

The **bbefkr** Package

Han Lin Shang
University of Southampton

Abstract

Recent advances in computer recording and storing technology have tremendously increased the presence of functional data, whose graphical representation can be infinite-dimensional curve, image or shape. When a set of function-valued predictor and scalar-valued response variables is observed, the functional kernel regression provides a flexible way to estimate its possible non-linear relationship. However, as with any type of kernel regression, it requires an optimal selection of a smoothing parameter, called bandwidth. In the literature of nonparametric functional regression, bandwidth parameter is often selected by a functional cross validation. In this article, we introduce another bandwidth estimation method that uses the information about error density to help with the optimal selection of bandwidth in the regression function. We first describe the proposed Bayesian method and its implementations for estimating bandwidth parameters in a nonparametric functional regression and a semi-functional partial linear regression, using a readily-available \mathbb{R} add-on package. The proposed method is then demonstrated using a series of simulation studies, as well as a spectroscopy data set.

Keywords: bandwidth selection, Bayesian model selection, functional Nadaraya-Watson estimator, kernel error density, marginal likelihood, random-walk Metropolis, simulation inefficiency factor.

Introduction

The aim of this article is to describe the \mathbb{R} functions that are readily-available in the **bbefkr** package, for estimating bandwidth parameters in a nonparametric functional regression and a semi-functional partial linear regression. This article was motivated by the recent developments in nonparametric functional regression estimators for analysing the relationship between function-valued predictors and real-valued responses. Some nonparametric functional estimators include: functional Nadaraya-Watson (NW) estimator ([Ferraty and Vieu 2006](#)), functional local linear estimator ([Barrientos-Marin, Ferraty, and Vieu 2010](#)), functional k -nearest neighbour estimator ([Burba, Ferraty, and Vieu 2009](#)), and distance-based local linear estimator ([Boj, Delicado, and Fortiana 2010](#)).

In all aforementioned functional estimators, the estimation accuracy of conditional mean, conditional median or conditional mode depends crucially on the optimal selection of the bandwidth parameter. Commonly, it is selected by the functional cross validation, which is designed to assess the predictive performance of a model by an average of certain measures for the ability of predicting a subset of functions by a model fit, after deleting just these functions from the data set. Functional cross validation (CV) has the appealing feature that no estimation of the error variance is required. However, since residuals affect the estimation accuracy of regression function, functional CV may select sub-optimal bandwidths. This in turn leads to inferior estimation accuracy of regression

functions (see for example, [Shang 2013a,b](#)). As an alternative, we present a Bayesian bandwidth estimation method that simultaneously estimates the bandwidths in the regression function and kernel-form error density (see [Shang 2013a](#), for the importance of error density estimation).

Before introducing the bandwidth estimation method, we first define the problem more precisely. Let $\mathbf{y} = (y_1, y_2, \dots, y_n)^\top$ be a vector of scalar responses, $T = (T_1, T_2, \dots, T_n)^\top$ be a set of functional predictors and $\mathbf{X} = (X_{i1}, \dots, X_{ip})_{i=1, \dots, n}$ be p -dimensional real-valued predictors. We consider two regression models, namely a nonparametric functional regression and a semi-functional partial linear regression, where the latter one is a generalisation of the former one. The semi-functional partial linear regression can be defined as

$$y_i = \mathbf{X}_i \boldsymbol{\beta} + m(T_i) + \varepsilon_i, \quad i = 1, 2, \dots, n, \quad (1)$$

where \mathbf{X}_i represents the i th observation of the real-valued predictors, $\boldsymbol{\beta} = (\beta_1, \dots, \beta_p)^\top$ is a vector of unknown regression coefficients in the parametric counterpart, m is an unknown smooth function that performs the mapping from function space to real space, and $(\varepsilon_1, \dots, \varepsilon_n)$ are identically distributed random error satisfying

$$E(\varepsilon_i | \mathbf{X}_i, T_i) = 0.$$

The nonparametric functional regression model has the only nonparametric part of the equation (1). In both regression models, we are interested in accurately estimating the regression function m , which is mainly determined by the optimal selection of bandwidth parameters.

This article proceeds as follows. Bayesian bandwidth estimation method is first described and its estimation accuracy is then compared based on marginal likelihood or Bayes factor. Through a series of simulation studies, the sampling algorithm is implemented using the **bbefkr** package. Conclusions are given in the end.

Bayesian bandwidth estimation

Estimation of error density

The unknown error density $f(\varepsilon)$ can be approximated by a location-mixture of Gaussian densities (see also [Roeder and Wasserman 1997](#)), given by

$$f(\varepsilon; b) = \frac{1}{n} \sum_{j=1}^n \frac{1}{b} \phi\left(\frac{\varepsilon - \varepsilon_j}{b}\right),$$

where $\phi(\cdot)$ is the probability density function of the standard Gaussian distribution, and the component Gaussian densities have means at ε_j , for $j = 1, 2, \dots, n$, and a common standard deviation b . Although error ε_j is unknown, it can be estimated by the functional NW estimator. Thus, the density of y_i is approximated by the estimated error density $\hat{f}(\varepsilon; b)$, expressed as

$$\hat{f}(\varepsilon; b) = \frac{1}{n} \sum_{j=1}^n \frac{1}{b} \phi\left(\frac{\varepsilon - \hat{\varepsilon}_j}{b}\right),$$

where b represents the estimate of residual bandwidth. In order to avoid the possible selection of $b = 0$, a leave-one-out version is used given by

$$\hat{f}(\hat{\varepsilon}_i; b) = \frac{1}{n} \sum_{\substack{j=1 \\ j \neq i}}^n \frac{1}{b} \phi\left(\frac{\hat{\varepsilon}_i - \hat{\varepsilon}_j}{b}\right),$$

where $\hat{\varepsilon}_i = y_i - \hat{m}(T_i; h)$ is the i th residual for $i = 1, 2, \dots, n$, in the nonparametric regression. In the semi-functional partial linear regression, $\hat{\varepsilon}_i = y_i - \mathbf{X}_i \hat{\boldsymbol{\beta}} - \hat{m}(T_i; h)$. Given (h, b) and independent and identically distributed (iid) assumption of the errors, the kernel likelihood of $\mathbf{y} = (y_1, y_2, \dots, y_n)^\top$ can be approximated by

$$\hat{L}(\mathbf{y}|h, b) = \prod_{i=1}^n \left[\frac{1}{n-1} \sum_{\substack{j=1 \\ j \neq i}}^n \frac{1}{b} \phi\left(\frac{\hat{\varepsilon}_i - \hat{\varepsilon}_j}{b}\right) \right].$$

Prior density

We now discuss the issue of prior density for the bandwidths. Let $\pi(h^2)$ and $\pi(b^2)$ be the prior of squared bandwidths h and b . Since h^2 and b^2 play the role of variance parameters in the Gaussian densities, we assume that the priors of h^2 and b^2 are inverse Gamma density, denoted by $\text{IG}(\alpha_h, \beta_h)$ and $\text{IG}(\alpha_b, \beta_b)$, respectively. Thus, the prior densities of h^2 and b^2 are given by

$$\begin{aligned} \pi(h^2) &= \frac{(\beta_h)^{\alpha_h}}{\Gamma(\alpha_h)} \left(\frac{1}{h^2}\right)^{\alpha_h+1} \exp\left(-\frac{\beta_h}{h^2}\right), \\ \pi(b^2) &= \frac{(\beta_b)^{\alpha_b}}{\Gamma(\alpha_b)} \left(\frac{1}{b^2}\right)^{\alpha_b+1} \exp\left(-\frac{\beta_b}{b^2}\right), \end{aligned}$$

where $\alpha_h = \alpha_b = 1.0$ and $\beta_h = \beta_b = 0.05$ as hyper-parameters.

Posterior density

According to the Bayes theorem, the posterior of h^2 and b^2 is approximated by (up to a normalising constant):

$$\pi(h^2, b^2 | \mathbf{y}) \propto \hat{L}(\mathbf{y}|h^2, b^2) \pi(h^2) \pi(b^2),$$

where $\hat{L}(\mathbf{y}|h^2, b^2)$ is the approximated likelihood function with squared bandwidths.

We use the adaptive random-walk Metropolis algorithm to sample (h^2, b^2) (see [Shang 2013a](#), for details). In order to assess the convergence of the Markov chain Monte Carlo (MCMC) algorithm, we use the notion of simulation inefficiency factor ([Meyer and Yu 2000](#)). It is a measure of autocorrelation among iterations, and provides an indication of how many iterations are required to have the iid draws from the posterior distributions.

Bayesian model selection

In Bayesian inference, model selection or averaging is often conducted through the Bayes factor of the model of interest against a competing model. The Bayes factor reflects a summary of evidence

provided by the data supporting the model as opposed to its competing model. The Bayes factor can be defined as the ratio of the marginal likelihoods under different models.

The marginal likelihood is defined as the expectation of likelihood with respect to the prior of parameters. It is seldom computed as the integral of the product of the likelihood and prior of parameters, but instead, is often computed numerically (Gelfand and Dey 1994; Newton and Raftery 1994; Chib 1995; Kass and Raftery 1995; Geweke 1999, among others). We utilised the method proposed by Chib (1995) to compute the marginal likelihood.

Let $\boldsymbol{\theta} = (h, b)$ be the parameter vector and \mathbf{y} be the data. Chib (1995) showed that the marginal likelihood under model A is expressed as

$$P_A(\mathbf{y}) = \frac{l_A(\mathbf{y}|\boldsymbol{\theta})\pi_A(\boldsymbol{\theta})}{\pi_A(\boldsymbol{\theta}|\mathbf{y})},$$

where $l_A(\mathbf{y}|\boldsymbol{\theta})$, $\pi_A(\boldsymbol{\theta})$ and $\pi_A(\boldsymbol{\theta}|\mathbf{y})$ denote the likelihood, prior and posterior under model A, respectively.

The Bayes factor of model A against model B is defined as

$$\text{BF} = \frac{P_A(\mathbf{y})}{P_B(\mathbf{y})}$$

Based on the Bayes factor, we can determine which model is more superiorer than a competing model with different levels of evidences (see Kass and Raftery 1995, for more details).

Simulation studies

We describe the construction of the simulated data. First of all, one builds simulated discretised data points of functional curves:

$$T_i(t_j) = a_i \cos(2t_j) + b_i \sin(4t_j) + c_i(t_j^2 - \pi t_j + \frac{2}{9}\pi^2), \quad i = 1, 2, \dots, n, \quad (2)$$

where $0 \leq t_1 \leq t_2 \leq \dots \leq t_{100} \leq \pi$ are equispaced points, a_i, b_i and c_i are independently drawn from a uniform distribution on $[0, 1]$, and n represents the sample size. The functional form in (2) is taken from Ferraty, Van Keilegom, and Vieu (2010), and has also been studied by Shang (2013a,b).

Once the curves are defined, one simulates a functional regression model to compute the responses, given below:

- 1) construct a nonparametric component $m(T_i) = 10 \times (a_i^2 - b_i^2)$, which performs the mapping from function-valued space to real-valued space.
- 1b) construct a parametric component, where X_{i1} and X_{i2} are independently drawn from a uniform distribution $U(0, 1)$ for $i = 1, 2, \dots, n$. The true values of regression coefficients are set to be $\boldsymbol{\beta} = (-1, 2)$. The regression function is constructed as

$$\tau_i = m(T_i) + \mathbf{X}_i\boldsymbol{\beta}, \quad \text{fore } i = 1, 2, \dots, n,$$

- 2) generate $\varepsilon_1, \varepsilon_2, \dots, \varepsilon_n$ from a standard normal distribution.

3) compute the corresponding response:

$$y_i = \tau_i + \varepsilon_i, \quad \text{for } i = 1, 2, \dots, n.$$

Smooth curves

First, we consider a set of smoothed curves as predictors, and choose a semi-metric based on 2nd derivative to measure the distances among curves. The relationship between predictors and responses can be modelled by a nonparametric functional regression or a semi-functional partial linear regression. In the case of a nonparametric functional regression, the estimated bandwidths can be obtained as follows:

```
# error density is approximated by a kernel density of residuals with a global bandwidth
dum_np_global = bayMCMC_np_global(data_x=simcurve_smooth_normerr, data_y=simresp_np_normerr,
data_xnew=simcurve_smooth_normerr, warm=1000, M=1000, range.grid=c(0,pi), q=2, nknot=20)

# estimated bandwidth parameters (h,b)
dum_np_global$xpfinalres

# estimated values for the regression function
dum_np_global$mhat

# simulation inefficiency factor to evaluate the convergence of MCMC
dum_np_global$sif_value

# log marginal likelihood computed by Chib (1995)'s method
dum_np_global$mlikeres

# acceptance rate of the bandwidths in the regression function and kernel-form error density
c(dum_np_global$acceptnwMCMC, dum_np_global$accepterroMCMC)

# estimated error density (probability density function)
dum_np_global$fore.den.mkr

# estimated error density (cumulative probability density function)
dum_np_global$fore.cdf.mkr
```

By using localised bandwidths, the estimation of error density can be more accurate than a global bandwidth because large absolute residuals are assigned relatively large bandwidths, while small absolute residuals are assigned relatively small bandwidths. An example is provided below

```
# error density is approximated by a kernel density of residuals with localised bandwidths
dum_np_local = bayMCMC_np_local(data_x=simcurve_smooth_normerr, data_y=simresp_np_normerr,
data_xnew=simcurve_smooth_normerr, warm=1000, M=1000, range.grid=c(0,pi), q=2, nknot=20)

# estimated bandwidth parameters (h, b, badj)
```

```

dum_np_local$xpfinalres

# estimated values for the regression function
dum_np_local$mhat

# simulation inefficiency factor to evaluate the convergence of MCMC
dum_np_local$sif_value

# log marginal likelihood computed by Chib (1995)'s method
dum_np_local$mlikeres

# acceptance rate of the bandwidths in the regression function and kernel-form error density
c(dum_np_local$acceptnwMCMC, dum_np_local$accepterroMCMC, dum_np_local$acceptepsilonMCMC)

# estimated error density (probability density function)
dum_np_local$fore.den.mkr

# estimated error density (cumulative probability density function)
dum_np_local$fore.cdf.mkr

```

Based on the log marginal likelihood (LML), the kernel-form error density with localised bandwidths is preferred to a global bandwidth. The Bayes factor can be derived from the difference in LML: $\exp(\text{LML}_A - \text{LML}_B)$, where A and B represent two different estimators.

In a semi-functional partial linear regression, the algorithm is similar to a nonparametric functional regression. Thus, we only highlight the additional output produced, such as the estimation of regression coefficients in the parametric part of the regression function. This is described as follows:

```

# error density is approximated by a kernel density of residuals with a global bandwidth
dum_semi_global = bayMCMC_semi_global(data_x=simcurve_smooth_normerr,
data_y=simresp_semi_normerr, data_xnew=simcurve_smooth_normerr, Xvar=Xvar, warm=1000, M=1000,
range.grid=c(0,pi), q=2, nknot=20)

# estimated regression coefficients
# other outputs have the same interpretation as the nonparametric functional regression
dum_semi_global$betahat

# log marginal likelihood computed by Chib (1995)'s method
dum_semi_global$mlikeres

# error density is approximated by a kernel density of residuals with localised bandwidths
dum_semi_local = bayMCMC_semi_local(data_x=simcurve_smooth_normerr,
data_y=simresp_semi_normerr, data_xnew=simcurve_smooth_normerr, Xvar=Xvar, warm=1000, M=1000,
range.grid=c(0,pi), q=2, nknot=20)

# estimated regression coefficients
# other outputs have the same interpretation as the nonparametric functional regression
dum_semi_local$betahat

# log marginal likelihood computed by Chib (1995)'s method
dum_semi_local$mlikeres

```

Rough curves

We also consider a set of rough curves as predictors, and use a semi-metric based on the functional principal components. To highlight the difference between a set of smooth curves and rough curves, Figure 1 presents the simulated curves as an example.

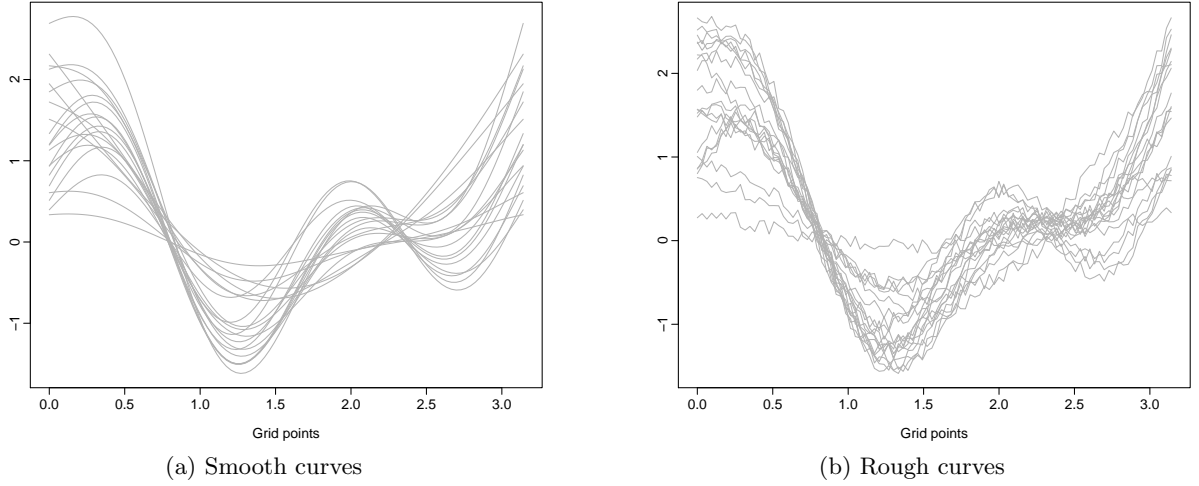


Figure 1: Simulated data

We compare the log marginal likelihoods between the semi-metric based on the 2nd derivative and the semi-metric based on three functional principal components.

```
# error density is approximated by a kernel density of residuals with a global bandwidth
# using the semi-metric based on 2nd derivative
rough_np_global_deriv = bayMCMC_np_global(data_x=simcurve_rough_normerr,
data_y=simresp_np_normerr, data_xnew=simcurve_rough_normerr, warm=1000, M=1000,
range.grid=c(0,pi), q=2, nknot=20)

# a global bandwidth using the semi-metric based on three functional principal components
rough_np_global_pca = bayMCMC_np_global(data_x=simcurve_rough_normerr,
data_y=simresp_np_normerr, data_xnew=simcurve_rough_normerr, warm=1000, M=1000,
semimetric="pca", q=3)

# comparing two semi-metrics based on their log marginal likelihoods
c(rough_np_global_deriv$mlikeres, rough_np_global_pca$mlikeres)
```

Similar findings also held for the nonparametric functional regression model with localised bandwidths. These can be implemented as follows:

```
# error density is approximated by a kernel density of residuals with localised bandwidths
# using the semi-metric based on 2nd derivative
rough_np_local_deriv = bayMCMC_np_local(data_x=simcurve_rough_normerr,
data_y=simresp_np_normerr, data_xnew=simcurve_rough_normerr, warm=1000, M=1000,
range.grid=c(0,pi), q=2, nknot=20)
```

```
# localised bandwidths using the semi-metric based on three functional principal components
rough_np_local_pca = bayMCMC_np_local(data_x=simcurve_rough_normerr,
data_y=simresp_np_normerr, data_xnew=simcurve_rough_normerr, warm=1000, M=1000,
semimetric="pca", q=3)
```

```
# comparing two semi-metrics based on their log marginal likelihoods
c(rough_np_local_deriv$mlikeres, rough_np_local_pca$mlikeres)
```

Thus, for a set of rough curves, the semi-metric based on functional principal components is more adequate than the semi-metric based on derivative. However, for a set of smooth curves, it is advantageous to use the semi-metric based on derivatives, as described below:

```
# error density is approximated by a kernel density of residuals with a global bandwidth
# using the semi-metric based on derivative
smooth_np_global_deriv = bayMCMC_np_global(data_x=simcurve_smooth_normerr,
data_y=simresp_np_normerr, data_xnew=simcurve_smooth_normerr, warm=1000, M=1000,
range.grid=c(0,pi), q=2, nknot=20)
```

```
# a global bandwidth using the semi-metric based on functional principal components
smooth_np_global_pca = bayMCMC_np_global(data_x=simcurve_smooth_normerr,
data_y=simresp_np_normerr, data_xnew=simcurve_smooth_normerr, warm=1000, M=1000,
semimetric="pca", q=3)
```

```
# comparing two semi-metrics based on their log marginal likelihoods
c(smooth_np_global_deriv$mlikeres, smooth_np_global_pca$mlikeres)
```

```
# localised bandwidths using the semi-metric based on derivative
smooth_np_local_deriv = bayMCMC_np_local(data_x=simcurve_smooth_normerr,
data_y=simresp_np_normerr, data_xnew=simcurve_smooth_normerr, warm=1000, M=1000,
range.grid=c(0,pi), q=2, nknot=20)
```

```
# localised bandwidths using the semi-metric based on functional principal components
smooth_np_local_pca = bayMCMC_np_local(data_x=simcurve_smooth_normerr,
data_y=simresp_np_normerr, data_xnew=simcurve_smooth_normerr, warm=1000, M=1000,
semimetric="pca", q=3)
```

```
# comparing two semi-metrics based on their log marginal likelihoods
c(smooth_np_local_deriv$mlikeres, smooth_np_local_pca$mlikeres)
```

Spectroscopy application

Let us consider a food quality control application, previously studied by [Ferraty and Vieu \(2006\)](#); [Aneiros-Pérez and Vieu \(2006\)](#); [Shang \(2013a\)](#) and [Shang \(2013b\)](#). The data set was obtained from <http://lib.stat.cmu.edu/datasets/tecator>. Each food sample contains finely chopped pure meat with different percentages of the fat, protein and moisture contents. For each unit i (among 215 pieces of finely chopped meat), we observe one spectrometric curve, denoted by T_i , which corresponds to the absorbance measured at a grid of 100 wavelengths. Given a new spectrometric

curve T_{new} , we aim to predict the corresponding fat, protein or moisture content. A graphical display of spectrometric curves is shown in Figure 2.

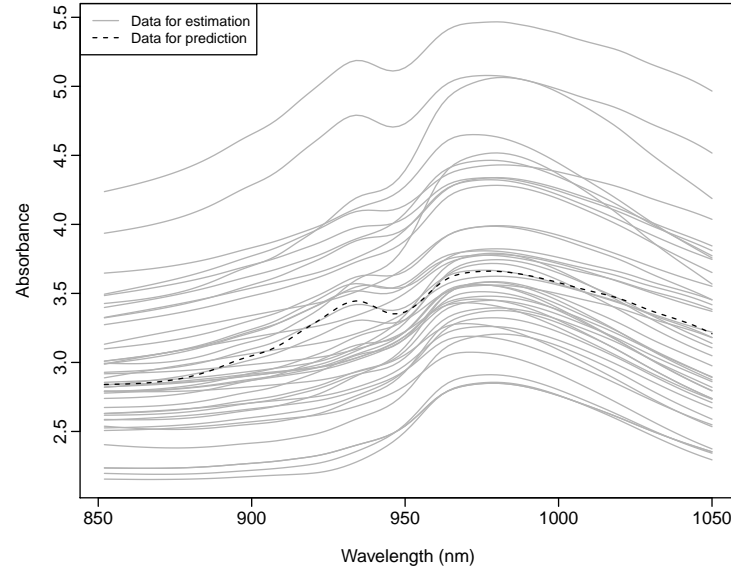


Figure 2: The first 45 spectrometric curves. Among these curves, the first 44 curves are used for the estimation of bandwidth parameters, while the 45th curve is used for the prediction of fat content

To predict the fat content, we consider a nonparametric functional regression and a semi-functional partial linear regression. As shown in Shang (2013b), the semi-functional partial linear regression gives more accurate forecasts than the nonparametric functional regression, since it uses additional information about protein and moisture contents. The forecasts can be obtained as follows:

```
# We use the first 44 pairs of data to estimate the relationship. Based on a new curve,
# we can then predict its corresponding response using a nonparametric functional regression
fat_np_global = bayMCMC_np_global(specurves[1:44,], fat[1:44], specurves[45,],
range.grid=c(0,pi), q=2, nknot=20)

# Point forecast
fat_np_global$pointforecast

# 95% prediction interval
fat_np_global$PI

# Implement localised bandwidths
fat_np_local = bayMCMC_np_local(specurves[1:44,], fat[1:44], specurves[45,],
range.grid=c(0,pi), q=2, nknot=20)

# Point forecast
fat_np_local$pointforecast

# 95% prediction interval
```

```
fat_np_local$PI

# Using a semi-functional partial linear regression with a global bandwidth
fat_semi_global = bayMCMC_semi_global(data_x = specurves[1:44,], data_y = fat[1:44],
data_xnew = specurves[45,], Xvar = cbind(protein[1:44], moisture[1:44]),
Xvarpred = matrix(c(protein[45], moisture[45]), nrow=1), range.grid=c(0,pi), q=2, nknot=20)

# Point forecast
fat_semi_global$pointforecast

# 95% prediction interval
fat_semi_global$PI

# Implement localised bandwidths
fat_semi_local = bayMCMC_semi_local(data_x = specurves[1:44,], data_y = fat[1:44],
data_xnew = specurves[45,], Xvar = cbind(protein[1:44], moisture[1:44]),
Xvarpred = matrix(c(protein[45], moisture[45]), nrow=1), range.grid=c(0,pi), q=2, nknot=20)

# Point forecast
fat_semi_local$pointforecast

# 95% prediction interval
fat_semi_local$PI
```

Conclusion

This article describes a Bayesian bandwidth estimation method and its implementations in a nonparametric functional regression and a semi-functional partial linear regression, using the **bbefkr** package. The method can simultaneously estimate bandwidths in the regression function estimated by the functional NW estimator and kernel-form error density. As illustrated by a series of simulation studies, we found that the use of localised bandwidths is preferred to a global bandwidth based on marginal likelihood or Bayes factor. For modelling a set of smooth curves, it is advantageous to use a semi-metric based on derivative, whereas a semi-metric based on functional principal components should be used for a set of rough curves. From the spectroscopy data set, we demonstrate the advantage of the proposed Bayesian method, that is to simultaneously produce point and interval forecasts.

In the future research, the method reviewed in this article can be extended to other nonparametric functional estimators and other nonparametric functional regression models.

References

- Aneiros-Pérez G, Vieu P (2006). “Semi-functional partial linear regression.” *Statistics & Probability Letters*, **76**(11), 1102–1110.
- Barrientos-Marin J, Ferraty F, Vieu P (2010). “Locally modelled regression and functional data.” *Journal of Nonparametric Statistics*, **22**(5), 617–632.

- Boj E, Delicado P, Fortiana J (2010). “Distance-based local linear regression for functional predictors.” *Computational Statistics & Data Analysis*, **54**(2), 429–437.
- Burba F, Ferraty F, Vieu P (2009). “ k -nearest neighbour method in functional nonparametric regression.” *Journal of Nonparametric Statistics*, **21**(4), 453–469.
- Chib S (1995). “Marginal likelihood from the Gibbs output.” *Journal of the American Statistical Association*, **90**(432), 1313–1321.
- Ferraty F, Van Keilegom I, Vieu P (2010). “On the validity of the bootstrap in non-parametric functional regression.” *Scandinavian Journal of Statistics*, **37**(2), 286–306.
- Ferraty F, Vieu P (2006). *Nonparametric Functional Data Analysis: Theory and Practice*. Springer, New York.
- Gelfand AE, Dey DK (1994). “Bayesian model choice: Asymptotics and exact calculations.” *Journal of the Royal Statistical Society, Series B*, **56**(3), 501–514.
- Geweke JF (1999). “Using simulation methods for Bayesian econometric models: Inference, development, and communication.” *Econometric Reviews*, **18**(1), 1–73.
- Kass RE, Raftery AE (1995). “Bayes factors.” *Journal of the American Statistical Association*, **90**(430), 773–795. ISSN 0162-1459.
- Meyer R, Yu J (2000). “BUGS for a Bayesian analysis of stochastic volatility models.” *Econometrics Journal*, **3**(2), 198–215.
- Newton MA, Raftery AE (1994). “Approximate Bayesian inference with the weighted likelihood bootstrap.” *Journal of the Royal Statistical Society, Series B*, **56**(1), 3–48.
- Roeder K, Wasserman L (1997). “Practical Bayesian density estimation using mixtures of normals.” *Journal of the American Statistical Association*, **92**(439), 894–902.
- Shang HL (2013a). “Bayesian bandwidth estimation for a nonparametric functional regression model with unknown error density.” *Computational Statistics & Data Analysis*, **67**, 185–198.
- Shang HL (2013b). “Bayesian bandwidth estimation for a semi-functional partial linear regression model with unknown error density.” *Computational Statistics*, **in press**.

Affiliation:

Han Lin Shang
 ESRC Centre for Population Change
 University of Southampton
 SO17 1BJ, United Kingdom
 E-mail: H.Shang@soton.ac.uk
 URL: <https://sites.google.com/site/hanlinshangwebsite/>