Practical Hilbert space approximate Bayesian Gaussian processes for probabilistic programming

Gabriel Riutort-Mayol [1], Paul-Christian Burkner[23], Michael R. Andersen[4], Arno Solin[3], Aki Vehtari[3]

[1]FISABIO-Public Health, Spain
[2]Excellence Cluster for Simulation Technology, University of Stuttgart, Germany
[3]Department of Computer Science, Aalto University, Espoo, Finland
[4]Department of Applied Mathematics and Computer Science, Technical University of Denmark, Lyngby, Denmark

November 8, 2024

Presented by Christine Shen

Outline of the presentation

1. What is a Gaussian Process (GP)
2. How to use GP in Bayesian spatial models
3. Hilbert space methods for reduced-rank Gaussian process regression (HSGP)
4. How to implement HSGP
5. Code demo and simulation results
6. Conclusion

What is a Gaussian Process

A Gaussian Process (GP) is

- a stochastic process, i.e., a collection of random variables indexed by $\mathbf{u} \in \mathcal{U}$, where any finite number of them have a joint Gaussian distribution
- an extension of the multivariate Gaussian to infinite dimensions.
  - E.g., we are familiar with random variables indexed by integers: $X_1, \ldots, X_n \sim N(\boldsymbol{\mu}, \Sigma)$
  - Now consider random variables $\{X(\mathbf{u})\}$, indexed by time, or location (spatial), or time and location (spatial-temporal)...

How to use GP in Bayesian spatial models

Consider data on n patients who have had upper extremety fractures. For each patient, we know their gender, age, address, and time to readmission since initial fracture (assume no censoring). We are interested in studying the geospatial pattern of patients' readmission risks, controlling for gender and age. We posit the following Bayesian model

$$\log y_i = \mathbf{x}_i^T \boldsymbol{\beta} + \boldsymbol{\theta}(\mathbf{u}_i) + \epsilon_i, \quad \text{where}$$

- y_i is the time to readmission for patient i
- $\mathbf{x}_i \in \mathbb{R}^3$ are the covariates for patient i including an intercept, gender, and age
- $\mathbf{u}_i \in \mathbb{R}^2$ is the location for patient i
- $\boldsymbol{\theta}(\mathbf{u}_i)$ is the spatial intercept, centered at 0
- $\epsilon_i \sim$ i.i.d. $N(0, \sigma^2)$ is noise
- Priors $\pi(\boldsymbol{\beta}) \sim N(0, \sigma_0^2 \mathbf{I})$, $\pi(\sigma^2) \sim IG(a, b)$

How to use GP in Bayesian spatial models

Recall the model

$$\log y_i = \mathbf{x}_i^T \boldsymbol{\beta} + \boldsymbol{\theta}(\mathbf{u}_i) + \epsilon_i.$$

How to use $\boldsymbol{\theta}(\mathbf{u}_i)$ to study the geospatial pattern of readmission risk?

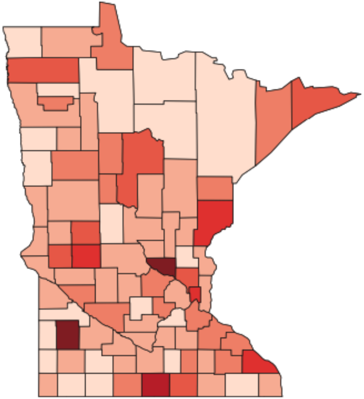
How to use GP in Bayesian spatial models

Recall the model

$$\log y_i = \mathbf{x}_i^T \boldsymbol{\beta} + \boldsymbol{\theta}(\mathbf{u}_i) + \epsilon_i.$$

How to use $\boldsymbol{\theta}(\mathbf{u}_i)$ to study the geospatial pattern of readmission risk?

1. if $\mathbf{u}_i$'s are areal data, i.e., $\mathbf{u}_i = \mathbf{q}_j$, $j \in \{1, \ldots, n_q\}$ are at zip code/ census block group/ census tract level, we can
    - model as random intercepts $\boldsymbol{\theta}(\mathbf{q}_j) \sim$ i.i.d. $N(0, \tau^2)$, or
    - use a CAR (conditional autoregressive) model



How to use GP in Bayesian spatial models

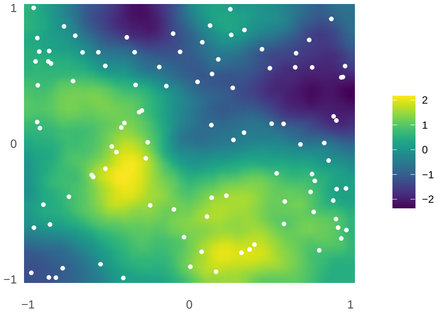
Recall the model

$$\log y_i = \mathbf{x}_i^T \boldsymbol{\beta} + \boldsymbol{\theta}(\mathbf{u}_i) + \epsilon_i.$$

How to use $\boldsymbol{\theta}(\mathbf{u}_i)$ to study the geospatial pattern of readmission risk?

1. if $\mathbf{u}_i$'s are areal data, i.e., $\mathbf{u}_i = \mathbf{q}_j$, $j \in \{1, \ldots, n_q\}$ are at zip code/ census block group/ census tract level, we can
   - model as random intercepts $\boldsymbol{\theta}(\mathbf{q}_j) \sim$ i.i.d. $N(0, \tau^2)$, or
   - use a CAR (conditional autoregressive) model
2. if $\mathbf{u}_i$'s are point-referenced data, i.e., we know the exact address/ longitude and latitude of each patient, we can consider using a GP prior:

$$\boldsymbol{\theta}(\mathbf{d}) \sim GP(0, k(\mathbf{u}, \mathbf{u}')).$$



Why we use GP for point-referenced spatial data

Flexibility

1. GP provides a prior distribution over function spaces for **non-parametric** latent functions.
2. We can use different covariance functions to characterize a wide variety of spatial dependence structures.

A GP can be fully specified by the mean function and covariance function. Consider

$$\boldsymbol{\theta}(\mathbf{u}) \sim GP(\mu(\mathbf{u}), \Sigma(\mathbf{u})), \quad \mathbf{u} \in \mathbb{R}^2 \text{ are locations}$$

- Each realization of $\boldsymbol{\theta}(\mathbf{u})$ is a surface
- $\mu(\mathbf{u})$ controls the level of the surface, typically set to 0 and modelled separately
- $\Sigma(\mathbf{u})$ controls covariance between different locations, continuity and smoothness of the surface

The Matérn covariance function

The Matérn covariance function is frequently used in spatial statistics. Let $\mathbf{r} = \mathbf{u} - \mathbf{u}'$, it is defined as

$$k(\mathbf{u}, \mathbf{u}') = k(\|\mathbf{r}\|) = \alpha^2 \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\sqrt{2\nu} \frac{\|\mathbf{r}\|}{l}\right)^{\nu} K_{\nu} \left(\sqrt{2\nu} \frac{\|\mathbf{r}\|}{l}\right), \quad \text{where}$$

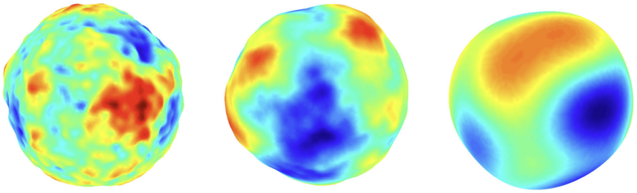
- K_ν is the modified Bessel function of the second kind, α is the GP scale, l is the lengthscale
- $k(\|\mathbf{r}\|)$ is $\lceil \nu \rceil - 1$ times differentiable, i.e., ν controls smoothness of the kernel.

(a) $\nu = 1/2$, exponential kernel

$$k(\mathbf{u}, \mathbf{u}') = \alpha^2 \exp\left(-\frac{1}{2} \frac{\|\mathbf{u} - \mathbf{u}'\|}{l}\right)$$

(c) $\nu = \infty$, squared exponential kernel

$$k(\mathbf{u}, \mathbf{u}') = \alpha^2 \exp\left(-\frac{1}{2} \frac{\|\mathbf{u} - \mathbf{u}'\|^2}{l^2}\right)$$



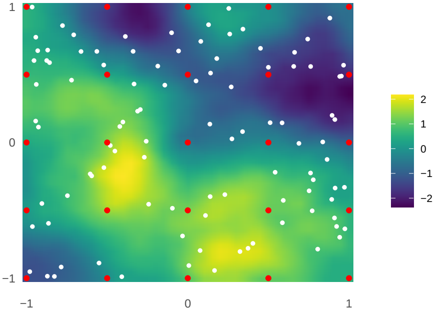
(a) $\nu = \frac{1}{2}$ and $\ell = 0.5$ **(b)** $\nu = \frac{3}{2}$ and $\ell = 0.5$ **(c)** $\nu \to \infty$ and $\ell = 0.5$

Why we use GP for point-referenced spatial data

Mathematical tractability

Recall Gaussian distribution is closed under linear transformation and conditioning, similarly,

1. GP is closed under linear operators
2. GP is closed under Bayesian conditioning, i.e., if the prior $f(\cdot)$ is a GP, the posterior $f(\cdot \mid \mathbf{y})$ is also a GP, with updated parameters
3. Out-of-sample prediction is natural under GP, with proper uncertainty quantification



How to fit a GP regression model

Consider the following Bayesian model:

$$\log y_i = \mathbf{x}_i^T \boldsymbol{\beta} + \boldsymbol{\theta}(\mathbf{u}_i) + \epsilon_i, \quad \boldsymbol{\epsilon} \sim \ N(0, \sigma^2 \mathbf{I}),$$

with priors:

- $\boldsymbol{\theta}(\mathbf{u}) \sim GP(0, K_{3/2}(\mathbf{u}, \mathbf{u}' \mid \alpha, l))$, where $K_{3/2}$ denotes the Matérn 3/2 kernel
- $\boldsymbol{\beta} \sim N(0, 25\mathbf{I})$, $\sigma \sim TN(0, 2)$, $\alpha \sim TN(0, 4)$, $l \sim IG(2, 1)$, where TN denotes truncated normal

How to fit this model:

- Gibbs sampling if conjugate priors are used
- Probabilistic programming platforms such as Stan, PyMC, or NumPyro

Computational issues with GP

GP is not scalable for a full Bayesian analysis, because for a dataset of size n,

$$\begin{pmatrix} \boldsymbol{\theta}(\mathbf{u}_1) \\ \vdots \\ \boldsymbol{\theta}(\mathbf{u}_n) \end{pmatrix} \sim N(0, \Sigma), \quad \Sigma \in \mathbb{R}^{n \times n}.$$

I.e., we need to work with an $n \times n$ covariance matrix for each MCMC iteration, which scales as $\mathcal{O}(n^3)$ for computational complexity, and $\mathcal{O}(n^2)$ for memory storage.

Hilbert space methods for reduced-rank Gaussian process regression

[Solin and Särkkä, 2020] introduced a Hilbert space method for reduced-rank Gaussian process regression (HSGP), based on an approximate series expansion of the covariance function in terms of an eigenfunction expansion of the Laplace operator in a compact subset of $\mathbb{R}^d$.

- The key idea is to approximate the covariance kernel with

$$k(\mathbf{u}, \mathbf{u}' \mid \alpha, l) \approx \sum_{j=1}^{m} c_j \phi_j(\mathbf{u}) \phi_j(\mathbf{u}'), \quad \text{for some } c_j \in \mathbb{R}, \text{ and basis functions } \phi_j$$

  so that the $n \times n$ covariance matrix can be approximated as

$$\Sigma \approx \Phi(\mathbf{U}) \Lambda(K) \Phi(\mathbf{U})^T, \quad \text{where } \Phi(\mathbf{U}) \in \mathbb{R}^{n \times m}, \quad \Lambda(K) \in \mathbb{R}^{m \times m} = \mathrm{diag}(c_1, \ldots, c_m),$$

  where $\mathbf{U}$ denotes the collection of n locations, and $m \ll n$ is the number of basis functions.

- Now $\boldsymbol{\theta} \overset{d}{\approx} \Phi(\mathbf{U}) \Lambda(K)^{1/2} \mathbf{z}$, where $\mathbf{z} \sim N(0, \mathbf{I})$.
- The basis matrix $\Phi(\mathbf{U})$ only depends on the locations and can be pre-calculated in $\mathcal{O}(m^2 n)$. Computation cost in each MCMC iteration is reduced to $\mathcal{O}(nm + m)$.

Theory behind HSGP

Definition

Let $\mathcal{U} \subset \mathbb{R}^d$, and let $\mathbf{r} = \mathbf{u} - \mathbf{u}'$ for $\mathbf{u}$, $\mathbf{u}' \in \mathcal{U}$. A covariance function $k(\mathbf{u}, \mathbf{u}')$ is **stationary** if $k(\mathbf{u}, \mathbf{u}') = k(\mathbf{r})$. It is **isotropic** if $k(\mathbf{u}, \mathbf{u}') = k(\|\mathbf{r}\|)$.

By the Bochner's theorem, a bounded stationary covariance function

$$k(\mathbf{r}) = \frac{1}{(2\pi)^d} \int \exp(i\mathbf{w}^T\mathbf{r})\mu(d\mathbf{w})$$

for some positive measure μ. If μ has a density $S(\mathbf{w})$ w.r.t. the Lebesgue measure, it is called the spectral density. By the Wiener-Khintchin theorem, $k(\mathbf{r})$ and $S(\mathbf{w})$ are Fourier duals, i.e.,

$$k(\mathbf{r}) = \frac{1}{(2\pi)^d} \int \exp(i\mathbf{w}^T\mathbf{r})S(\mathbf{w})d\mathbf{w}$$

$$S(\mathbf{w}) = \int k(\mathbf{r}) \exp(i\mathbf{w}^T\mathbf{r})d\mathbf{r}.$$

Theory behind HSGP

For isotropic covariance functions where $k(\mathbf{u}, \mathbf{u}') = k(\|\mathbf{r}\|)$, the spectral density $S(\mathbf{w}) = S(\|\mathbf{w}\|)$. For regular covariance functions, S admits a closed form polynomial expansion of $\|\mathbf{w}\|^2$, i.e.,

$$S(\|\mathbf{w}\|) = \psi(\|\mathbf{w}\|^2) = a_0 + a_1(-\|\mathbf{w}\|^2) + a_2(-\|\mathbf{w}\|^2)^2 + \ldots \tag{1}$$

Note that $-\|\mathbf{w}\|^2$ is the transfer function for the Laplace operator ∇^2, i.e.,

$$[\mathcal{F}(\nabla^2 f)](\mathbf{w}) = -\|\mathbf{w}\|^2[\mathcal{F}(f)](\mathbf{w}),$$

where $\mathcal{F}$ denotes the Fourier transform operator. Therefore with an inverse Fourier transform on both sides of equation (1), we have

$$\mathcal{K} = a_0 + a_1(-\nabla^2) + a_2(-\nabla^2)^2 + \ldots, \tag{2}$$

where $\mathcal{K}$ is the covariance operator such that

$$\mathcal{K}f(\mathbf{u}) = \int k(\mathbf{u}, \mathbf{u}')f(\mathbf{u}')d\mathbf{u}'.$$

Theory behind HSGP

Continuing from equation (2)

$$\mathcal{K} = a_0 + a_1(-\nabla^2) + a_2(-\nabla^2)^2 + \dots.$$

This shows that if we are able to approximate the Laplace operator, we can approximate $\mathcal{K}$.

Given a compact set $\Omega \subset \mathbb{R}^d$, and sufficiently smooth boundary $\partial\Omega$, solution exists for the eigenvalue problem for the Laplace operator with Dirichlet boundary conditions

$$\begin{gathered} -\nabla^2 \phi_j(\mathbf{u}) = \lambda_j \phi_j(\mathbf{u}), \quad \mathbf{u} \in \Omega \\ \phi_j(\mathbf{u}) = 0, \quad \mathbf{u} \in \partial\Omega. \end{gathered}$$

Because $-\nabla^2$ is positive definite Hermitian,

- eigenvalues $\lambda_j \in \mathbb{R}+$ for all j
- basis function ϕ_j's are orthonormal w.r.t. inner product $< f, g >= \int_\Omega f(\mathbf{u})g(\mathbf{u})d\mathbf{u}$, i.e.,

$$\int_\Omega \phi_i(\mathbf{u})\phi_j(\mathbf{u})d\mathbf{u} = \delta_{ij}.$$

Theory behind HSGP

Therefore for sufficiently smooth functions f, we have

$$-\nabla^2 f(\mathbf{u}) = \int l(\mathbf{u},\mathbf{u}')f(\mathbf{u}')d\mathbf{u}', \quad \text{where the kernel } l(\mathbf{u},\mathbf{u}') = \sum_{j=1}^{\infty} \lambda_j \phi_j(\mathbf{u})\phi_j(\mathbf{u}').$$

Because of orthonormality, $l^s(\mathbf{u},\mathbf{u}') = \sum_{j=1}^{\infty} \lambda_j^s \phi_j(\mathbf{u})\phi_j(\mathbf{u}')$. Therefore putting everything together,

$$\begin{aligned}
\mathcal{K}f(\mathbf{u}) = \int k(\mathbf{u},\mathbf{u}')f(\mathbf{u}')d\mathbf{u}' &= [a_0 + a_1(-\nabla^2) + a_2(-\nabla^2)^2 + \ldots]f(\mathbf{u}) \\
&= \int [a_0 + a_1 l(\mathbf{u},\mathbf{u}') + a_2 l^2(\mathbf{u},\mathbf{u}') + \ldots]f(\mathbf{u}')d\mathbf{u}' \\
\implies k(\mathbf{u},\mathbf{u}') &\approx a_0 + a_1 l(\mathbf{u},\mathbf{u}') + a_2 l^2(\mathbf{u},\mathbf{u}') + \ldots \\
&= \sum_{j=1}^{\infty} [a_0 + a_1\lambda_j + a_2\lambda_j^2 + \ldots]\phi_j(\mathbf{u})\phi_j(\mathbf{u}').
\end{aligned}$$

Theory behind HSGP

Continuing from the derivations, we have

$$\begin{aligned} k(\mathbf{u}, \mathbf{u}') &\approx \sum_{j=1}^{\infty} [a_0 + a_1 \lambda_j + a_2 \lambda_j^2 + \dots] \phi_j(\mathbf{u}) \phi_j(\mathbf{u}') \\ &= \sum_{j=1}^{\infty} S(\sqrt{\lambda_j}) \phi_j(\mathbf{u}) \phi_j(\mathbf{u}') \qquad (3) \\ &\approx \sum_{j=1}^{m} S(\sqrt{\lambda_j}) \phi_j(\mathbf{u}) \phi_j(\mathbf{u}'). \end{aligned}$$

Remarks: Even with an inifinite sum, equation (3) is still just an approximation because the eigenvalues and basis functions are restricted to the domain Ω.

Posterior predictive under HSGP

Suppose we want to predict for the spatial intercepts at n_2 locations $\boldsymbol{\theta}_2 = (\boldsymbol{\theta}(\mathbf{d}'_1), \ldots, \boldsymbol{\theta}(\mathbf{d}'_{n_2}))$, based on n data points. Let $\Phi_1 \Lambda \Phi_1^T$ be the HSGP approximation for the covariance matrix of the observed data, and let Φ_2 denote the basis matrix for the prediction locations. Then approximately

$$\begin{pmatrix} \log \mathbf{y} - \mathbf{X}\beta \\ \boldsymbol{\theta}_2 \end{pmatrix} \sim N\left(0, \begin{pmatrix} \Phi_1 \Lambda \Phi_1^T + \sigma^2 \mathbf{I}_n & \Phi_1 \Lambda \Phi_2^T \\ \Phi_2 \Lambda \Phi_1^T & \Phi_2 \Lambda \Phi_2^T . \end{pmatrix}\right)$$

Therefore the conditionanl distribution of $\boldsymbol{\theta}_2$ is approximately:

$$\begin{aligned} &(\boldsymbol{\theta}_2 \mid \beta, \Lambda) \sim N(\mathbf{m}, \mathbf{S}) \\ &\mathbf{m} = \Phi_2 \Lambda \Phi_1^T (\Phi_1 \Lambda \Phi_1^T + \sigma^2 \mathbf{I}_n)^{-1} \mathbf{y}, \quad \mathbf{S} = \Phi_2 \Lambda \Phi_2^T - \Phi_2 \Lambda \Phi_1^T (\Phi_1 \Lambda \Phi_1^T + \sigma^2 \mathbf{I}_n)^{-1} \Phi_1 \Lambda \Phi_2^T \end{aligned}$$

where by the Woodbury formula,

$$(\Phi_1 \Lambda \Phi_1^T + \sigma^2 \mathbf{I}_n)^{-1} = \frac{1}{\sigma^2} \mathbf{I}_n - \Phi_1 (\sigma^2 \Lambda^{-1} + \Phi_1^T \Phi_1)^{-1} \Phi_1^T / \sigma^2$$

only requires inversion of an $m \times m$ matrix.

Summary of HSGP

To summarize, HSGP approximates the GP covariance function via an eigenfunction expansion of the Laplace operator in a compact set Ω.

- $k(\mathbf{u}, \mathbf{u}') \approx \sum_{j=1}^{m} S(\sqrt{\lambda_j})\phi_j(\mathbf{u})\phi_j(\mathbf{u}')$

Remarks:

- user needs to choose Ω. The solution to the eigenvalue problem (λ_j's and ϕ_j's) are independent of the choice of covariance function.
- user needs to choose m_i, number of basis functions for each dimension of the input domain $i \in \{1, \dots, d\}$. The approximation can be made arbitrarily accurate as m and Ω increase (see Theorem 1 and 4 of [Solin and Särkkä, 2020]).
- works best for $d \leq 3$, at most 4.
- works for covariance functions which admits a power spectral density, e.g., the Matérn family.

How to implement HSGP

To implement HSGP, user needs to decide on:

1. the domain Ω, e.g., rectangles $\prod_{i=1}^{d}[-L_i, L_i]$
2. number of basis function for each dimension m_i, $i \in \{1, \ldots, d\}$

To have **accuracy** and **speed**, we want to

1. choose minimal L_i's that are:
   - able to cover all the locations of interest, i.e., set $L_i = c_i S_i$, where S_i is the minimum half-length of dimension i that covers all locations, and $c_i \geq 1$
   - large enough so that errors at the boundary wouldn't affect the overall accuracy
2. choose m_i's that are:
   - large enough to ensure accuracy
   - small to minimize computation costs

Relationship between boundary factor c and number of basis functions m

[Riutort-Mayol et al., 2023] used extensive simulations to investigate the relationship between the boundary factor and number of basis functions.

Suppose $d = 1$, and the covariance kernel is known, we first fix c at a large enough value, and see how changes in m affect approximation accuracy.

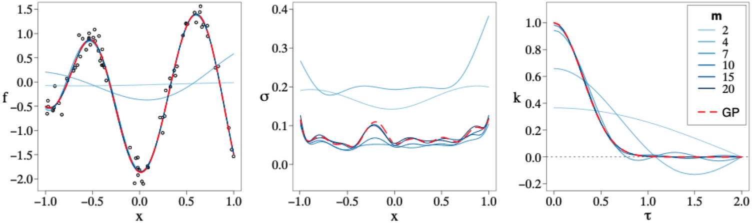


Figure 1: Mean posterior predictive functions (left), posterior standard deviations (center), and covariance functions (right) of both the exact GP model (dashed red line) and the HSGP model for different number of basis functions m, with the boundary factor fixed to a large enough value.

Relationship between boundary factor c and number of basis functions m

Still assuming the covariance kernel is known, we fix m at a large enough value, and see how changes in c affect approximation accuracy .

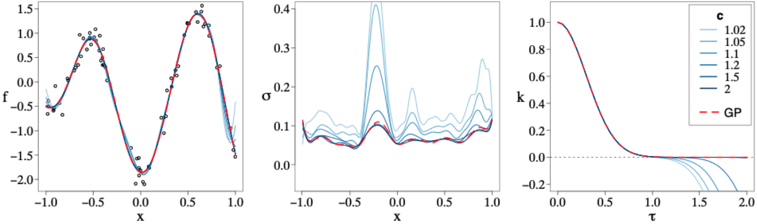


Figure 2: Mean posterior predictive functions (left), posterior standard deviations (center), and covariance functions (right) of both the exact GP model (dashed red line) and the HSGP model for different values of the boundary factor c, with a large enough fixed number of basis functions.

Relationship between boundary factor c and number of basis functions m

Finally, suppose the kernel parameters α and l are unknown and modeled, we look at results over combinations of different values of c's and m's.

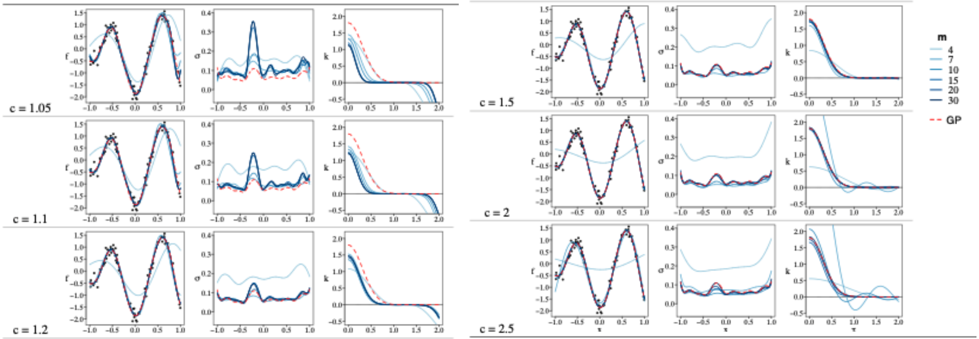


Figure: Posterior mean predictive functions (left), posterior standard deviations (center) and covariance functions (right) of both the exact GP model and the HSGP model for different m and for different c.

Near linear proportionality bewteen m, c and l

Given c, l, and the covariance kernel, [Riutort-Mayol et al., 2023] computed a priori the number of basis functions m needed to explain almost 100% of the variation.

For the Matérn family, when c is larger than the minimal recommended value, the minimum m required is near linear with l and c, i.e.,

- as l decreases, i.e., the process is less smooth, m approximately increases at $1/l$ rate
- as c increases, i.e., larger area, m approximately increases with c

[Riutort-Mayol et al., 2023] further empirically derived the numerical form for the Matérn family. E.g., for Matérn 3/2,

$$m = 3.42 \frac{c}{l/S} \Leftrightarrow \frac{l}{S} = 3.42 \frac{c}{m}, \quad c \geq 4.5 \frac{l}{S}, \quad c \geq 1.2.$$

How to implement HSGP

In applications, we typically do not know l. Therefore implementation of HSGP requires an iterative process. Again assuming $d = 1$,

0. Decide on a covariance kernel (e.g., Matérn 3/2) based on prior knowledge of the data, and set S based on the data
1. Set initial guess of l to say, 0.5, $c = \min(4.5l/S, 1.2)$, $m = 3.42l/S$
2. Run HSGP, get the posterior mean of lengthscale $\hat{l}$,
   - perform a diagnostic check on whether $\hat{l} + 0.01 > l$
   - if passed, increase m by 2, set $c = \min(4.5\hat{l}/S, 1.2)$, $l = 3.42Sc/m$
   - otherwise, set $l = \hat{l}$, $c = \min(4.5l/S, 1.2)$, and $m = 3.42l/S$
3. Repeat step 2 until the diagnostic check is passed for the last two iterations

Code Demos

[Riutort-Mayol et al., 2023] provides step-by-step tutorials, and implementation codes in stan. These are available on GitHub.

Simulation setup

We compared the performance of HSGP vs GP using simulated data under the Bayesian model discussed earlier:

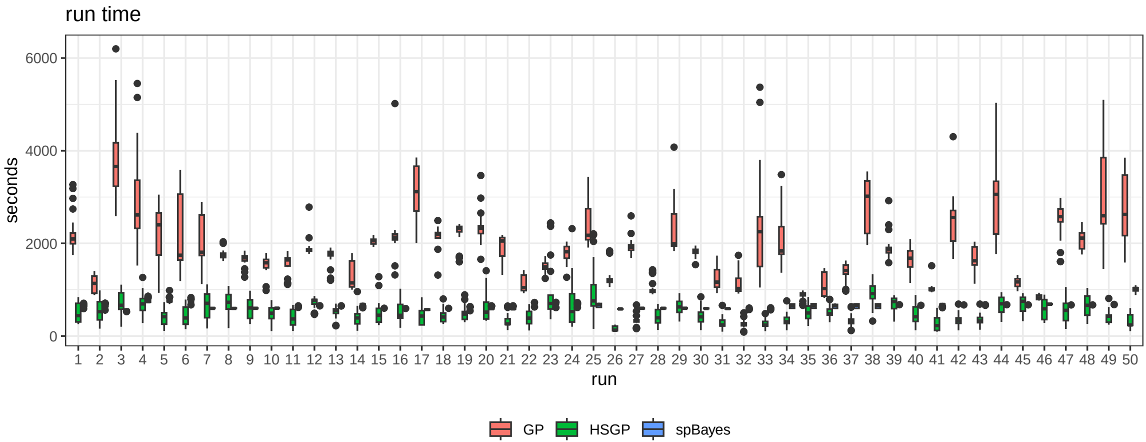
$$\log y_i = \mathbf{x}_i^T \boldsymbol{\beta} + \boldsymbol{\theta}(\mathbf{u}_i) + \epsilon_i,$$

- $\boldsymbol{\beta} \sim N(0, 25\mathbf{I})$, $\boldsymbol{\epsilon} \sim \ N(0, \sigma^2\mathbf{I})$, $\boldsymbol{\theta}(\mathbf{u}) \sim GP(0, K_{3/2}(\mathbf{u}, \mathbf{u}' \mid \alpha, l))$
- $\sigma \sim TN(0, 2)$, $\alpha \sim TN(0, 4)$, $l \sim IG(2, 1)$, where TN denotes truncated normal

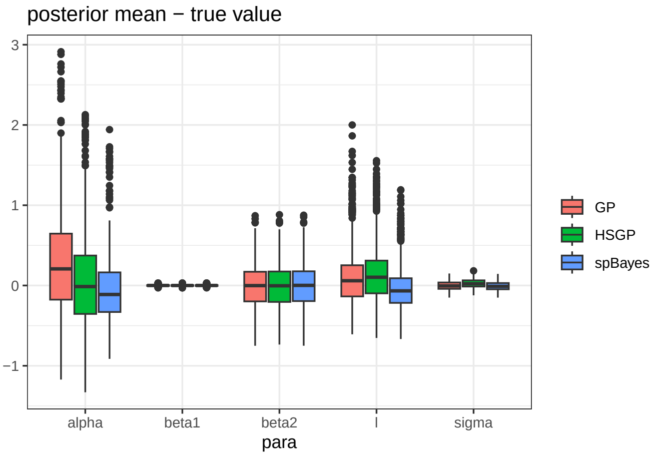
Simulation setup

- we used 50 batches, each with a different realization of $\boldsymbol{\theta}(\mathbf{u}_i)$'s, and different design matrix $\mathbf{X}$. For each batch, we simulated 25 datasets with $n = 200$
- we used $p = 2$, $\boldsymbol{\beta} = (-0.5, 1.2)^T$, $\alpha = 2$, $l = 1$, $\sigma = 1$

Simulation results

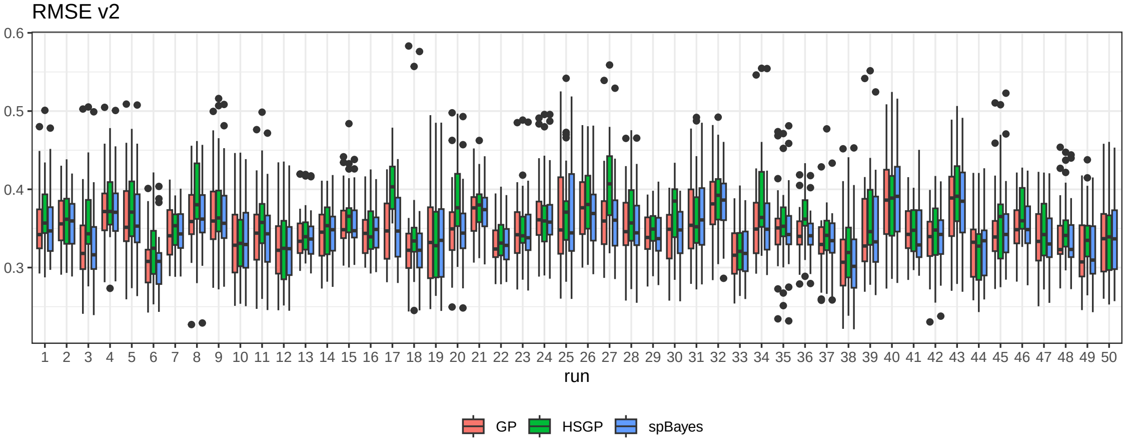


Simulation results



Simulation results

Out-of-sample prediction results by batch.



Conclusion and recommendations

[Solin and Särkkä, 2020] introduced a Hilbert space method for reduced-rank Gaussian process regression. [Riutort-Mayol et al., 2023] further provided detailed analysis of the performance, and practical implementation guides.

I would recommend reading and experimenting HSGP

- quick and accurate approximation of GP under covariance kernels with power spectral densities
- codes for Stan are readily available on GitHub
- useful when $d \leq 3$ and n is large

Potential Limitations

- not efficient when $d \geq 4$, or when the number of basis functions needed is large
- not sure how to make predictions if my GP setup doesn't have a nugget

References I

Riutort-Mayol, G., Bürkner, P.-C., Andersen, M. R., Solin, A., and Vehtari, A. (2023).
Practical hilbert space approximate bayesian gaussian processes for probabilistic programming.
Statistics and Computing, 33(1):17.

Solin, A. and Särkkä, S. (2020).
Hilbert space methods for reduced-rank gaussian process regression.
Statistics and Computing, 30(2):419–446.