EXTENDING THE LATTICE-BASED SMOOTHER USING A GENERALIZED ADDITIVE MODEL

By

Gulfaya Rakhmetova, M.S.

A Project Submitted in Partial Fulfillment of the Requirements

for the Degree of

Master of science
in

Statistics

University of Alaska Fairbanks

December 2017

APPROVED:

Julie McIntyre, Committee Chair

Margaret Short, Committee Member

Scott Goddard, Committee Member

Leah Berman, Chair

Department of Mathematics and Statistics

Abstract

The Lattice Based Smoother was introduced by McIntyre and Barry (2017) to estimate a surface defined over an irregularly-shaped region. In this paper we consider extending their method to allow for additional covariates and non-continuous responses. We describe our extension which utilizes the framework of generalized additive models. A simulation study shows that our method is comparable to the Soap film smoother of Wood et al. (2008), under a number of different conditions. Finally we illustrate the method's practical use by applying it to a real data set.

Table of Contents

1. Introduction

A main goal in environmental and geospatial studies is to model spatially distributed data as a function of the geographic location in order to construct a predicted surface. However, an issue occurs when a function is defined on a region with an irregular boundary. Conventional spatial smoothing methods, including kernel smoothers, kriging and spline models, lack the ability to appropriately account for complex boundaries. They ignore irregular boundaries such as peninsulas inside the region, and may smooth across the boundaries if points on either side of the boundary are close in terms of Euclidean distance. But since such points may be affected by different ecological processes, smoothing across boundaries might result in inappropriate estimates. This problem is known as 'leakage'.

Several statistical tools have been developed to address this problem. Wood et al. (2008) introduce a method called soap film smoothing. This is a spline estimator that uses two sets of basis functions, one for the interior region and one for the boundary. Scott-Hayward et al. (2014) illustrate the Complex Region Spatial Smoother (CReSS), which is based on the geodesic distances between data locations. This approach was an improvement to the Geodesic Low-Rank Thin Plate Splines (GLTPS) method of Wang and Ranalli (2007), which also employs geodesic distance, but allows some leakage depending on the chosen size of the local neighbourhoods around points. McIntyre and Barry (2017) introduced a kernel-based estimator of the spatial function, the Lattice-based smoother (LBS), that accounts for an irregular domain. The method is based on a diffusion process approximated by random walks along a lattice constructed over the region.

The inclusion of covariates, available at the same locations as the response, could lead to improved estimates and provide more information about the behaviour of the response variable. In this paper we investigate extensions to the LBS model by using the framework of generalized additive models (GAM; Hastie and Tibshirani, 1990). Our objectives are to incorporate information from additional covariates into the LBS model to improve spatial prediction, and to extend these models to non-normal responses.

This paper is organized as follows. In Section 2 we briefly review the LBS method of McIntyre and Barry (2017) and also review Hastie and Tibshirani's (1990) backfitting and local scoring algorithms for fitting GAMs. Section 3 is devoted to the simulation study to illustrate the performance of the method under different scenarios and compare it to the soap film smoother (SOAP; Wood et al., 2008). Section 4 presents the application of the method to a real data set on the sediment phosphorus concentration in Lake Michigan.

2. Background

2.1. **Lattice-based smoother**

Let Y be a response variable and $s = (z_1, z_2)$ be a location. Suppose that these are related by

$$Y_i = g(s_i) + \epsilon_i \qquad i = 1, \ldots, n$$

where the function $g(s)$ is a smooth function defined at locations $s \in \Omega$. The lattice-based smoother (McIntyre and Barry 2017) is a kernel regression estimator based on a two-dimensional diffusion process constrained to stay within Ω. The process is approximated by constructing a lattice comprised of N nodes within Ω. A kernel is defined from the distribution of random walks on the lattice, where walks originate at each observed data location (Barry and McIntyre 2012). This kernel is then used to estimate the regression surface.

We describe this process in more detail. Let $s_j = (x_j, y_j)$ denote the location of each node $j = 1, ..., N$ within the domain Ω. Denote nodes where responses Z_i are observed by $s_{j_i}, i = 1, ..., n$. If the location of the observed data doesn't fall on one of the nodes, it is moved to the nearest node with a small distortion. Alternatively the grid may be made as dense as needed to capture all the data positions. Let $Q_k(s_j; s_i)$ be the probability that a random walk of length-k, starting from the location s_i of the observed data, lands on node s_j. Then the probability mass function of the random walks' positions after k steps, originating at each node of observed data $s_{j_i}, i = 1, ..., n$ and ending at s_j is defined by

$$p_k(s_j) = \frac{1}{n} \sum_{i=1}^{n} Q_k(s_j; s_{j_i}), j = 1, ..., N.$$

Let A be the area of the whole domain. Then by scaling by area A, we get the kernel density estimator of the probability density function (Barry and McIntyre 2011)

$$\hat{f}(s_j) = \frac{1}{n} \sum_{i=1}^{n} \frac{N}{A} Q_k(s_j; s_{j_i}) = \frac{N}{A} p_k(s_j). \tag{1}$$

Then using this kernel function, the Nadaraya-Watson type regression estimator follows from equation 1,

$$\hat{g}(s_j) = \frac{\frac{1}{n}\frac{N}{A}\sum_{i=1}^{n} Z_i Q_k(s_j; s_{j_i})}{\frac{1}{n}\frac{N}{A}\sum_{i=1}^{n} Q_k(s_j; s_{j_i})} = \frac{\sum_{i=1}^{n} Z_i Q_k(s_j; s_{j_i})}{\sum_{i=1}^{n} Q_k(s_j; s_{j_i})}. \tag{2}$$

We note that the estimator depends on two important components: the set of transition probabilities defined for the random walks and the number of steps. The random walk transition is defined by a Markov chain. First we state the transition probabilities, $P(X_{k+1} = s_j | X_k = s_i)$, from one node to the neighbouring one. By neighbor nodes, we mean nodes located in eight directions from each other (N, S, E, W, NE, NW, SE, SW). We use the following transition probabilities defined by Barry and McIntyre (2011),

$$P(X_{k+1} = s_j | X_k = s_i) = \begin{cases} 1 - M\dfrac{q_j}{\max(q_j)}, & i = j \\ M\dfrac{1}{\max(q_j)}, & i \neq j, i \text{ and } j \text{ neighbours} \\ 0, & i \neq j, i \text{ and } j \text{ not neighbours} \end{cases}$$

where q_i is a number of neighbors at node $s_i, i = 1, \ldots, N$. M is the parameter between 0 and 1 controlling the probability of the random walk transition between nodes. Hence the probability of moving to a neighboring node is higher at the interior nodes than the probability on the edge. Therefore the random walks are more likely to remain at the same location on the boundary of the region at a given step. McIntyre and Barry (2017) show that the estimator in equation (2) can be computed efficiently by defining transition matrices and using matrix multiplication.

A second important component of the estimator is the number of steps k. This controls the smoothness of the estimator, similar to a bandwidth parameter. We select the number of steps k using cross-validation as described in McIntyre and Barry (2017).

2.2. **Additive and Generalized Additive Models**

Generalized additive models (GAM; Hastie and Tibshirani, 1990) provide a flexible framework for expanding nonparametric regression models to multiple dimensions and non-normal responses. First we describe the additive model, which is a generalization of the linear regression model. This is followed by a description of GAM, which relaxes assumption about the distribution of the response variable.

For models with multiple predictors, the linear regression model is simple and easy to interpret. However, the linear relationship between response and each predictor doesn't always hold. One way of generalizing this model is to assume

$$y = g(x_1, ..., x_p) + \epsilon$$

where $g()$ is any smooth function. However, there is a dimensionality problem in fitting this model (Hastie and Tibshirani, 1990). For high dimensional surfaces, i.e. number of predictors, it is hard to estimate and interpret the surface smoothers. The additive model makes it easier by making the model additive in predictors,

$$y_i = \beta_0 + g_1(x_{i1}) + g_2(x_{i2}) + ... + g_p(x_{ip}) + \varepsilon_i, \qquad i = 1, ..., n \tag{3}$$

where $g_1(x_{i1}), ..., g_p(x_{ip})$ can be any smooth or linear function. The role of each predictor in the model is examined separately. The error terms are assumed to be independently and identically distributed with zero mean and constant variance. Thereby the additive model makes a balance between fully nonparametric and parametric models by assuming the effect of each regressor is nonlinear but additive.

2.2.1. *Backfitting algorithm*

A method that enables estimation in the additive model is called the Backfitting Algorithm (Hastie and Tibshirani, 1990). It iteratively fits partial residuals on each predictor in turn by removing the effects of all other predictors. From equation (3) we can write

$$y_i - \beta_0 - \sum_{k \neq j} g_k(x_{ik}) = g_j(x_{ij}) + \varepsilon_i$$

The left hand side of the equation defines the new response, $y_i^{(j)} = g_j(x_{ij}) + \varepsilon_i$. Let $\boldsymbol{y^{(j)}} = (y_1^{(j)}, \ldots, y_n^{(j)})^T$, $\boldsymbol{x_{(j)}} = (x_{1j}, \ldots, x_{nj})^T$ and let $\boldsymbol{\hat{g}_j} = (\hat{g}_j(x_{1j}), \ldots, \hat{g}_j(x_{nj}))^T$ be a vector of the estimated values of the function g_j. We fit a nonparametric regression of $\boldsymbol{y^{(j)}}$ on $\boldsymbol{x_{(j)}}$ to estimate the function g_j and repeat this for all the predictors $j = 1, ..., p$ in the model. Let $S(\boldsymbol{x}, \boldsymbol{y})$ denote any smoother used to estimate the regression of $\boldsymbol{y}$ on $\boldsymbol{x}$ and let $\hat{S}(x_i; \boldsymbol{x}, \boldsymbol{y})$ denote its predicted value at $x = x_i$. The procedure of the backfitting algorithm entails

1. Initial estimate: $\hat{\beta_0} = \bar{y}$; $\hat{g}_j(x_{ij}) = 0$ for $i = 1, \ldots, n$ and $j = 1, \ldots, p$;

2. Compute the new response variable: for $j = 1, \ldots, p$

$$y_i^{(j)} = y_i - \hat{\beta}_0 - \sum_{k \neq j} \hat{g}_k(x_{ik})$$

3. Update $\boldsymbol{\hat{g}_j}$ by regressing the new outcome $\boldsymbol{y^{(j)}}$ on j^{th} predictor and center:

$$\hat{g}_j(x_{ij}) = \hat{S}(x_{ij}; \boldsymbol{x_{(j)}}, \boldsymbol{y^{(j)}})$$

$$\hat{g}_j(x_{ij}) = \hat{g}_j(x_{ij}) - \frac{1}{n}\sum_{i=1}^{n} \hat{g}_j(x_{ij}) = \hat{g}_j - \bar{g}_j$$

4. Repeat steps 2 and 3 until the change in $\boldsymbol{\hat{g}_j}$ doesn't exceed some pre-specified threshold for each $j = 1, \ldots, p$.

The final fitted regression function is given by

$$\hat{y}_i = \hat{\beta}_0 + \sum_{j=1}^{p} \hat{g}_j(x_{ij})$$

Also we note that the smoothers used can be any combination of regression estimators including linear regression, kernel regression, splines or LBS.

2.2.2. *Local Scoring algorithm*

The generalized additive model can allow a non-normal distribution of the response variable and a complex variance structure. The backfitting algorithm defined previously can be adapted to fit these models as well. We illustrate the algorithm for the logistic regression model.

Logistic regression is a type of generalized linear model with binary response 0 and 1. Let $\boldsymbol{x} = (x_1, x_2, \ldots, x_p)^T$ be a vector of predictors. Hence given the values of the covariates $\boldsymbol{x}$, $Y \sim$ Bern(p), i.e. the response has a Bernoulli distribution with probability p. Then the conditional mean is defined by

$$\mu = E[Y|\mathbf{x}] = p(Y = 1|\boldsymbol{x}) = p(\boldsymbol{x}).$$

The logistic model equates the conditional mean to a function of predictor variables via the logit link η,

$$\eta = g(\mu) = \text{logit}(p(\boldsymbol{x})) = \log \frac{p(\boldsymbol{x})}{1 - p(\boldsymbol{x})} = \beta_0 + \sum_{j}^{p} g_j(x_j)$$

Then

$$p(\boldsymbol{x}) = \frac{\exp(\beta_0 + \sum_j^p g_j(x_j))}{1 + \exp(\beta_0 + \sum_j^p g_j(x_j))}.$$

Finally the conditional variance is defined by

$$\text{Var}(Y|\boldsymbol{x}) = p(\boldsymbol{x})(1 - p(\boldsymbol{x})) = \mu(1 - \mu).$$

The variance of the response is a function of x, that is, it changes as values of predictors change. Therefore there is a heteroskedasticity issue, which could be resolved by applying iteratively weighted least squares. Hence in GAM, the model is estimated simply using the weighted backfitting algorithm called local scoring (Hastie and Tibshirani, 1990). The intuition is the same as in the backfitting method, but local scoring involves repeatedly fitting weighted regression to estimate functions $g_1, g_2, ..., g_p$.

The procedure includes two iterative processes, one nested in another. After initializing starting values we define the link function η and compute the probability of an event as given in step 2 below. Then we construct a new response z_i and weights w_i which will serve as starting values in the next weighted backfitting iteration. New estimates of functions $\tilde{\boldsymbol{g}}_{\boldsymbol{j}} = (\tilde{g}_j(x_{1j}), \dots, \tilde{g}_j(x_{nj}))^T$ are obtained by regressing weighted partial residuals $w_i z_i^{(j)}$ and weights w_i on x_{ij}. S can be any smoother or simple linear regression. This process is continued until $\tilde{\boldsymbol{g}}_{\boldsymbol{j}}$ converges. Then we update the outer cycle iteration estimates and repeat the whole procedure until it meets the convergence criterion. We note that these steps are described for estimation with kernel smoothers. Local scoring algorithm steps entails

1. Set the initial values: $\hat{\beta}_0 = \log(\frac{\bar{y}}{1-\bar{y}})$ and $\hat{g}_j(x_{ij}) = 0$ for $i = 1, \dots, n$ and $j = 1, \dots, p$.

2. Compute the following: for $i = 1, \dots, n$

   $\hat{\eta}_i = \hat{\beta}_0 + \sum_j \hat{g}_j(x_{ij})$ and $\hat{p}_i = \dfrac{1}{1 + \exp(-\hat{\eta}_i)}$

3. Construct new dependent variable z_i and weights w_i:

$$z_i = \hat{\eta}_i + \frac{y_i - \hat{p}_i}{\hat{p}_i(1 - \hat{p}_i)}, \qquad w_i = \hat{p}_i(1 - \hat{p}_i), \qquad i = 1, \dots, n$$

4. Estimate new β_0 and $g_j(x_{ij})$ for $i = 1, \dots, n$ and $j = 1, \dots, p$, iteratively using a weighted backfitting algorithm with dependent variable and weights estimated in step 3. The steps for this algorithm are

- Set starting values: $\tilde{\beta}_0 = \bar{z}$ and $\tilde{g}_j(x_{ij}) = 0$ for $i = 1, \ldots, n$ and $j = 1, \ldots, p$
- Define partial residuals

$$z_i^{(j)} = z_i - \tilde{\beta}_0 - \sum_{k \neq j} \tilde{g}_k(x_{ik}) \qquad i = 1, \ldots, n; \qquad j = 1, \ldots, p$$

  and weighted partial residuals $w_i z_i^{(j)}$ with weights defined in step 3.

- Fit a weighted model to estimate the numerator and denominator of kernel estimator defined in equation 2. Let $\boldsymbol{z^{(j)}} = (z_1^{(j)}, \ldots, z_n^{(j)})^T$ and $\boldsymbol{w} = (w_1, \ldots, w_n)^T$. Then

$$\tilde{g}_j(x_{ij}) = \frac{S(x_{ij}; \boldsymbol{x_j}, \boldsymbol{w z^{(j)}})}{S(x_{ij}; \boldsymbol{x_j}, \boldsymbol{w})}$$

- Update the function $\boldsymbol{\tilde{g}_j}$ and repeat step 4 until a convergence criterion is reached for each $j = 1, \ldots, p$.

5. Update the functions $\hat{g}_j(x_{ij})$ in step 1 with new obtained functions $\boldsymbol{\tilde{g}_j}$. Repeat the entire process until convergence.

Using this algorithm, we incorporate the LBS estimator with additional covariates and nonnormal responses.

3. Simulation studies

We conducted a simulation study to investigate the proposed extensions to models fit with LBS. Our objective is to illustrate how the method works and compare it with soap film smoothing proposed by Wood et al. (2008). We focus on introducing additional covariates to the LBS model. We give an example of modelling with a binary response but do not do a full simulation study for that case. Implementation of the method is still ongoing.

3.1. **Ramsay Horseshoe simulation using backfitting algorithm**

Ramsay (2002) introduced a function over a domain with irregular boundaries in a horseshoe-shaped region (Figure 2(a)). The horseshoe-shaped domain is a good example with a complex domain to evaluate the smoothing abilities of methods across the boundary. We used this test function to generate a surface function over the U-shaped domain. Specifically, we added dependence with a covariate X according to

$$Y_{true} = 2 + \frac{1}{4}X + g(Z), \tag{4}$$

where $g(Z)$ is the Ramsay function at location $Z = (z_1, z_2)$. Then for our simulation study we performed Monte Carlo replicates of the method. First we generated response variables at randomly sampled locations using the equation (4) above and adding an error term generated from $N(0, \sigma_\varepsilon^2)$. Then we applied the backfitting algorithm as described in Section 2.2.1, where for the smooth function S we used a lattice-based smoother and incorporated it with the linear term X.

For the lattice-based smoother (LBS) the distance between grid locations in the constructed lattice was chosen to be 0.05 units vertically and horizontally. The R package latticeDensity (Barry, 2012) was employed to fit this model. The SOAP was implemented by the R package mgcv. We fit models having 32 interior knots and 40-knot cyclic cubic regression spline as the boundary curve in correspondence with the papers Wood et al. (2008) and McIntyre and Barry (2017). The generalized cross validation was used to select the smoothing parameters.

We compared performances of the two estimators under different scenarios. First, we considered correlation between X and $Z = (z_1, z_2)$ according to $X = \rho\sqrt{z_1^2 + z_2^2} + (1 - \rho)U$, where $U \sim$ Uniform(0, 2). Thus, when $\rho = 0$, the covariate X is independent of the location and simply generated from the uniform distribution, whereas $\rho = 0.35$ gives a correlation coefficient r^2 of about 0.60 between X and longitude. Second, we considered two different sample sizes, n= 600 and n=100. And third, we considered two levels of noise, $\sigma_\varepsilon = 1.36$ and $\sigma_\varepsilon = 2.36$, which correspond to the signal to noise ratio $\text{var}(y_{true})/(\text{var}(y_{true}) + \sigma_\varepsilon^2) = 0.75$ and 0.5 respectively. To evaluate the performance of the methods and compare them we made predictions at 987 points inside the region. We computed mean squared error (MSE) for each simulated dataset and pointwise bias by

$$MSE = \frac{1}{N}\sum_{j=1}^{N}(y_j - \hat{y}_j)^2$$

$$\text{Bias}(y_j) = y_j - \hat{y}_j \qquad j = 1, \ldots, N$$

The Wilcoxon signed rank test was employed for pairwise comparison of the medians of MSE scores at 0.05 level of significance. The null hypothesis assumes that there is no significant difference between medians of two methods. The results are based on the 50 simulated data sets.

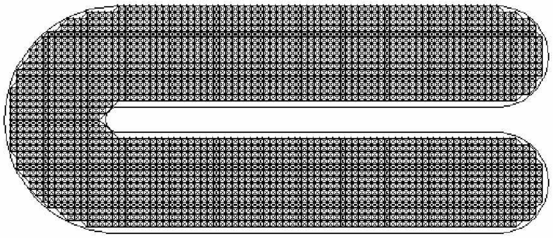


FIGURE 1. Lattice over the horseshoe domain

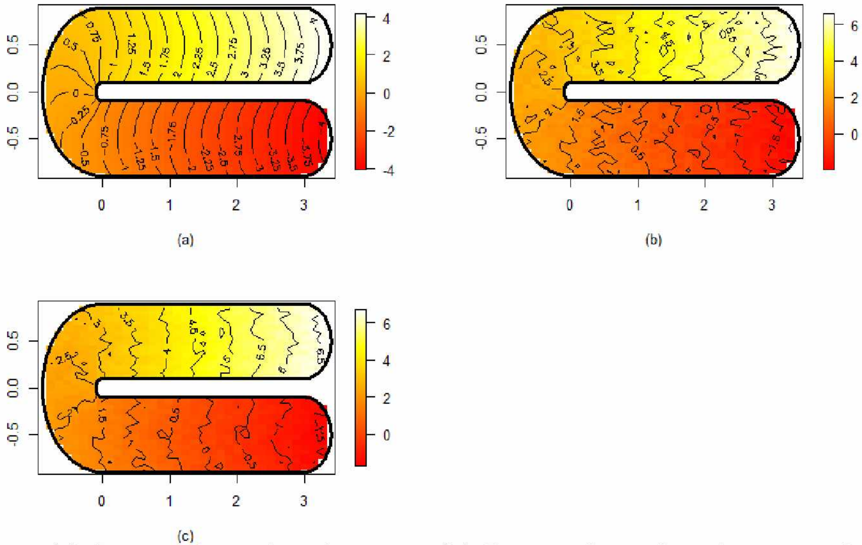


FIGURE 2. (a) Ramsay horseshoe function; (b) Ramsay horseshoe domain with true response variable ($\rho = 0$) ; (c) Ramsay horseshoe domain with true response variable ($\rho = 0.35$)

The contour map of Ramsay Horseshoe is given in Figure 2 (a) with the function ranging from -4 to 4 and the resulting response variable generated by equation 4 with uncorrelated and correlated covariates are shown in Figure 2 (b) and (c) respectively. The average pointwise bias for LBS and SOAP with sample size of 600 and dependent and independent covariates showed quite similar results based on the 50 simulated datasets (Figure 3). However, in both cases of the covariates LBS has less bias than that for SOAP, especially in the elbow region and on the edge. Figure 4 considers the same model, but with sample size of 100. LBS tends to outperform SOAP in the tips of upper and lower arms for uncorrelated covariates, but has greater bias in the outer curve. When the covariates are related with correlation coefficient of 0.6, LBS seems to perform better on the entire region.

The natural log of MSE for each sample size, correlation and noise levels is plotted in Figure 5 in terms of the boxplots. SOAP has lower median of MSE score, but larger spread. In general, MSE scores tend to be lower with larger sample size and smaller noise level. In all cases a Wilcoxon signed rank test showed that the difference in medians of MSE between LBS and SOAP is significantly greater than zero.

The pointwise bias plots for the noise level 2.36 are given in Appendix. In spite of the higher median of MSE score, LBS appears to perform better. SOAP shows the greatest error in the tips of both arms as well as in the elbow region and near to the outer regions. When the covariates are correlated LBS seems to outperform SOAP on the entire domain.

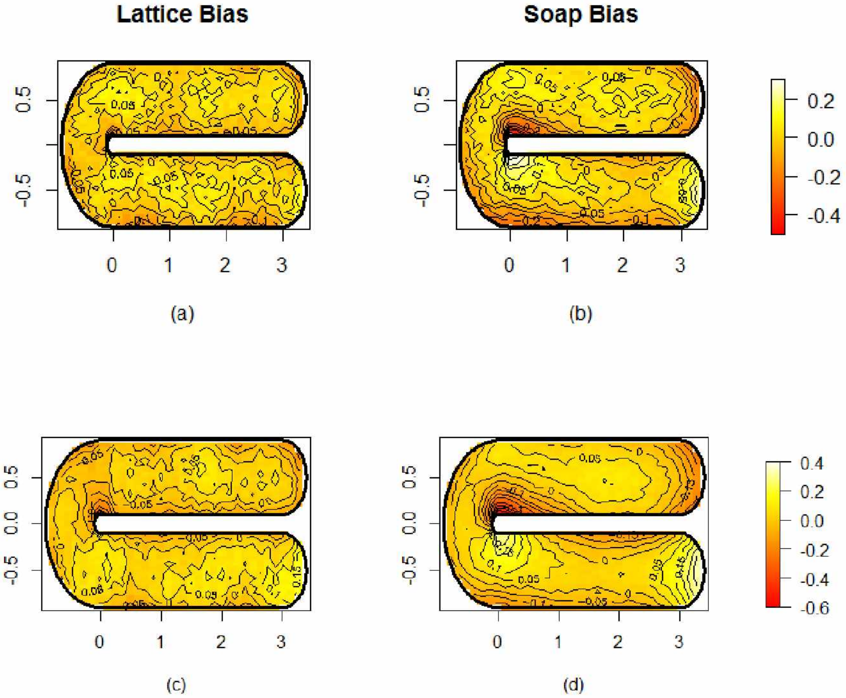


FIGURE 3. Average pointwise bias with n=600 and $\sigma = 1.36$: (a) LBS with $r^2 = 0$; (b) SOAP with $r^2 = 0$; (c) LBS with $r^2 = 0.60$; (d) SOAP with $r^2 = 0.60$

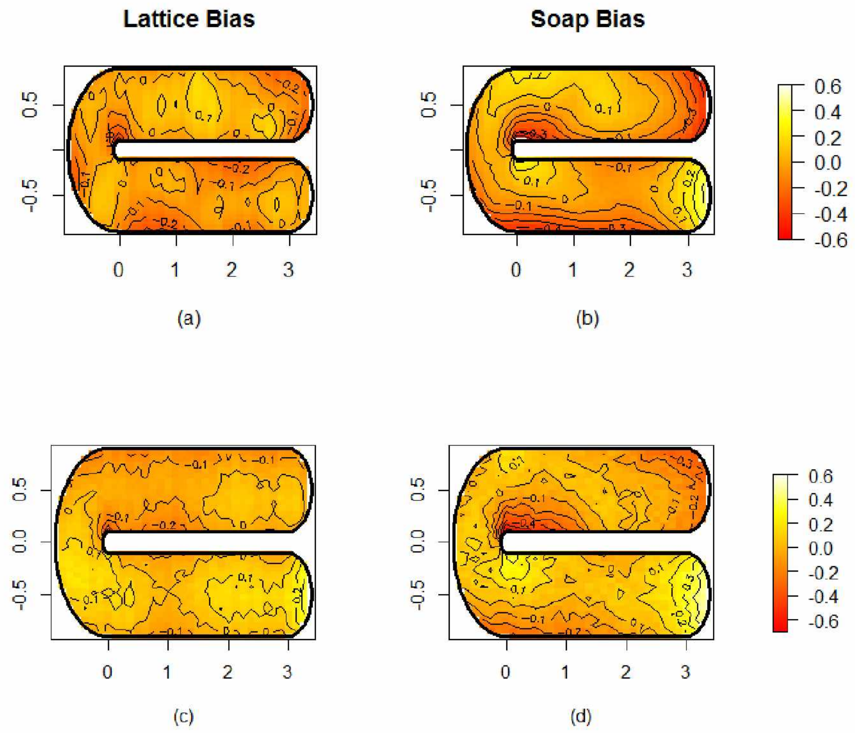


FIGURE 4. Average pointwise bias with n=100 and $\sigma = 1.36$: (a) LBS with $r^2 = 0$; (b) SOAP with $r^2 = 0$; (c) LBS with $r^2 = 0.60$; (d) SOAP with $r^2 = 0.60$

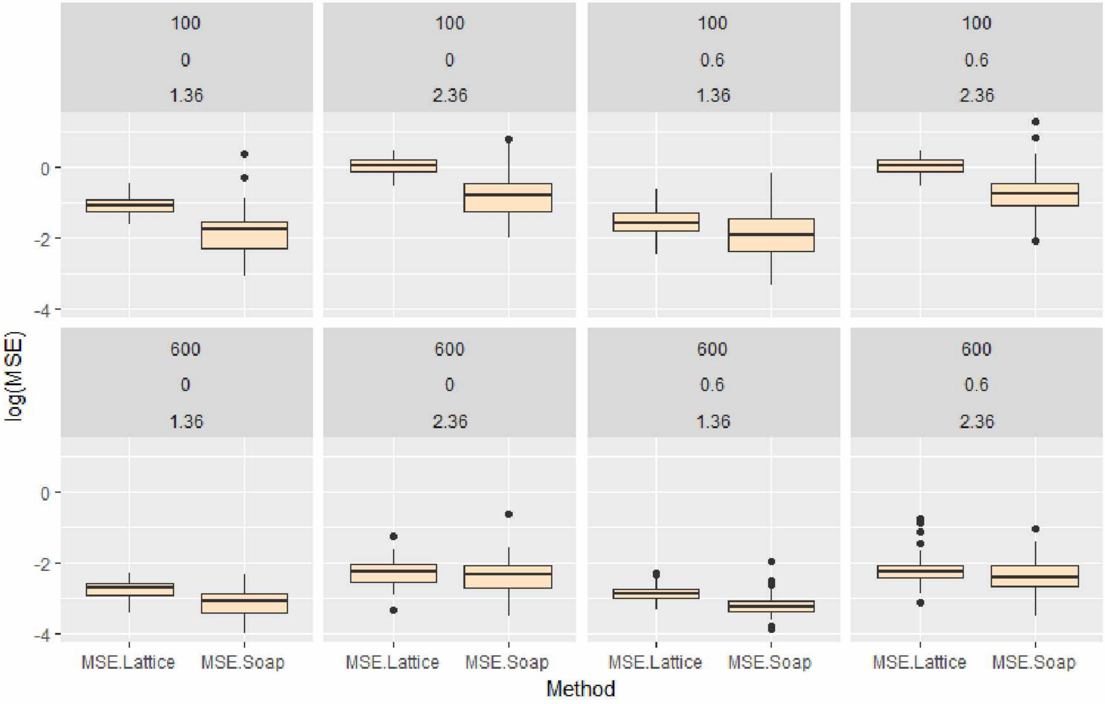


FIGURE 5. Boxplots of log(MSE) for LBS and SOAP. The first row of the figure illustrates boxplots for sample size of n=100 with $r^2 = 0$ and 0.6; two noise levels, $\sigma = 1.36$ and 2.36. The bottom row shows the same, but with n=600

.

3.2. Example using local scoring algorithm

We created a polygon with irregular boundaries and made up a link function of the logistic regression as defined in section 2.2.2 to relate the response to a non-linear additive component for location. In addition we included a linear component of an additional covariate from the uniform distribution with the values between zero and three. Then, the true link function, true probability and binary response variable were defined over the domain by the following function and visually represented in Figure 6 (a),

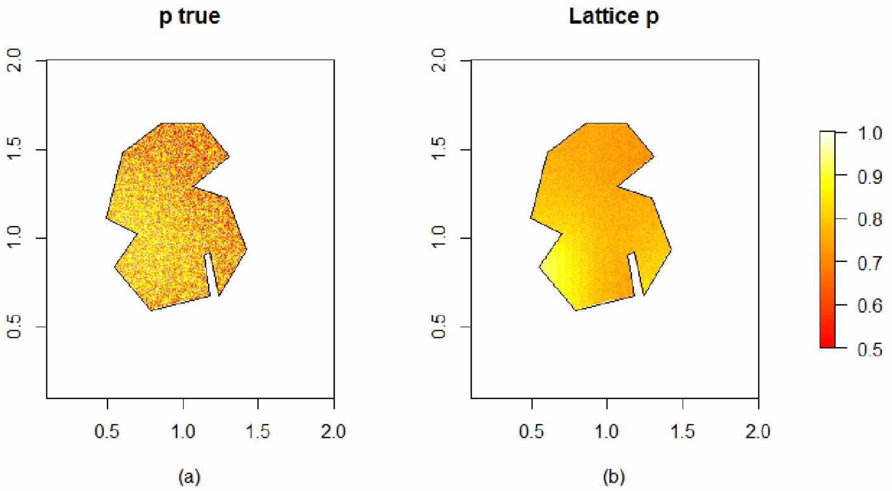
$$x \sim \text{Uniform } (0,3)$$

$$\eta = 0.7x - \sin(2z_1)\cos(3z_2)\cos(z_1 z_2)$$

$$p = \frac{1}{1+\exp^{-\eta}}$$

$$y \sim \text{Bern}(p)$$

The simulation was conducted based on $n = 400$ randomly selected points from the interior polygon, and responses in terms of η were generated according to the above equation. Estimation followed the local scoring algorithm described in Section 2.2.2. The prediction was made at 16,338 points on the surface and predicted probabilities are plotted in Figure 6 (b). It captured pretty well the higher values of p on the bottom right side of the region.



(a)

(b)

FIGURE 6. (a) True values of probability p; (b) Predicted probability

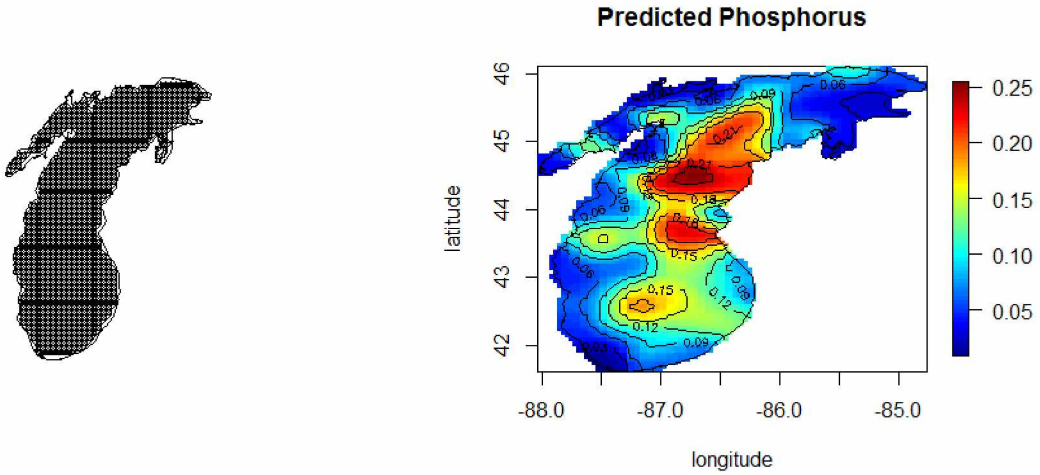


Figure 7. The figure on the left illustrates the lattice used to fit the lattice-based smoother. Right figure is the map of the Lake Michigan with estimated phosphorus concentration fitted without depth variable

4. Application to Lake Michigan data

We apply the proposed method to estimate the concentration of a certain nutrient, namely sodium hydroxide extractable phosphorus, in Lake Michigan sediments. The sampling was conducted as part of Lake Michigan Mass Balance Study carried out by U.S. Environmental Protection Agency (EPA CDX, 2002) through cruising multiple times starting from July 1994 and ending in May 1996 (Miller et al., 2016). The data were obtained from EPA's database and include sediment phosphorus measurements at 117 stations in Lake Michigan along with the water depth where the samples were collected. There were multiple measurements at some locations, therefore we found the mean at those stations. The phosphorus concentration ranges from 0.006 to 0.28 mg/g after taking the mean.

First we employed the lattice-based smoother of McIntyre and Barry (2017) to model the phosphorus concentration as a function of its location. The lattice with node space of 0.8 units was constructed (Figure 7) to estimate the function and prediction was made on a grid of 39,910 points on the surface. It can be observed, for example, that the estimates are not smoothed out across boundaries on the west and north-east sides of the polygon, where there is a bay creating a complex boundary (Figure 7).

The phosphorus concentration appears to increase with depth, suggesting that there is a potential for depth to improve spatial estimates of phosphorus (Figure 8 (a)). We modelled it as a linear function of depth and smooth function of location within a generalized additive model context as shown in previous sections. For further prediction of the measurement at other locations, we obtained the data on the bathymetry of lake Michigan from National Oceanic and Atmospheric Administration database (NOAA) and randomly sampled 25,000 observations, of which 24,521 observations were defined inside of the region. Visual representation can be found in Figure 9 (a) and the resulting estimated values of sediment phosphorus concentrations are mapped in Figure 9 (b) with the higher values as the lake gets deeper towards the inner side.

In order to estimate the performance of the LBS method, we divided the data into training set and test set with the fractions 0.75 and 0.25 respectively. Then we fitted the LBS and SOAP models using the training set and predicted the values of phosphorus at locations and depths contained in the test set. The difference between the estimated and actual values of phosphorus concentrations from the test set were computed for both models as well as the MSEs were calculated. Both models present quite good fit giving fairly small and identical MSE: 0.00159 for LBS and 0.00161 for SOAP. However, the boxplot for the difference shows greater spread for LBS, but possible outliers with SOAP (Figure 10).

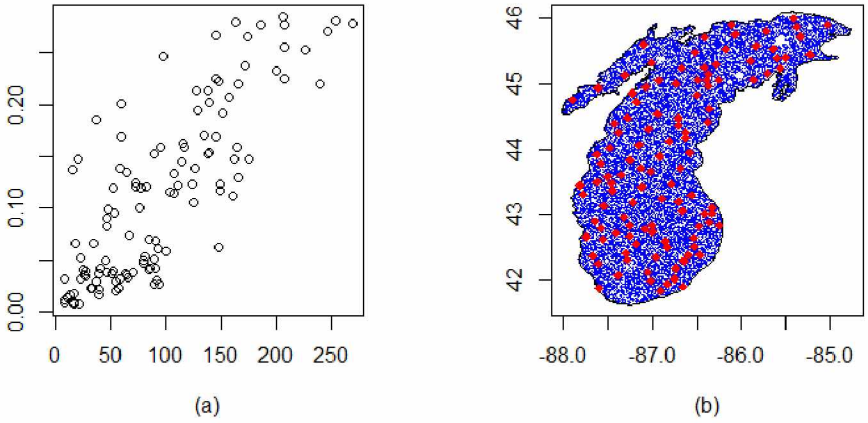


Figure 8. (a) Phosphorus concentration vs. Depth; (b) Polygon of Lake Michigan. The locations of measured phosphorus concentration are given in red dots. Blue dots are the locations where the predictions are made after fitting the model.

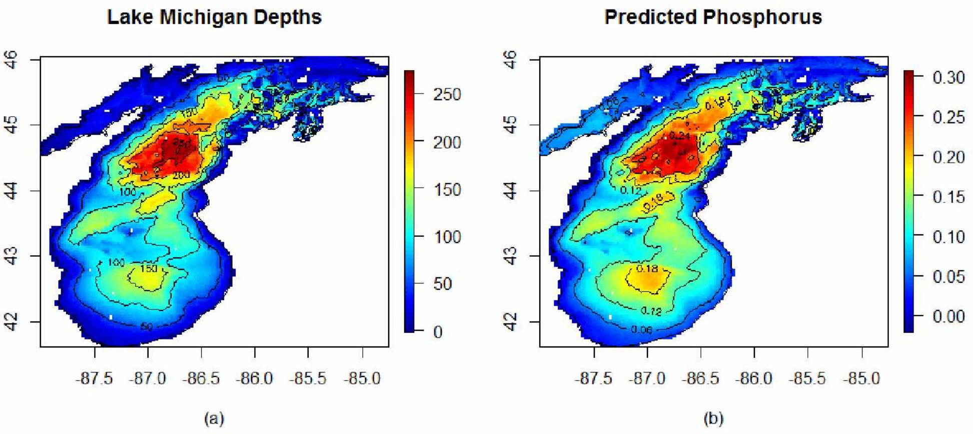


FIGURE 9. Map of the Lake Michigan with predicted phosphorus concentration

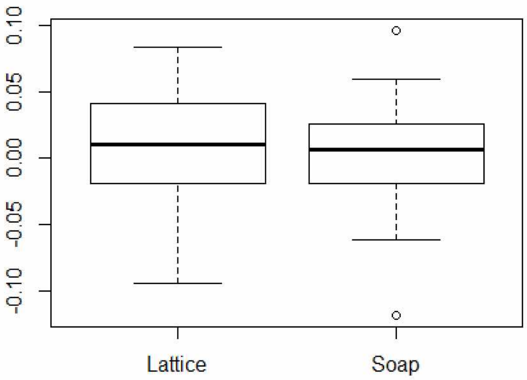


FIGURE 10. Boxplots of the differences between the estimated and actual values of phosphorus concentrations using LBS and SOAP

5. Conclusion

In this paper we extended the lattice-based smoother in a way to include additional covariates. We illustrated backfitting and local scoring algorithms, which allow estimation of generalized additive models.

Simulation studies revealed that this method is comparable with the well-known soap film smoother, which has already been tested in multiple papers to perform well with complex boundaries. Even though the medians of MSE scores were a little bit higher for LBS, the average pointwise bias seemed to perform better. Especially this was distinct near the edges and inner curves of the horseshoe domain, where SOAP showed higher bias. The cases with greater noise level and correlated covariates didn't reveal any issue. The implementation of the latter required more time to run. The average pointwise bias showed that the LBS outperformed SOAP on the entire domain in the cases with the correlated covariates.

Although we illustrated the application of the local scoring algorithm to binary response based on a simulated data set, the method's implementation is still ongoing. Furthermore, one of the future studies could be making the inference from the model such as defining the statistical significance of predictors, obtaining standard errors, etc. Moreover fitting the model for the data with different distributions might also be a topic of interest.

Acknowledgements

I would like to thank my advisor Dr Julie McIntyre for her great help and support throughout the project. I'm also grateful to the committee members, Margaret Short and Scott Goddard for their feedback on the draft.

References

ArcGIS (2015). *Lake Michigan Shoreline.* Retrieved from `https://www.arcgis.com`, on September 20, 2017.

Barry, R.P. and McIntyre, J. (2017). A Lattice-Based Smoother for Regions with Irregular Boundaries and Holes. In press. *Journal of Computational and Graphical Statistics.*

Barry, R. P. (2012). *latticeDensity: Density estimation and nonparametric regression on irregular regions.* R package version 1.0.7. Available at: `http://CRAN.R-project.org/package=latticeDensity`

Barry, R. P. and McIntyre, J. (2011). Estimating animal densities and home range in regions with irregular boundaries and holes: A lattice-based alternative to the kernel density estimator. *Ecological Modelling* **222,** 1666–1672.

Environmental Protection Agency (EPA) Central Data Exchange (CDX) (2002). *Lake Michigan Mass Balance Results.* Retrieved from `https://cdx.epa.gov/CDX/MyCDX/`, on September 20, 2017.

Hastie, T.J. and Tibshirani, R. J. (1999). *Generalized Additive Models.* Chapman and Hall.

Miller, D. H., Xia, X. S, W.-C. and Rossmann, R (2016). Distribution of Sediment Measurements in Lake Michigan as a Case Study: Implications for Estimating Sediment and Water Interactions in Eutrophication and Bioaccumulation Models. *Applied Mathematics* **7,** 1846–1867.

National Oceanic and Atmospheric Administration (NOAA) (n.d.). *Bathymetry of Lake Michigan.* Retrieved from `https://www.ngdc.noaa.gov/mgg/greatlakes/michigan.html`, on September 20, 2017.

Ramsay, T. (2002). Spline smoothing over difficult regions. *Journal of the Royal Statistical Society, Series B* **64,** 307–319.

Scott-Hayward, L. A. S., Mackenzie, M. L., Donovan, C. R., Walker, C. G., and Ashe, E. (2014). Complex region spatial smoother (CReSS). *Journal of Computational and Graphical Statistics* **23,** 340–360.

Wang, H. and Ranalli, M. G. (2007). Low-rank smoothing splines on complicated domains. *Biometrics* **63,** 209–217.

Wood, S. N., Bravington, M. V., and Hedley, S. L. (2008). Soap film smoothing. *Journal of the Royal Statistical Society, Series B* **70,** 931–955.

Appendix

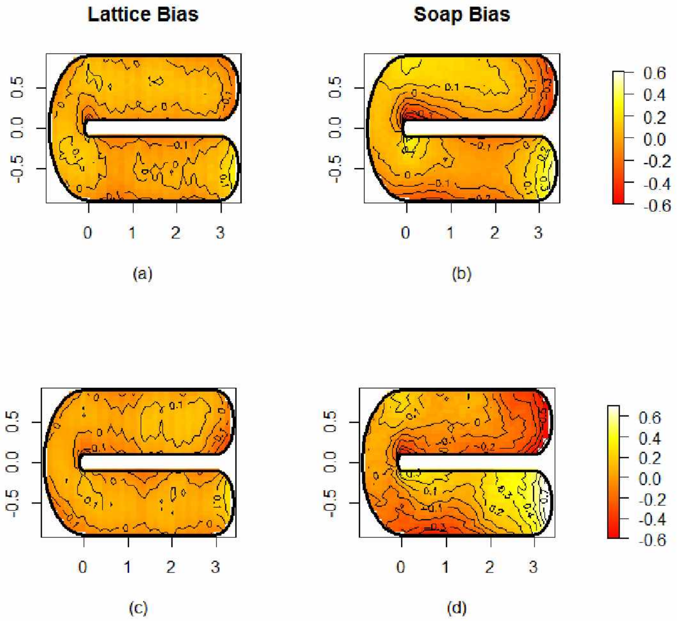


Figure 11. Average pointwise bias with n=600 and $\sigma = 2.36$: (a) LBS with $r^2 = 0$; (b): Soap with $r^2 = 0$; (c) LBS with $r^2 = 0.60$; (d) Soap with $r^2 = 0.60$

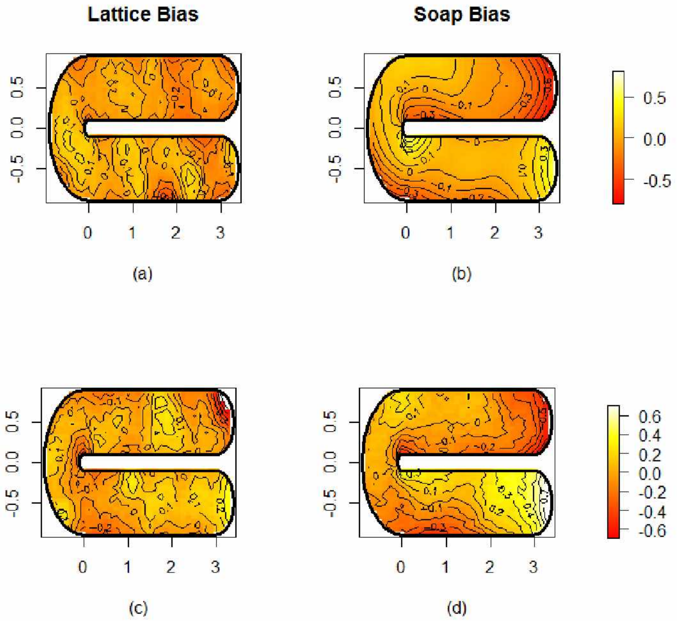


Figure 12. Average pointwise bias with n=100 and $\sigma = 2.36$: (a) LBS with $r^2 = 0$; (b): Soap with $r^2 = 0$; (c) LBS with $r^2 = 0.60$; (d) Soap with $r^2 = 0.60$