`lineqGPR`: an R package for Gaussian process modelling with inequality constraints

Andrés F. López-Lopera[1,2,3], François Bachoc[2], Nicolas Durrande[1,4], and Olivier Roustant[1,2]

[1]Mines Saint-Etienne, France.
[2]Toulouse Mathematics Institute (IMT), France.
[3]French Geological Survey BRGM, Orléans, France.
[4]PROWLER.io, Cambridge, UK.

This work was funded by the chair in applied mathematics OQUAIDO

OQUAIDO scientific meeting

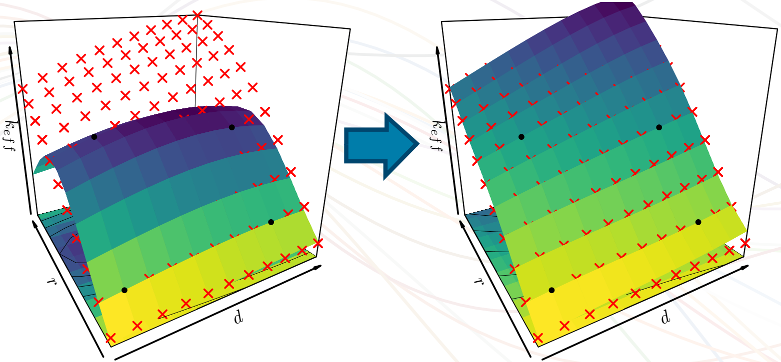
November 26, 2019

Table of Contents

Table of Contents

2D models for interpolating the IRSN's dataset (López-Lopera et al., 2018).

k_{eff}: effective neutron multiplication factor



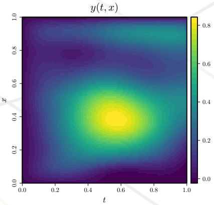
(a) unconstrained model (b) model with positivity & monotonicity

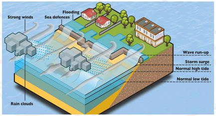
• interpolation points ($n = 4$) × test data

Other applications

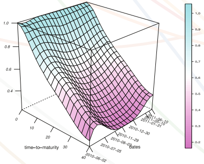
- Regulation of gene expressions – positivity constraints (Lawrence et al., 2007).
- Coastal flooding – positivity & monotonicity (Rohmer and Idier, 2012).
- Econometrics – positivity or monotonicity (Cousin et al., 2016).
- Nuclear physics – monotonicity & convexity (Zhou et al., 2019).

$\vdots$



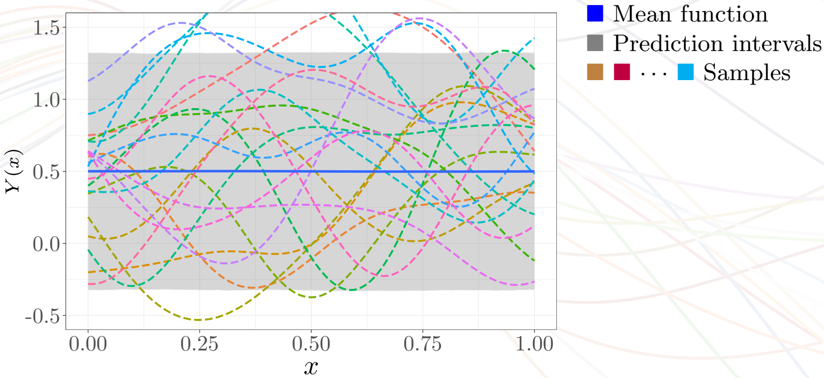


https://nerc.ukri.org/



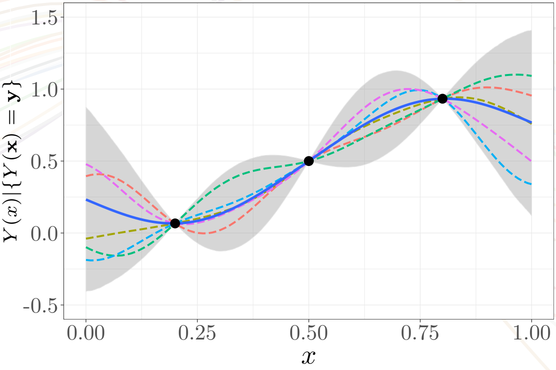
Cousin et al. (2016)

GPs form a flexible **prior over functions** (Rasmussen and Williams, 2005):



$$Y \sim \mathcal{GP}\left(\underbrace{\mu(x)}_{\text{mean}} = 0.5,\ \underbrace{k_\theta(x,x')}_{\text{covariance}} = \sigma^2 \exp\left\{-\frac{(x-x')^2}{2\ell^2}\right\}\right)$$

GPs form a flexible **prior over functions** (Rasmussen and Williams, 2005):

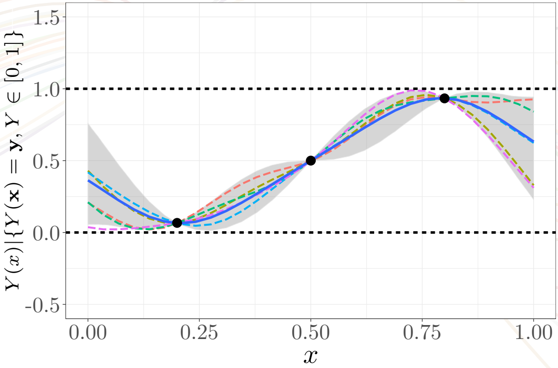


- Mean function
- Prediction intervals
- ··· Samples
- Interpolation conditions

$$\mathcal{D} = (x_i, y_i)_{i=1}^{n} = (\boldsymbol{x}, \boldsymbol{y})$$

$$Y|\{Y(\mathbf{x}) = \mathbf{y}\} \sim \mathcal{GP}(\underbrace{m(x)}_{\text{cond. mean}}, \underbrace{c_\theta(x, x')}_{\text{cond. covariance}})$$

GPs form a flexible **prior over functions** (Rasmussen and Williams, 2005):



Mean function

Prediction intervals

$\cdots$ Samples

● Interpolation conditions

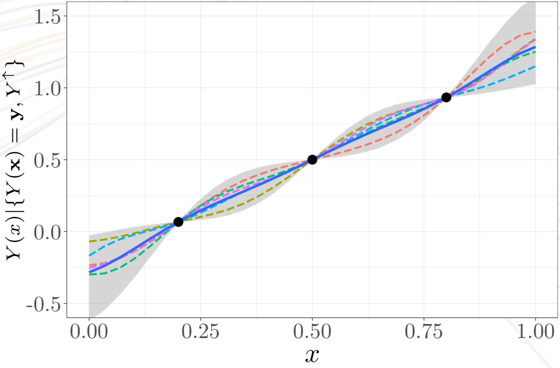
$$\mathcal{D} = (x_i, y_i)_{i=1}^{n} = (\boldsymbol{x}, \boldsymbol{y})$$

- - Boundedness condition

$$Y \in [0, 1]$$

$$Y|\{Y(\mathbf{x}) = \mathbf{y}, Y \in [0,1]\} \sim \textbf{?}$$

GPs form a flexible **prior over functions** (Rasmussen and Williams, 2005):



$Y|\{Y(\mathbf{x}) = \mathbf{y}, Y^{\uparrow}\} \sim$?

- Mean function
- Prediction intervals
- ⋯ Samples

● Interpolation conditions

$$\mathcal{D} = (x_i, y_i)_{i=1}^{n} = (\boldsymbol{x}, \boldsymbol{y})$$

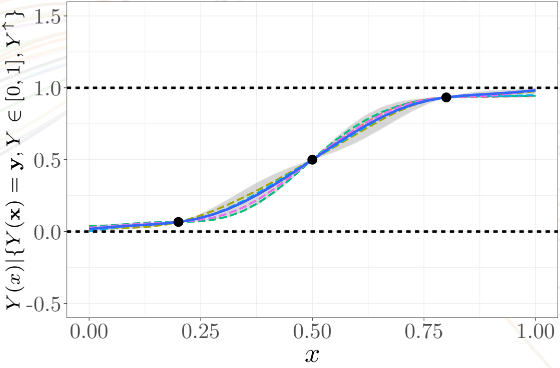
- - Boundedness condition

$$Y \in [0, 1]$$

↑ Monotonicity condition

$$Y(x) \leq Y(x'), \quad \forall\, x \leq x'.$$

GPs form a flexible **prior over functions** (Rasmussen and Williams, 2005):



$Y | \{Y(\mathbf{x}) = \mathbf{y}, Y \in [0,1], Y^{\uparrow}\} \sim$ **?**

- Mean function
- Prediction intervals
- ⋯ Samples

● Interpolation conditions

$$\mathcal{D} = (x_i, y_i)_{i=1}^{n} = (\boldsymbol{x}, \boldsymbol{y})$$

- - Boundedness condition

$$Y \in [0, 1]$$

↑ Monotonicity condition

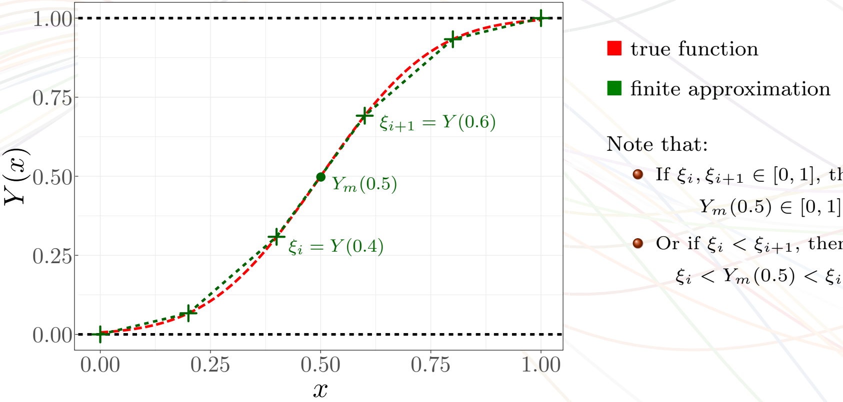
$$Y(x) \le Y(x'), \quad \forall\, x \le x'.$$

The main contributions in `lineqGPR` are threefold:

1. The improvement of the **applicability of GPs under constraints**.
2. The **scalability** of constrained GPs **to high dimensions**:
   - i.e. involving hundreds of input variables under additivity conditions.
3. **Parameter estimation under inequality constraints**.

1D finite-dimensional Gaussian approximation

Finite-dimensional representation Y_m: also bounded & monotonic.



Note that:

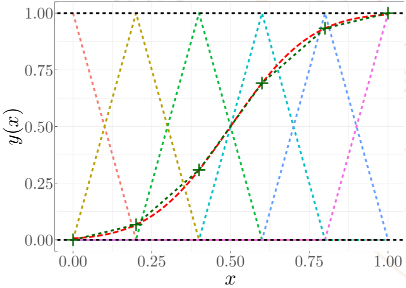
- If $\xi_i, \xi_{i+1} \in [0, 1]$, then $Y_m(0.5) \in [0, 1]$.
- Or if $\xi_i < \xi_{i+1}$, then $\xi_i < Y_m(0.5) < \xi_{i+1}$.

Imposing constraints on the knots is enough (Maatouk and Bay, 2017).

- Let the (constrained) finite-dimensional GP Y_m be defined as

$$Y_m(x) = \sum_{j=1}^{m} \xi_j \phi_j(x), \text{ s.t. } \begin{cases} Y_m(x_i) = y_i & \text{(interpolation conditions)}, \\ \boldsymbol{l} \leq \boldsymbol{\Lambda\xi} \leq \boldsymbol{u} & \text{(linear inequality conditions)}, \end{cases} \quad (1)$$

where $x_i \in [0, 1]$, $y_i \in \mathbb{R}$ for $i = 1, \ldots, n$; and
  - $\boldsymbol{\xi} = [\xi_1, \ldots, \xi_m]^\top \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Gamma}_\theta)$ with covariance matrix $\boldsymbol{\Gamma}_\theta$,
  - $(\boldsymbol{\Lambda}, \boldsymbol{l}, \boldsymbol{u})$ defines the inequality conditions, and
  - $\phi_j : [0, 1] \mapsto \mathbb{R}$ are hat basis functions:



- Since the **Gaussianity** is preserved for *linear operations*:

$$\boldsymbol{\Lambda}\boldsymbol{\xi}|\{\boldsymbol{\Phi}\boldsymbol{\xi} = \boldsymbol{y}\} \sim \mathcal{N}\left(\boldsymbol{\Lambda}\boldsymbol{\mu}, \boldsymbol{\Lambda}\boldsymbol{\Sigma}\boldsymbol{\Lambda}^\top\right), \quad \text{(conditional distribution)} \tag{2}$$

with conditional parameters $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ given by

$$\mathbf{K} = \boldsymbol{\Phi}\boldsymbol{\Gamma}\boldsymbol{\Phi}^\top, \quad \boldsymbol{\mu} = \boldsymbol{\Gamma}\boldsymbol{\Phi}^\top\mathbf{K}^{-1}\boldsymbol{y}, \quad \boldsymbol{\Sigma} = \boldsymbol{\Gamma} - \boldsymbol{\Gamma}\boldsymbol{\Phi}^\top\mathbf{K}^{-1}\boldsymbol{\Phi}\boldsymbol{\Gamma}.$$

- Since the **Gaussianity** is preserved for *linear operations*:

$$\boldsymbol{\Lambda\xi} | \{\boldsymbol{\Phi\xi} = \boldsymbol{y}\} \sim \mathcal{N}\left(\boldsymbol{\Lambda\mu}, \boldsymbol{\Lambda\Sigma\Lambda}^\top\right), \quad \text{(conditional distribution)} \tag{2}$$

with conditional parameters $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ given by

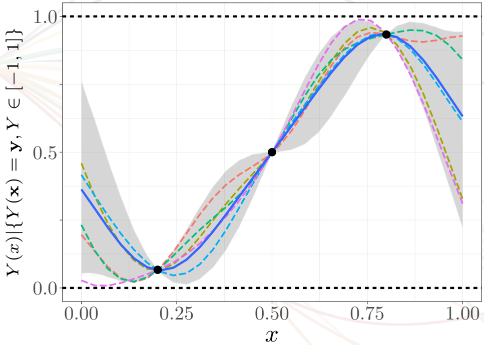
$$\mathbf{K} = \boldsymbol{\Phi\Gamma\Phi}^\top, \quad \boldsymbol{\mu} = \boldsymbol{\Gamma\Phi}^\top\mathbf{K}^{-1}\boldsymbol{y}, \quad \boldsymbol{\Sigma} = \boldsymbol{\Gamma} - \boldsymbol{\Gamma\Phi}^\top\mathbf{K}^{-1}\boldsymbol{\Phi\Gamma}.$$

- Then, *quantifying uncertainty* on $Y_m(x) = \sum_{j=1}^m \xi_j \phi_j(x)$ relies on simulating the **truncated Gaussian vector** $\boldsymbol{\xi}$:

$$\boldsymbol{\Lambda\xi} | \{\boldsymbol{\Phi\xi} = \boldsymbol{y}, \boldsymbol{l} \leq \boldsymbol{\Lambda\xi} \leq \boldsymbol{u}\} \sim \mathcal{TN}(\boldsymbol{\Lambda\mu}, \boldsymbol{\Lambda\Sigma\Lambda}^\top, \boldsymbol{l}, \boldsymbol{u}). \tag{3}$$

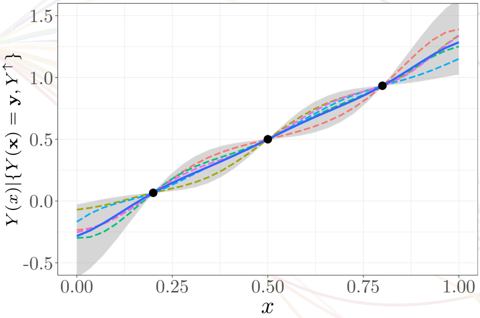
(3) is computed via *Monte Carlo* (MC) or *Markov Chain MC* (MCMC):
- **e.g. *Hamiltonian MC* (HMC) (Pakman and Paninski, 2014).**

1D example with **boundedness** constraints via HMC



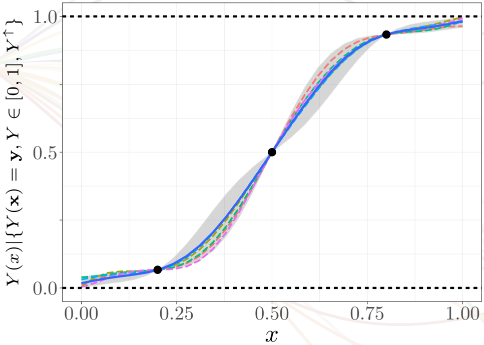
$$\underbrace{\begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix}}_{\boldsymbol{l}} \leq \underbrace{\begin{bmatrix} 1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 1 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 1 & 0 \\ 0 & 0 & 0 & \cdots & 0 & 1 \end{bmatrix}}_{\boldsymbol{\Lambda}} \underbrace{\begin{bmatrix} \xi_1 \\ \xi_2 \\ \xi_3 \\ \vdots \\ \xi_{m-1} \\ \xi_m \end{bmatrix}}_{\boldsymbol{\xi}} \leq \underbrace{\begin{bmatrix} 1 \\ 1 \\ 1 \\ \vdots \\ 1 \\ 1 \end{bmatrix}}_{\boldsymbol{u}}$$

1D example with **monotonicity** constraints via HMC



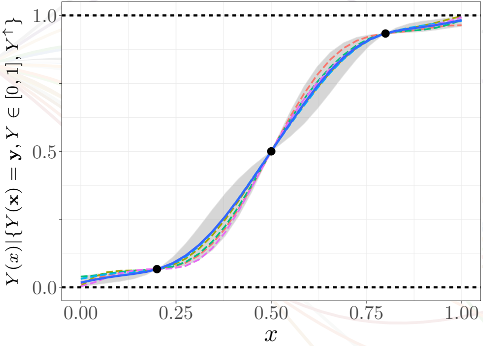
$$\underbrace{\begin{bmatrix}0\\0\\0\\\vdots\\0\\0\end{bmatrix}}_{\boldsymbol{l}} \leq \underbrace{\begin{bmatrix} -1 & 1 & 0 & \cdots & 0 & 0\\ 0 & -1 & 1 & \cdots & 0 & 0\\ 0 & 0 & -1 & \cdots & 0 & 0\\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots\\ 0 & 0 & 0 & \cdots & 1 & 0\\ 0 & 0 & 0 & \cdots & -1 & 1\end{bmatrix}}_{\boldsymbol{\Lambda}} \underbrace{\begin{bmatrix}\xi_1\\ \xi_2\\ \xi_3\\ \vdots\\ \xi_{m-1}\\ \xi_m\end{bmatrix}}_{\boldsymbol{\xi}} \leq \underbrace{\begin{bmatrix}\infty\\ \infty\\ \vdots\\ \infty\\ \infty\end{bmatrix}}_{\boldsymbol{u}}$$

1D example with **boundedness** & **monotonicity** constraints via HMC



$$\underbrace{\begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix}}_{\boldsymbol{l}} \leq \underbrace{\begin{bmatrix} 1 & 0 & 0 & \cdots & 0 & 0 \\ -1 & 1 & 0 & \cdots & 0 & 0 \\ 0 & -1 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & -1 & 1 \\ 0 & 0 & 0 & \cdots & 0 & 1 \end{bmatrix}}_{\boldsymbol{\Lambda}} \underbrace{\begin{bmatrix} \xi_1 \\ \xi_2 \\ \xi_3 \\ \vdots \\ \xi_{m-1} \\ \xi_m \end{bmatrix}}_{\boldsymbol{\xi}} \leq \underbrace{\begin{bmatrix} 1 \\ \infty \\ \infty \\ \vdots \\ \infty \\ 1 \end{bmatrix}}_{\boldsymbol{u}}$$

1D example with **boundedness** & **monotonicity** constraints via HMC



or simply,

$$\underbrace{\begin{bmatrix} \boldsymbol{l}_b \\ \boldsymbol{l}_m \end{bmatrix}}_{\boldsymbol{l}} \leq \underbrace{\begin{bmatrix} \boldsymbol{\Lambda}_b \\ \boldsymbol{\Lambda}_m \end{bmatrix}}_{\boldsymbol{\Lambda}} \boldsymbol{\xi} \leq \underbrace{\begin{bmatrix} \boldsymbol{u}_b \\ \boldsymbol{u}_m \end{bmatrix}}_{\boldsymbol{u}}.$$

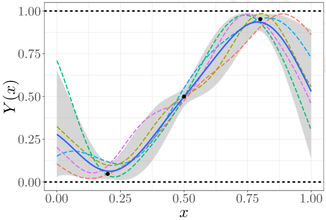
Considering noisy observations

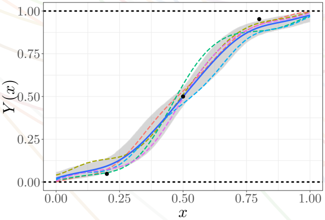
$$Y_m(x) = \sum_{j=1}^{m} \xi_j \phi_j(x), \text{ s.t. } \begin{cases} Y_m(x_i) + \varepsilon_i = y_i & \text{(interpolation conditions)}, \\ \boldsymbol{l} \leq \boldsymbol{\Lambda\xi} \leq \boldsymbol{u} & \text{(linear inequality conditions)}, \end{cases} \quad (4)$$

where $\varepsilon_i \sim \mathcal{N}(0, \tau^2)$ with noise variance τ^2.

Modification: $\mathbf{K} = \boldsymbol{\Phi\Gamma\Phi}^\top \quad \rightarrow \quad \mathbf{K} = \boldsymbol{\Phi\Gamma\Phi}^\top + \tau^2\mathbf{I}$

boundedness | **boundedness** & **monotonicity**





Adding ε leads to more flexible GP models and less restrictive sample spaces for MC/MCMC algorithms (López-Lopera et al., 2019).

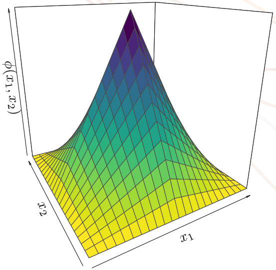
Finite-dimensional approximation in d dimensions

- The extension to d dimensions is obtained by **tensorisation**:

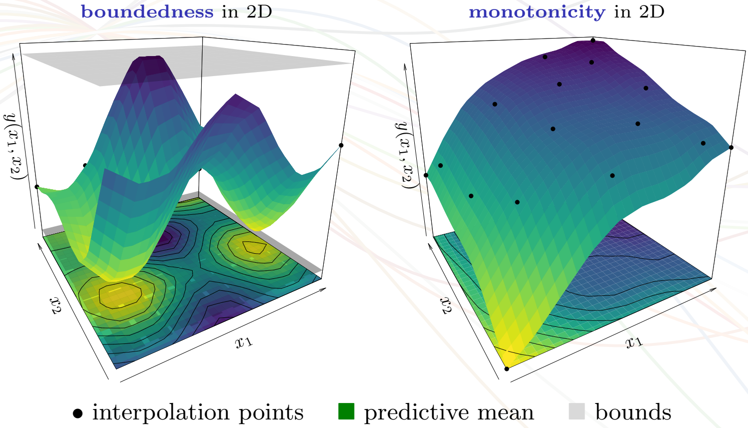
$$Y_m(\boldsymbol{x}) = \sum_{j_1,\ldots,j_d=1}^{m_1,\ldots,m_d} \left[\prod_{p=1,\ldots,d} \phi_{j_p}^{(p)}(x_p) \right] \xi_{j_1,\cdots,j_d}, \text{ s.t. } \begin{cases} Y_m(\boldsymbol{x}_i) + \varepsilon_i = y_i, \\ \boldsymbol{\xi} \in \mathcal{C}, \end{cases} \quad (5)$$

where $\boldsymbol{x}_i \in [0,1]^d$, $y_i \in \mathbb{R}$, $\varepsilon_i \sim \mathcal{N}\left(0, \tau^2\right)$, for $i = 1, \ldots, n$; and

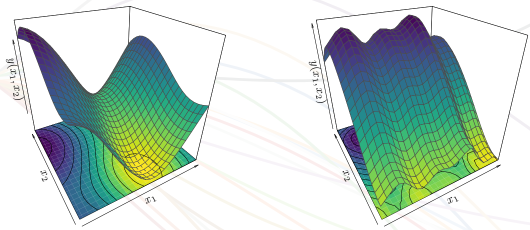
- $\boldsymbol{\xi} = [\xi_{1,\ldots,1}, \ldots, \xi_{m_1,\ldots,m_d}]^\top \sim \mathcal{N}(\mathbf{0}, \mathbf{\Gamma}_\theta)$,
- $\mathcal{C}$ is a convex set of linear inequality constraints, and
- $\phi_{j_i}^{(i)} : [0,1] \mapsto \mathbb{R}$ are hat basis functions.



2D numerical illustration



Additive GPs



2D examples of additive GPs.

- Let Y be the additive process on $\mathbb{R}^d$ given by

$$Y(\boldsymbol{x}) = \sum_{p=1}^{d} Y_p(x_p), \quad \text{with} \quad Y_p \sim \mathcal{GP}(0, k_p). \tag{6}$$

- Assume that $Y_1, \ldots, Y_d$ are independent. Then, $Y \sim \mathcal{GP}(0, k)$ with:

$$k(\boldsymbol{x}, \boldsymbol{x}') = \sum_{p=1}^{d} k_p(x_p, x_p{}').$$

- Assume that Y exhibits certain constraints along Y_p, then Y_m is

$$Y_p \to Y_{p,m_p}$$

$$Y_m(\boldsymbol{x}) = \sum_{p=1}^{d} Y_{p,m_p}(x_p), \quad \text{s.t.} \quad \begin{cases} Y_m(\boldsymbol{x}_i) + \varepsilon_i = y_i, \\ \boldsymbol{\xi}_\kappa \in \mathcal{C}_\kappa, \end{cases} \tag{7}$$

with $\boldsymbol{x}_i \in [0,1]^d$, $y_i \in \mathbb{R}$ and $\varepsilon_i \sim \mathcal{N}\left(0, \tau^2\right)$ for $i = 1, \ldots, n$.

- Assume that Y exhibits certain constraints along Y_p, then Y_m is

$$Y_p \to Y_{p,m_p}$$

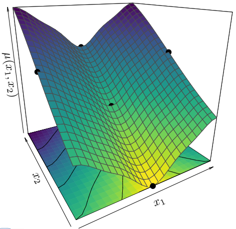
$$Y_m(\boldsymbol{x}) = \sum_{p=1}^{d} Y_{p,m_p}(x_p), \quad \text{s.t.} \quad \begin{cases} Y_m(\boldsymbol{x}_i) + \varepsilon_i = y_i, \\ \boldsymbol{\xi}_\kappa \in \mathcal{C}_\kappa, \end{cases} \tag{7}$$

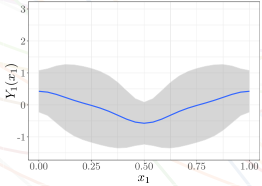
with $\boldsymbol{x}_i \in [0,1]^d$, $y_i \in \mathbb{R}$ and $\varepsilon_i \sim \mathcal{N}\left(0, \tau^2\right)$ for $i = 1, \ldots, n$.

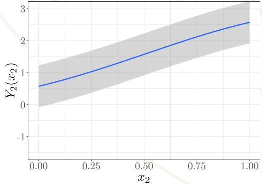
2D toy example:

$$Y(x_1, x_2) = \underbrace{4(x_1 - 0.5)^2}_{Y_1(x_1)} + \underbrace{2x_2}_{Y_2(x_2)}$$

Predictive mean without constraints







Additive GPs under inequality constraints

- Assume that Y exhibits certain constraints along Y_p, then Y_m is

$$Y_p \rightarrow Y_{p,m_p}$$

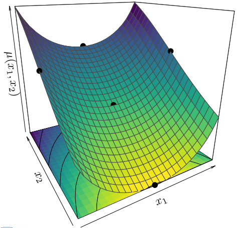
$$Y_m(\boldsymbol{x}) = \sum_{p=1}^{d} Y_{p,m_p}(x_p), \quad \text{s.t.} \quad \begin{cases} Y_m(\boldsymbol{x}_i) + \varepsilon_i = y_i, \\ \boldsymbol{\xi}_\kappa \in \mathcal{C}_\kappa, \end{cases} \tag{7}$$

with $\boldsymbol{x}_i \in [0,1]^d$, $y_i \in \mathbb{R}$ and $\varepsilon_i \sim \mathcal{N}\left(0, \tau^2\right)$ for $i = 1, \ldots, n$.

2D toy example:

$$Y(x_1, x_2) = \underbrace{4(x_1 - 0.5)^2}_{Y_1(x_1)} + \underbrace{2x_2}_{Y_2(x_2)}$$

Predictive mean with constraints



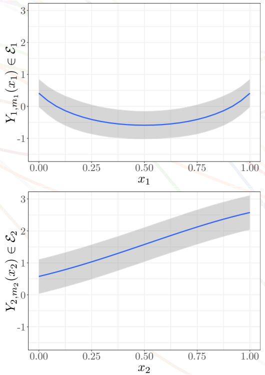


Table of Contents

- `lineqGPR` is an R package for GP modelling under inequality constraints.
- It is based on multiple contributions:
  - L-L, A.F., Bachoc, F., Durrande, N., and Roustant, O. (2018). Finite-dimensional Gaussian approximation with linear inequality constraints. In *SIAM/ASA Journal on Uncertainty Quantification*.
  - Bachoc, F., Lagnoux, A., and L-L, A.F. (2019). Maximum likelihood estimation for Gaussian processes under inequality constraints. In *Electronic Journal of Statistics*.
  - L-L, A.F., Bachoc, F., Durrande, N., Rohmer, J., Idier, D., and Roustant, O (2019). Approximating Gaussian process emulators with linear inequality constraints and noisy observations via MC and MCMC. In *MCQMC*.

The main functionalities of `lineqGPR` are implemented as S3 methods.

Method Name	Description
create	Creation of GP models under inequality constraints.
lineqGPOptim	Parameter estimation under inequality constraints.
predict	Prediction of the objective function at new points.
simulate	Simulation of GP models under inequality constraints.
plot, ggplot	Plot for GP models.

Example

```
## ----------------------------------------------------------------- ##
## Gaussian process regression modelling under boundedness constraints ##
## ----------------------------------------------------------------- ##
library(lineqGPR)
library(ggplot2)

#### generating synthetic data ####
sigfun <- function(x) return(1/(1+exp(-7*(x-0.5))))
x <- seq(0, 1, 0.001); y <- sigfun(x)
DoE <- splitDoE(x, y, DoE.idx = c(201, 501, 801))

#### GP with active boundedness constraints ####
# creating the "lineqGP" model
model <- create(class = "lineqGP",
                x = DoE$xdesign, y = DoE$ydesign,
                constrType = "boundedness")
model$localParam$m <- 100 # nb of knots
model$bounds <- c(0,1) # defining the bounds

# sampling from the model
sim.model <- simulate(model, nsim = 1e4, seed = 1, xtest = DoE$xtest)
plot(sim.model, bounds = model$bounds, xlab = "x", ylab = "y(x)")
lines(x, y, lty = 2)
```

Note: see more example in the Jupyter Notebooks.

Bachoc, F., Lagnoux, A., and López-Lopera, A. F. (2019). Maximum likelihood estimation for Gaussian processes under inequality constraints. *Electronic J. of Statistics*, 13(2):2921–2969.

Cousin, A., Maatouk, H., and Rullière, D. (2016). Kriging of financial term-structures. *European Journal of Operational Research*, 255(2):631–648.

Durrande, N., Ginsbourger, D., and Roustant, O. (2012). Additive covariance kernels for high-dimensional Gaussian process modeling. *Annales de la Faculté de Sciences de Toulouse*, 21(3):481–499.

Lawrence, N. D., Sanguinetti, G., and Rattray, M. (2007). Modelling transcriptional regulation using Gaussian processes. In *Neural Information Processing Systems*, pages 785–792.

López-Lopera, A. F. (2018). lineqGPR: Gaussian process regression models with linear inequality constraints. `https://cran.r-project.org/web/packages/lineqGPR/index.html`.

López-Lopera, A. F., Bachoc, F., Durrande, N., and Roustant, O. (2018). Finite-dimensional Gaussian approximation with linear inequality constraints. *SIAM/ASA Journal on Uncertainty Quantification*, 6(3):1224–1255.

López-Lopera, A. F., Bachoc, F., Durrande, N., and Roustant, O. (2019). Approximating Gaussian process emulators with linear inequality constraints and noisy observations via MC and MCMC. *To appear in Proceedings in Monte Carlo and Quasi-Monte Carlo Methods*.

López-Lopera, A. F., John, S., and Durrande, N. (2019). Gaussian process modulated Cox processes under linear inequality constraints. In *AISTATS*, pages 1997–2006.

Maatouk, H. and Bay, X. (2017). Gaussian process emulators for computer experiments with inequality constraints. *Mathematical Geosciences*, 49(5):557–582.

Pakman, A. and Paninski, L. (2014). Exact Hamiltonian Monte Carlo for truncated multivariate Gaussians. *Journal of Computational and Graphical Statistics*, 23(2):518–542.

Rasmussen, C. E. and Williams, C. K. I. (2005). *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. MIT Press.

Rohmer, J. and Idier, D. (2012). A meta-modelling strategy to identify the critical offshore conditions for coastal flooding. *Natural Hazards and Earth System Sciences*, 12(9):2943–2955.

Zhou, S., Giulani, P., Piekarewicz, J., Bhattacharya, A., and Pati, D. (2019). Reexamining the proton-radius problem using constrained Gaussian processes. *Physical Review C*, 99:055202.