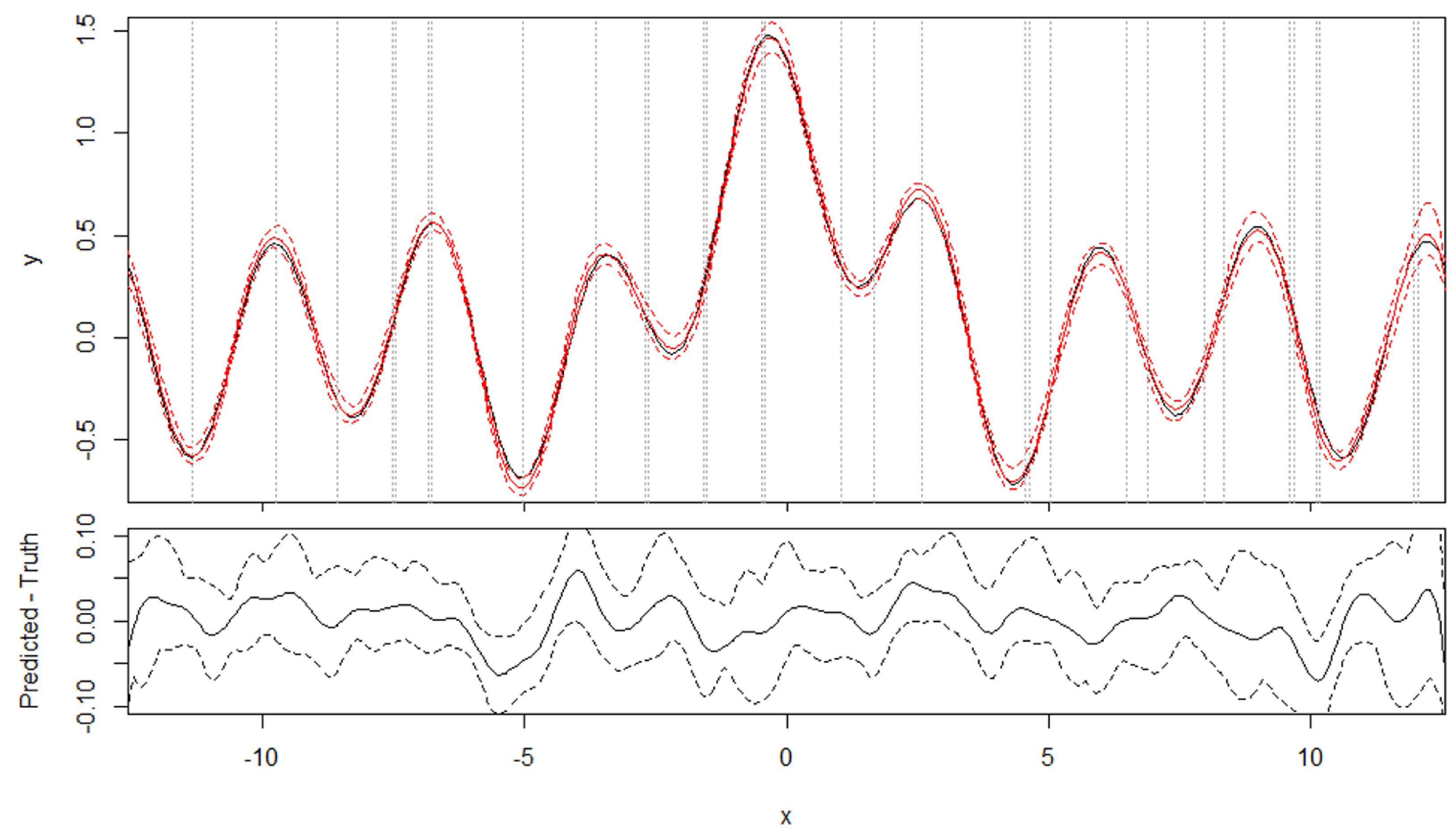

Figure 2: The upper panel shows the true function in black with the fit using the genetic algorithm and the `mgcv` package and its 99% confidence intervals shown in red. The vertical dotted lines denote the knot positions. The lower panel shows the fit and the confidence intervals with the true data values subtracted.

Analysis & Results

- We ran two different simulations in the project.
- Firstly we used bootstrapping with the `mgcv` package to find our fits and confidence intervals. The fit obtained is shown in Figure 1.
- In the second method we ran the genetic algorithm to find the best set of knots for our data set. We then reused these knots to fit to all the bootstrapped data sets. The fit obtained from this is shown in Figure 2.
- We assessed the quality of the fits by considering the mean squared error given by:

$$\frac{1}{n}\sum_{i=1}^{n}(\hat{f}(x_i) - q(x_i))^2, \tag{3}$$

where the x_i are our grid points. The results are shown in Table 1.
- In the future we would like to try running the genetic algorithm for each bootstrapped data set to find their individual set of optimum knots.

References

- S. N. Wood, *Generalized additive models: an introduction with R.* Chapman and Hall/CRC, 2017.
- G. James, D. Witten, T. Hastie, and R. Tibshirani, *An introduction to statistical learning*, vol. 112. Springer, 2013.
- S. Spiriti, R. Eubank, P. W. Smith, and D. Young, "Knot selection for least-squares and penalized splines," *Journal of Statistical Computation and Simulation*, vol. 83, no. 6, pp. 1020–1036, 2013.

j.holey@lancaster.ac.uk