

Online supplemental material of the article:

’Practical Hilbert space approximate Bayesian Gaussian
processes for probabilistic programming’

by

Gabriel Riutort-Mayol and Paul-Christian Bürkner and Michael R. Andersen and
Arno Solin and Aki Vehtari

This document consists of four case studies of application of the article ’Practical Hilbert space approximate Bayesian Gaussian processes for probabilistic programming’.

Contents

A Same-sex marriage case study

This data set relates the proportion of support for same-sex marriage to the age. The data consists of 74 observations of the amount of people y_i supporting same-sex marriage from a population n_i per age group i ($i = 1, \dots, 74$). The observational model is a binomial model with parameters population n_i and probability of supporting same-sex marriage p_i per age group i ,

$$y_i \sim \text{Binomial}(p_i, n_i).$$

The population per age group n_i is a known quantity and the goal is to estimate the same-sex support probability p_i or mean number of support people per age group. Probabilities $\mathbf{p} = (p_1, \dots, p_{74})$ are modeled by a Gaussian process (GP) function $f : \mathbb{R} \rightarrow \mathbb{R}$ with a squared exponential covariance function k , as a function of age input values $\mathbf{x} = (x_1, \dots, x_{74})$, and through the *logit* function as a link function,

$$p_i = \text{logit}(f(x_i))$$
$$f(x) \sim \mathcal{GP}(0, k(x, x', \theta)).$$

Saying that the function $f(\cdot)$ follows a GP model is equivalent to say that \mathbf{f} is multivariate Gaussian distributed with covariance matrix \mathbf{K} , where $K_{ij} = k(x_i, x_j, \theta)$, with $i, j = 1, \dots, 74$. The covariance function k depends on the inputs \mathbf{x} and hyperparameters $\theta = \{\alpha, \ell\}$. The hyperparameters α and ℓ represent the marginal variance and lengthscale, respectively, of the GP process.

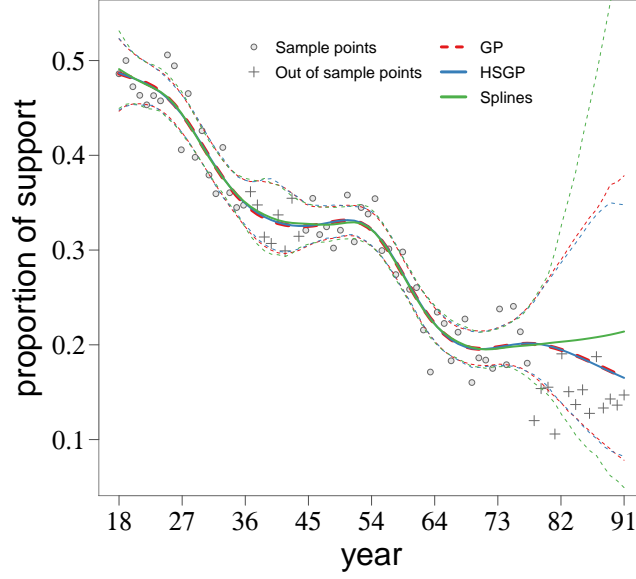


Figure A.1: Posterior predictive means of the proposed HSGP model, the regular GP model, and the Splines model. 95% confident intervals are plotted as dashed lines.

In the HSGP model, the function $f(x)$ is approximated as in equation (7), with the squared exponential spectral density as in equation (1), and eigenvalues λ_j and eigenfunctions ϕ_j as in equations (5) and (6).

In order to do model comparison, in addition to the regular GP model and HSGP model, an splines-based model is also fitted using the Thin Plate Regression Splines approach in ? and implemented in the R-package *mgcv* (?). A Bayesian approach is used to fit this splines model using the R-package *brms* (?).

Figure ?? shows the posterior mean predictive distributions of the three models, the regular GP, the HSGP model with $m = 20$ basis functions and boundary factor $c = 1.5$, and the splines model with 20 knots. Sample observations are plotted as circles in the figure, and the out-of-sample observations, which have been used for testing, are plotted as crosses.

For the HSGP model, different models with different number of basis functions and boundary factor have been fitted. The root mean square errors (RMSE) for every one of these models have been computed against the regular GP model, and plotted as a function of the number of basis functions m and boundary factor c in Figure ??-left, for sample data, and in Figure ??-right, for test data. As the boundary factor increases, more basis functions are needed for a accurate approximation.

Figure ?? shows the RMSE of the regular GP, HSGP and splines models computed against the actual data, for training and test data, as a function of the number of basis functions m and boundary factor c for the HSGP model, and knots for the splines model. We can see how the splines models do not extrapolate data properly (Figure ??-right).

Figure ?? shows computational times, in seconds per iteration (iteration of the HMC sampling

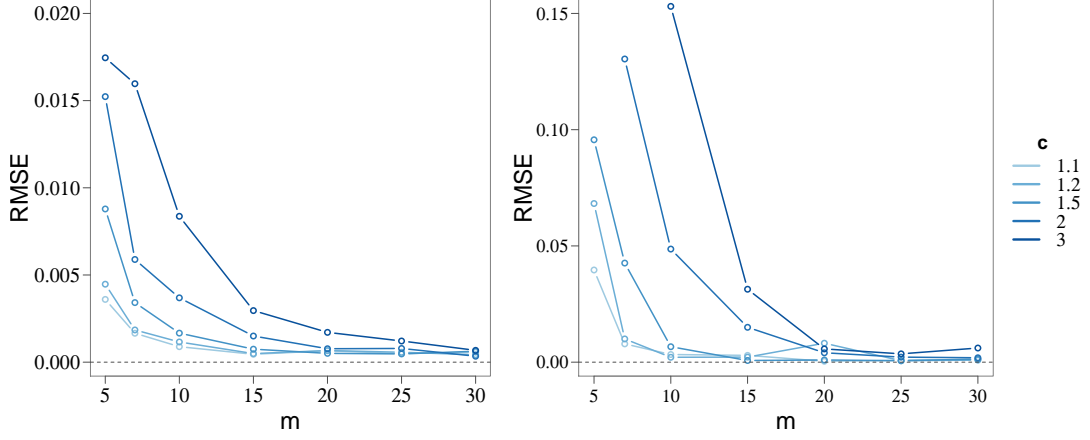


Figure A.2: Root mean square error (RMSE) of the HSGP model, computed against the regular GP model, as a function of the number of basis functions m and boundary factor c . RMSE for sample data (left) and RMSE for out-of-sample data (right).

method), as a function of the number of basis functions m , for the HSGP model, and knots, for the splines model. The computational times is represented in the y-axis which is on a logarithmic scale. The HSGP model is on average roughly 15 times faster than the regular GP and 5 times faster than the spline model, for this particular case and univariate input space. Computation time increases relatively slight with the number of basis functions in a univariate input space, as can be seen in the figure.

The Stan model codes for the exact GP, the approximate GP and the splines models of this case study can be found online at https://github.com/gabriuma/basis_functions_approach_to_GP/tree/master/Paper/Case-study_Same-sex-marriage-data.

B Case study of simulated data for a bivariate function

This example consists of a simulated dataset with $n = 120$ ($i = 1, \dots, n$) single draws from a GP prior as function of input values $\mathbf{x}_i \in \{[-1, 1], [-1, 1]\} \subset \mathbb{R}^2$ in a bivariate input space ($D = 2$). A squared exponential covariance function, with hyperparameters marginal variance $\alpha = 1$ and lengthscales $\ell_1 = 0.10$, for the first input dimension, and $\ell_2 = 0.35$, for the second input dimension, is used for the GP prior. Gaussian noise standard deviation $\sigma = 0.2$ was added to the GP draws to form the final noisy set of observations $\mathbf{y} \in \mathbb{R}^{120}$.

The regular GP model over the outcome variable \mathbf{y} can be written as follows,

$$\begin{aligned} \mathbf{y} &= \mathbf{f} + \boldsymbol{\epsilon} \\ \boldsymbol{\epsilon} &\sim \text{Normal}(0, \sigma^2 I) \\ f(\mathbf{x}) &\sim \mathcal{GP}(0, k(\mathbf{x}, \mathbf{x}', \theta)), \end{aligned}$$

where $\mathbf{f} = \{f(\mathbf{x}_i)\}_{i=1}^{120}$ represents the underlying function at the input values $\mathbf{x}_i \in \mathbb{R}^2$, $\boldsymbol{\epsilon}$ is the Gaussian noise term with variance σ^2 , and I represents the identity matrix. The function

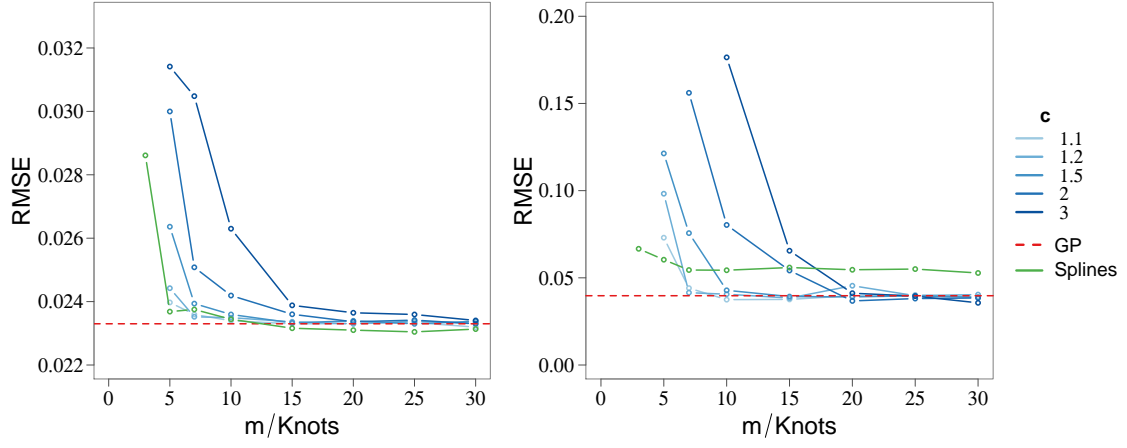


Figure A.3: Root mean square error (RMSE) of the different methods, regular GP, HSGP and splines models, computed against the actual data, as a function of the number of basis functions m and boundary factor c for the HSGP model, and knots for the splines model. RMSE for sample data (left) and RMSE for out-of-sample data (right).

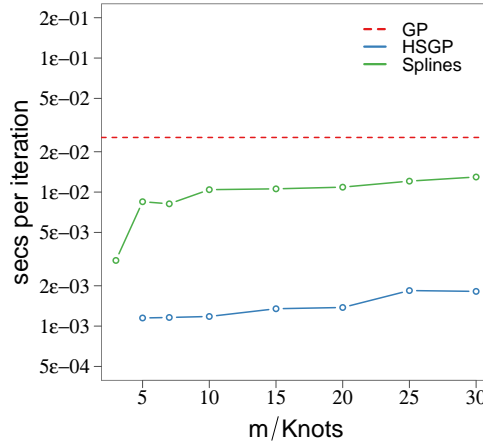


Figure A.4: Computational time (y-axis), in seconds per iteration (iteration of the HMC sampling method), as a function of the number of basis functions m and knots. The y-axis is in a logarithmic scale.

$f : \mathbb{R}^2 \rightarrow \mathbb{R}$ is a GP prior with a multivariate squared exponential covariance function $k(\mathbf{x}_i, \mathbf{x}_j, \theta)$, which depends on the inputs \mathbf{x} and hyperparameters $\theta = \{\alpha, \ell_1, \ell_2\}$. The hyperparameters α , ℓ_1 and ℓ_2 represent the marginal variance and lengthscales for first and second input dimensions, respectively, of the GP process. Saying that the function $f(\cdot)$ follows a GP model is equivalent to say that \mathbf{f} is multivariate Gaussian distributed with covariance matrix \mathbf{K} , where $K_{ij} = k(\mathbf{x}_i, \mathbf{x}_j, \theta)$, with $i, j = 1, \dots, 120$.

The marginalized form, by integrating out the latent values \mathbf{f} , of the previous latent GP model results:

$$\mathbf{y} \sim \text{Normal}(0, \mathbf{K} + \sigma^2 \mathbf{I}).$$

In the HSGP model with 2 input dimensions, the latent function $f(\mathbf{x})$ is approximated as in equation (12), with the 2-dimensional squared exponential spectral density S as in equation (1), and the 2-vector of eigenvalues λ_j and the multivariate eigenfunctions ϕ_j as in equations (9) and (10), respectively.

In order to do model comparison, in addition to the regular GP and HSGP models, a two-dimensional splines-based model is also fitted using a cubic spline basis, penalized by the conventional integrated square second derivative cubic spline penalty (?), and implemented in the R-package *mgcv* (?). A Bayesian approach is used to fit this spline model using the R-package *brms* (?).

Figure ?? shows the data-generating GP function, from where the dataset was drawn, and the mean posterior predictive functions of the three models, regular GP, HSGP, and splines, fitted over the dataset. Sample observations are also plotted as circles. For the HSGP model, $m_1 = 40$ and $m_2 = 40$ basis functions for each dimension respectively, were used, which lead to a total of 1600 multivariate basis functions. A boundary factor for each dimension $c_1 = 1.5$ and $c_2 = 1.5$ were used. For the splines model, 40 knots in each dimension were used.

Figure ?? shows the difference functions between the data-generating function and the GP, HSGP and splines models, respectively.

In order to assess performance of the models as a function of the number of basis functions and knots, different models with different number of basis functions, for the HSGP model, and different number of knots, for the splines model, have been fitted. In all models, the same number of basis functions and knots per dimension were used. Figure ??-left shows the root mean squared error (RMSE), computed against the data-generating function, as a function of the boundary factor c , and the number of univariate basis functions m , for the HSGP model, and knots, for the splines model. From Figures ?? and ??-left, it can be seen a close approximation of the HSGP model to the regular GP model. However, the performance of the splines model is significantly worse.

Figure ??-right shows the computational times of the different models as a function of the boundary factor, number of basis functions and knots. Figure ?? reveals that choosing the optimal boundary factor allows for less number of basis functions and less computation time. Even though in a bivariate input space the computation demand increases significantly with the number of dimensions or knots, either the HSGP or spline models work significantly better than regular GP, even for highly wiggly functions that require a high number of basis functions or knots for an accurate approximation. However, serious difficulties with computation time were encountered in building the spline model with 50 knots for this bivariate function.

The Stan model codes for the exact GP, the approximate GP and the splines models of this case study

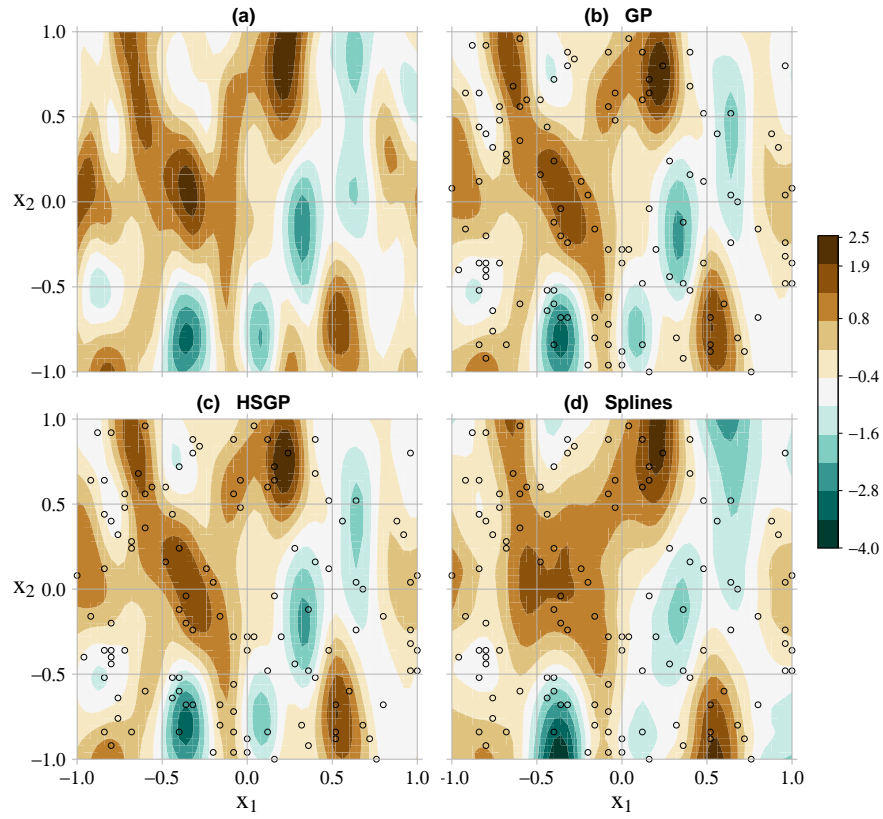


Figure B.1: Data-generating function (a) and posterior predictive mean functions of the GP (b), HSGP (c) and spline (d) models. Sample points are plotted as circles

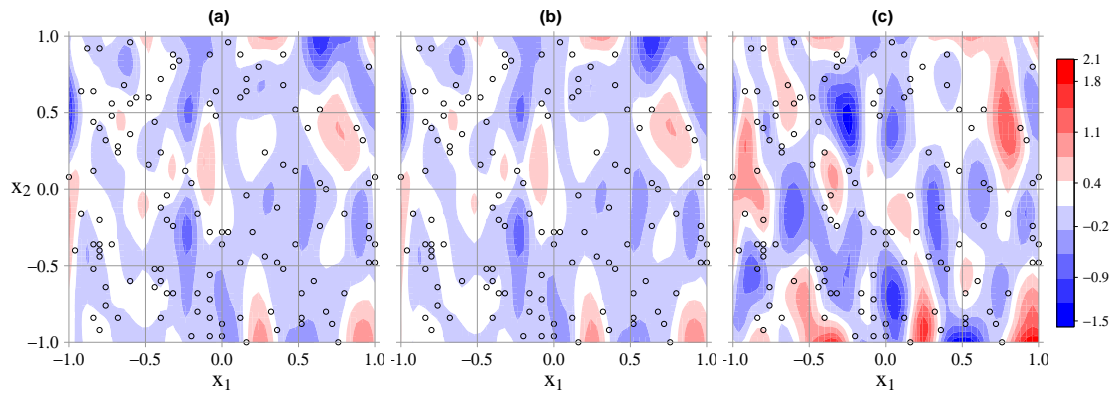


Figure B.2: Error between the data-generating function and posterior mean functions of the GP (a), HSGP (b) and spline (c) models. Sample points are plotted as circles.

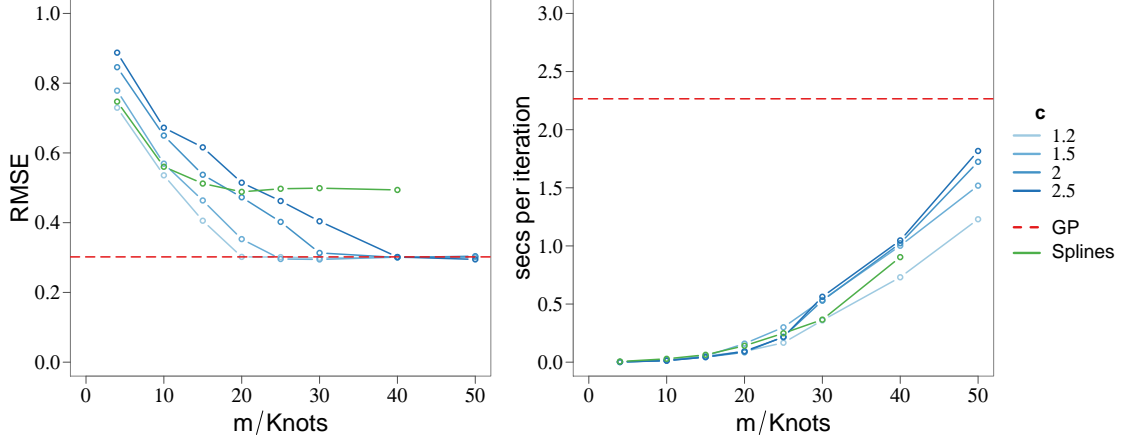


Figure B.3: Root mean square error (RMSE) (left) and computational time (right) in seconds per iteration (iteration of the HMC sampling method) of the different methods computed against the data-generating function, as a function of the boundary factor c , number of basis functions m and knots.

can be found online at https://github.com/gabriuma/basis_functions_approach_to_GP/tree/master/Paper/Case-study_2D-Simulated-data.

C Diabetes case study

The next example presents an epidemiological study of diabetes disease. The study aims to relate the probability of suffering from diabetes to some risk factors. The data contains $n = 392$ individuals ($i = 1, \dots, n$) from which the binary variable of suffering ($y_i = 1$) or not suffering ($y_i = 0$) from diabetes have been observed. The input vector $\mathbf{x}_i = (x_{i1}, x_{i2}, x_{i3}, x_{i4}) \in \mathbb{R}^4$, in a 4D input space ($D = 4$), contains the risk factors, *Glucose* (x_{i1}), *Pregnancy* (x_{i2}), *Age* (x_{i3}) and *BMI* (x_{i4}), per individual i . The observational model is a Bernoulli model with parameter the probability p_i of suffering from diabetes per observation i ,

$$y_i \sim \text{Bernoulli}(p_i).$$

The goal is to estimate the probability p_i as a function of the risk factors, which function $f(\cdot) : \mathbb{R}^4 \rightarrow \mathbb{R}$ is modeled as a GP with a multivariate squared exponential covariance function k depending on the risk factors \mathbf{x} and hyperparameters $\theta = \{\alpha, \ell\}$, and related to the probabilities p_i through the *logit* link function,

$$p_i = \text{logit}(f(\mathbf{x}_i))$$

$$f(\mathbf{x}) \sim \mathcal{GP}(0, k(\mathbf{x}, \mathbf{x}', \theta)).$$

Saying that the function $f(\cdot)$ follows a GP model is equivalent to say that $\mathbf{f} = \{f(\mathbf{x}_i)\}_{i=1}^n$ are multivariate Gaussian distributed with covariance matrix \mathbf{K} , where $K_{ij} = k(\mathbf{x}_i, \mathbf{x}_j, \theta)$, with $i, j = 1, \dots, n$. The hyperparameters α and ℓ represent the marginal variance and lengthscale,

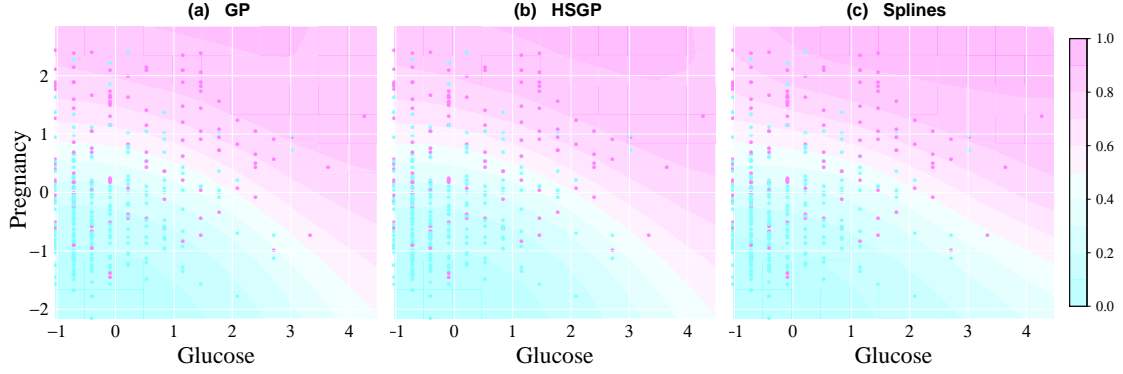


Figure C.1: Posterior predictive mean functions of the GP (a), HSGP (b) and spline (c) models. Samples observations of suffering (red points) and not suffering (blue points) from the disease are plotted.

respectively, of the GP process. A scalar lengthscale has been considered in the multivariate covariance function k .

In the HSGP model with D input dimensions, the function $f(\mathbf{x})$ evaluated at input vector $\mathbf{x} \in \mathbb{R}^D$ is approximated as in equation (12), with the D -dimensional (with a scalar lengthscale) squared exponential spectral density S as in equation (1) and the D -vector of eigenvalues λ_j and the multivariate eigenfunctions ϕ_j as in equations (9) and (10), respectively.

In order to do model comparison, in addition to the regular GP and HSGP models, a D -dimensional spline-based model is also fitted using a cubic spline basis penalized by the conventional integrated square second derivative cubic spline penalty (?) and implemented in the R-package *mgcv* (?). A Bayesian approach is used to fit this spline model using the R-package *brms* (?).

Figure ?? shows the posterior mean predictions of probabilities (p_i) of the three models, regular GP, HSGP and spline, fitted over the dataset with the 2 input dimensions *Glucose* and *Pregnancy* ($D = 2$). The binary observations y_i are also plotted in the plots as colored points. For the HSGP model, $m_1 = 20$ and $m_2 = 20$ basis functions for each dimension, respectively, were used, which lead to a total of 400 multivariate basis functions. A boundary factor for each dimension $c_1 = 4$ and $c_2 = 4$ were used. For the spline model, 20 knots per dimension were used.

In order to assess the performance of the models as a function of the boundary factor, the number of basis functions and knots, different models with different number of basis functions and boundary factor for the HSGP model and different number of knots for the spline model have been fitted. In all models, the same boundary factor, number of basis functions and knots per dimension were used. Figure ?? shows the expected log predictive density (ELPD; see ?) as a function of the boundary factor c and the number of univariate basis functions m , for the HSGP model, and knots, for the spline model. The ELPD is computed over the actual observations by cross-validation. Basically, with slightly differences, all models show similar performances, due to the fact that the process is very smooth with a relatively very large lengthscale estimate $\ell = 4.51$. Even though, a slight pattern of performance improvement can be appreciated as the boundary factor c increases, which fact is because small boundary factors are not allowed when large lengthscales (Figure 6).

Figure ?? shows the computational times of the different models, regular GP, HSGP and spline, fitted over the dataset with 2, 3 and 4 input dimensions, as a function of the boundary factor c and

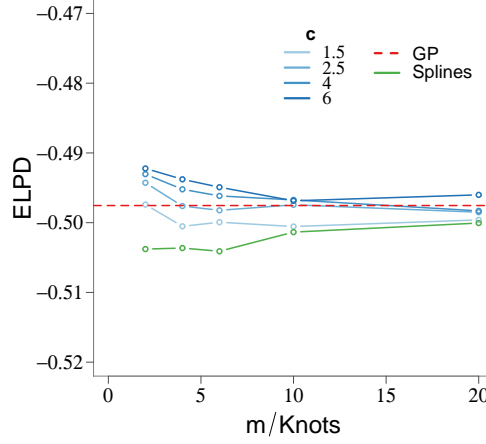


Figure C.2: Expected log predictive density (ELPD) of the different methods as a function of the boundary factor c and the number of basis functions m , for the HSGP model, and knots, for the spline model.

number of univariate basis functions m and knots. The time of omputation increases significantly with higher dimensions for the HSGP and spline models. In this sense, choosing optimal values for the number of basis functions and boundary factor, looking at the recommendations and diagnosis provided by Figure 6, can be essential to avoid a excessive computation time, especially in multivariate input spaces. Serious difficulties with computation time were encountered in building spline models in a 4D input space or even in a 3D input space with more than 10 knots.

The Stan model codes for the exact GP, the approximate GP and the spline models of this case study can be found online at https://github.com/gabriuma/basis_functions_approach_to_GP/tree/master/Paper/Case-study_Diabetes-data.

D Spatio-temporal land-use classification case study

The next example presents an spatio-temporal classification in land-use of plots between 2006 and 2015 in a part of the territory of Valencia in Spain dedicated to growing citrus fruits. A sampling set consists of $n = 200$ plots with known class. The data is recorded in a time series of $T = 5$ years (2006, 2008, 2010, 2012, 2015) within the period. The class of each parcel i and time t is stored by a categorical variable y_{it} representing the $K = 5$ different possible classes ($k = 1, \dots, K$): $k = 1$, adult independent citrus fruits; $k = 2$, aligned citrus fruits; $k = 3$, irregular citrus fruits; $k = 4$, abandoned citrus fruits; $k = 5$, young citrus fruits.

A bunch of 52 characteristic variables was available for every parcel and time. These variables were computed from satellite color images and cadastral map by using the software FETEX for automatic descriptive feature extraction from image-objects (?). These variables concern spectral intensities and empirical semivariogram of the pixels within a plot, as well as descriptive statistics of the shape of the plots.

Due to the fact that 52 input variables are too high-dimensional for a multivariate HSGP model, which computational cost scales as $O(nm^D + m^D)$, with m the number of basis functions and

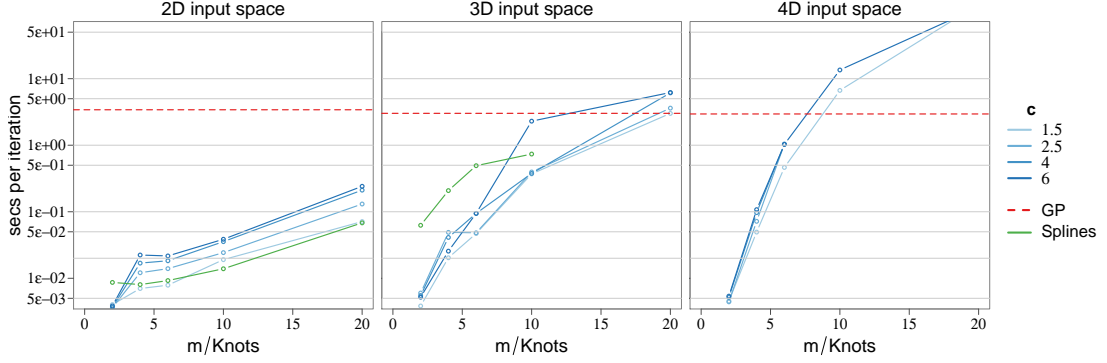


Figure C.3: Time of computation in seconds per iteration (iteration of the HMC sampling method) of the different models fitted over the dataset, with 2 input dimensions (left) and 3 input dimensions (center) and 4 input dimensions (right), as a function of the boundary factor c and number of basis functions m , for the HSGP model, and knots, for the spline model. The y-axis is on a logarithmic scale.

D the number of input variables, the multivariate HSGP model will be formulated as an additive HSGP model.

As it is known, the computational demand of a multivariate HSGP model component increases quickly with the number of dimensions, so we should avoid high-dimensional HSGP components in the additive model. Original input variables are highly correlated, which would imply the use of high-order interaction components in the additive model to achieve accurate model performance. Therefore, instead of using the original variables as inputs, we use their principal components, which are expected to be linearly uncorrelated. Using the principal components as inputs helps, in principle, not to have to use as many high-order interaction components in the additive model.

The principal components (PCs) will be used jointly with the time variable as inputs in the classifying additive HSGP model. Let's denote the matrix $\mathbf{X} = [\mathbf{x}_{11} \cdots \mathbf{x}_{it} \cdots \mathbf{x}_{nT}]^\top \in \mathbb{R}^{nT \times D}$, which contains the input vectors $\mathbf{x}_{it} \in \mathbb{R}^D$, $D = 53$ (52 PCs plus time) for the spatio-temporal observations (plots $i = 1, \dots, 200$, and times $t = 1, \dots, 5$).

The observational model is a multinomial model with parameters the vector of probabilities $\mathbf{p}_{it} = (p_{it,1}, p_{it,2}, \dots, p_{it,K})$, where $p_{it,k}$ is the probability of belonging to class k per parcel i and time t ,

$$y_{it} \sim \text{Multinomial}(\mathbf{p}_{it}).$$

The goal is to estimate the vector of probabilities \mathbf{p}_{it} as a function of the predictors, which is a multivariate function $f(\mathbf{x}_{it}) : \mathbb{R}^D \rightarrow \mathbb{R}^K$,

$$f(\mathbf{x}_{it}) = (f_1(\mathbf{x}_{it}), \dots, f_K(\mathbf{x}_{it})),$$

which is related to the vector of probabilities \mathbf{p}_{it} through the 'softmax' link function:

$$\mathbf{p}_{it} = \text{softmax}(f(\mathbf{x}_{it})).$$

Each individual function $f_k(\mathbf{x})$ is modeled as a first-order additive model plus the second-order

additive effects between time input variable (x^D) and all the other inputs as follows:

$$f_k(\mathbf{x}) = \sum_{d=1}^D g_d(x^d) + \sum_{d=1}^{D-1} h_{d,D}(x^d, x^D). \quad (\text{D.1})$$

The first-order components $\{g_d(x^d)\}_{d=1}^D$ in equation (??) are modeled as unidimensional HSGP models:

$$g_d(x^d) \sim \mathcal{HSGP}(x^d, S, \theta_{d,1}).$$

In the HSGP model, a first-order components $g_d(x^d)$, evaluated at input value $x^d \in \mathbb{R}$, is approximated as in equation (7) with the squared exponential spectral density S as in equation (1) and eigenvalues λ_j and eigenfunctions ϕ_j as in equations (5) and (6), respectively.

The second-order components $\{h_{d,D}(x^d, x^D)\}_{d=1}^{D-1}$ in equation (??) are modeled as two-dimensional HSGP models:

$$h_{d,D}(x^d, x^D) \sim \mathcal{HSGP}(x^d, x^D, S, \theta_{d,D}).$$

In the HSGP model, a second-order component $h_{d,D}(x^d, x^D)$, evaluated at inputs $x^d \in \mathbb{R}$ and $x^D \in \mathbb{R}$, is approximated as in equation (12) with the two-dimensional (with a scalar lengthscale) squared exponential spectral density S as in equation (1) and the D -vector of eigenvalues λ_j and the multivariate eigenfunctions ϕ_j as in equations (9) and (10), respectively.

The vector of hyperparameters $\theta_{d,1} = (\alpha_{d,1}, \ell_{d,1})$ contains the marginal variance $\alpha_{d,1}$ and lengthscale $\ell_{d,1}$ of the $g_d(x^d)$ model component. And, the vector of hyperparameters $\theta_{d,D} = (\alpha_{d,D}, \ell_{d,D})$ contains the marginal variance $\alpha_{d,D}$ and lengthscale $\ell_{d,D}$ of the $h_{d,D}(x^d, x^D)$ model component.

For the first-order components $g_d(x^d)$, $m = 15$ basis functions and a boundary factor $c = 2.5$ were used. For the second-order components $h_{d,D}(x^d, x^D)$, $m_1 = 15$ and $m_2 = 15$ basis functions for each dimension, respectively, were used, which lead to a total of 225 multivariate basis functions. A boundary factor for each dimension $c_1 = 2.5$ and $c_2 = 2.5$ were used. All the input variables were previously standardized.

In the case of the first-order components, the normalized lengthscale estimates $\left(\frac{2 \cdot \hat{\ell}_{d,1}}{|x_{max}^d - x_{min}^d|}\right)$ are all bigger than the minimum lengthscale reported by Figure (6) as a function of m and c . Which means that the used number of basis functions ($m = 15$) and boundary factor ($c = 2.5$) are suitable values for modeling accurately the input effects.

For the second-order components, the relationships between the number of basis functions, the boundary factor and the lengthscale is not available for the multivariate case. However, we can approximately analyze the lengthscale estimates of the second-order HSGP components analyzing each dimension separately as unidimensional HSGP models.

Table ?? shows the confusion matrix after fitting the model following a Q -fold cross-validation procedure, with $Q = 100$, over the training data. Thus, every fold contains 10 observations. The confusion matrix evaluates the rate of misclassification per class. Columns represent the true classes and rows represent the estimated classes. The values within the matrix correspond to the number of items that fall into every cell. The marginals of the columns (true classes) represent the percentage of misclassified items in relation to the 'truth', commonly known as the *omission error*. And the marginals of the rows (estimated classes) represent the percentage of misclassified items in relation

True Estimate	k = 1	k = 2	k = 3	k = 4	k = 5	
k = 1	90	39	14	3	11	42%
k = 2	46	301	8	2	3	16%
k = 3	8	4	59	4	1	19%
k = 4	5	2	6	342	5	5%
k = 5	8	2	1	0	38	19%
	42%	13%	32%	2%	34%	17%

Table D.1: Confusion matrix after the Q-fold cross-validation procedure over the training data.

to the estimates (classifier), commonly known as the *commission error*. The percentage in the down right cell of the matrix is the overall mean misclassification rate. As can be seen, there exist a high misclassification rate between classes $k = 1$ and $k = 2$, between classes $k = 1$ and $k = 3$, and between classes $k = 1$ and $k = 5$.

The Stan model code for the approximate GP model of this case study can be found at https://github.com/gabriuma/basis_functions_approach_to_GP/tree/master/Paper/Case-study_Land-use-classification.

References

- Paul-Christian Bürkner. brms: An R package for Bayesian multilevel models using Stan. *Journal of Statistical Software*, 80(1):1–28, 2017.
- Luis Ángel Ruiz, Jorge Abel Recio, Alfonso Fernández-Sarría, and Txomin Hermosilla. A feature extraction software tool for agricultural object-based image analysis. *Computers and Electronics in Agriculture*, 76(2):284–296, 2011.
- Aki Vehtari and Janne Ojanen. A survey of Bayesian predictive methods for model assessment, selection and comparison. *Statistics Surveys*, 6:142–228, 2012.
- Simon N Wood. Thin plate regression splines. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 65(1):95–114, 2003.
- Simon N Wood. Fast stable restricted maximum likelihood and marginal likelihood estimation of semi-parametric generalized linear models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 73(1):3–36, 2011.
- Simon N Wood. *Generalized additive models: an introduction with R*. Chapman and Hall/CRC, 2017.