MINES Saint-Étienne · Institut de Mathématiques de Toulouse

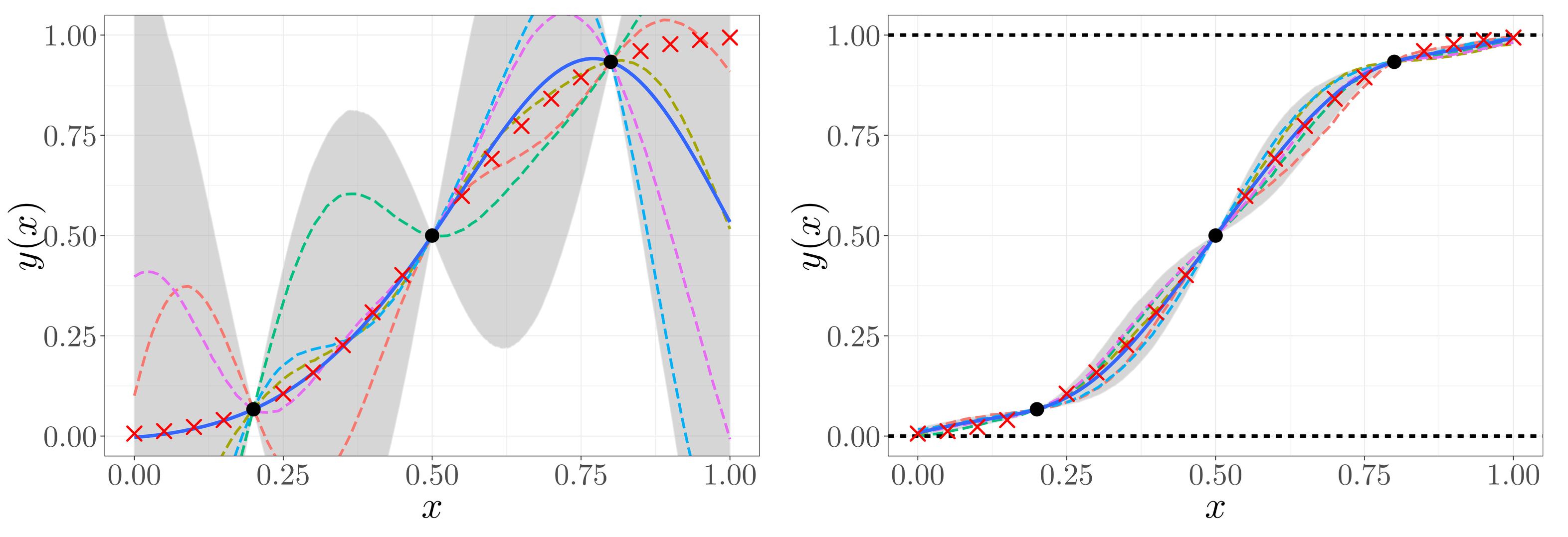
lineqGPR: An R Package for Gaussian Process Regression Modelling with Linear Inequality Constraints

A. F. López-Lopera[1], F. Bachoc[2], N. Durrande[1,3], and O. Roustant[1].
[1]Mines Saint-Étienne, France. [2]Institut de Mathmatiques de Toulouse, France. [3]PROWLER.io, UK.

UseR! International Conference

Gaussian Process (GP) Models under Inequality Constraints

- A GP is a collection of random variables, any finite number of which have a joint Gaussian distribution [1].
- Conditioning GPs by inequality constraints gives more realistic models [2, 3].



(a) Unconstrained GP model (b) With boundedness and monotonicity constraints

R Package lineqGPR [4]

- `lineqGPR` gathers GP implementations under inequality constraints.
- It is based on previous R packages such as `DiceKriging` [5] and `kergp` [6].

Main functionalities of `lineqGPR` are implemented as S3 methods.

Method Name	Description
create	Creation function of GP models under inequality constraints.
lineqGPOptim	Covariance 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 a constrained GP models.

Further GP implementations.

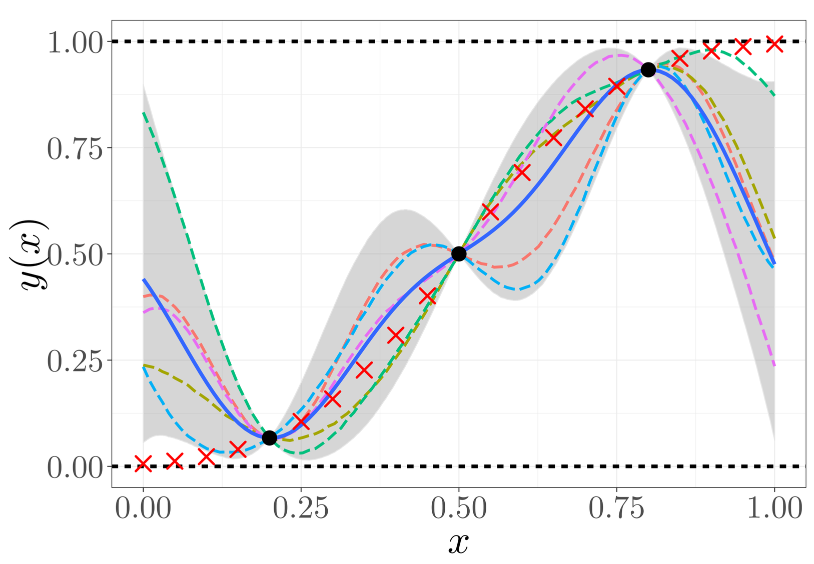
Class	Description
lineqDGP	Framework in [2] with **derivative information**.
lineqGP	**Derivative-free** framework in [3].
lineqAGP	**Additive GP models** under linear inequality constraints.

Demo under Boundedness Constraints

```
In [3]: #### Generating the synthetic dataset ####
    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 [0,1] ####
    model <- create(class = "lineqGP", x = DoE$xdesign, y = DoE$ydesign,
                    constrType = "boundedness")
    model$localParam$m <- 100 # changing the (default) number of knots
    model$bounds <- c(0,1) # changing the (default) bounds

    # sampling from the model
    sim.model <- simulate(model, nsim = 1e3, seed = 1, xtest = DoE$xtest)
    ggplotLineqGPModel <- ggplot(sim.model)
```



Acknowledgment

- This work was funded by the chair of applied mathematics OQUAIDO.

References

[1] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. Cambridge, MA: The MIT Press, 2005.

[2] H. Maatouk and X. Bay, "Gaussian process emulators for computer experiments with inequality constraints," *Mathematical Geosciences*, 2017.

[3] A. F. López-Lopera, F. Bachoc, N. Durrande, and O. Roustant, "Finite-dimensional Gaussian approximation with linear inequality constraints," *SIAM/ASA Journal on Uncertainty Quantification*, 2018.

[4] A. F. López-Lopera, *lineqGPR: Gaussian Process Regression Models with Linear Inequality Constraints*, 2019, R package version 0.0.4. [Online]. Available: https://CRAN.R-project.org/package=lineqGPR

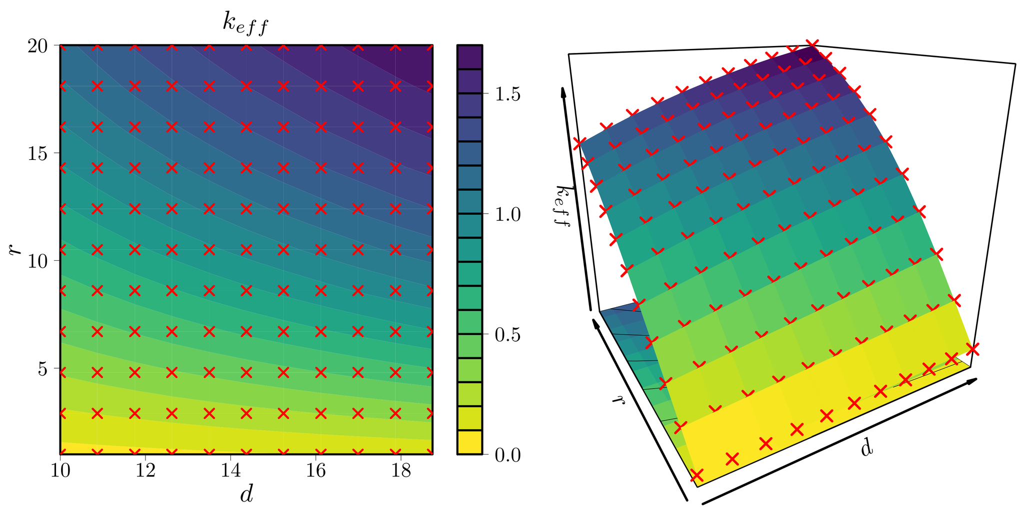
[5] O. Roustant, D. Ginsbourger, and Y. Deville, "DiceKriging, DiceOptim: Two R packages for the analysis of computer experiments by Kriging-based metamodeling and optimization," *Journal of Statistical Software*, 2012.

[6] Y. Deville, D. Ginsbourger, and O. Roustant, *kergp: Gaussian Process Laboratory*, 2015, R package version 0.2.0. [Online]. Available: https://CRAN.R-project.org/package=kergp

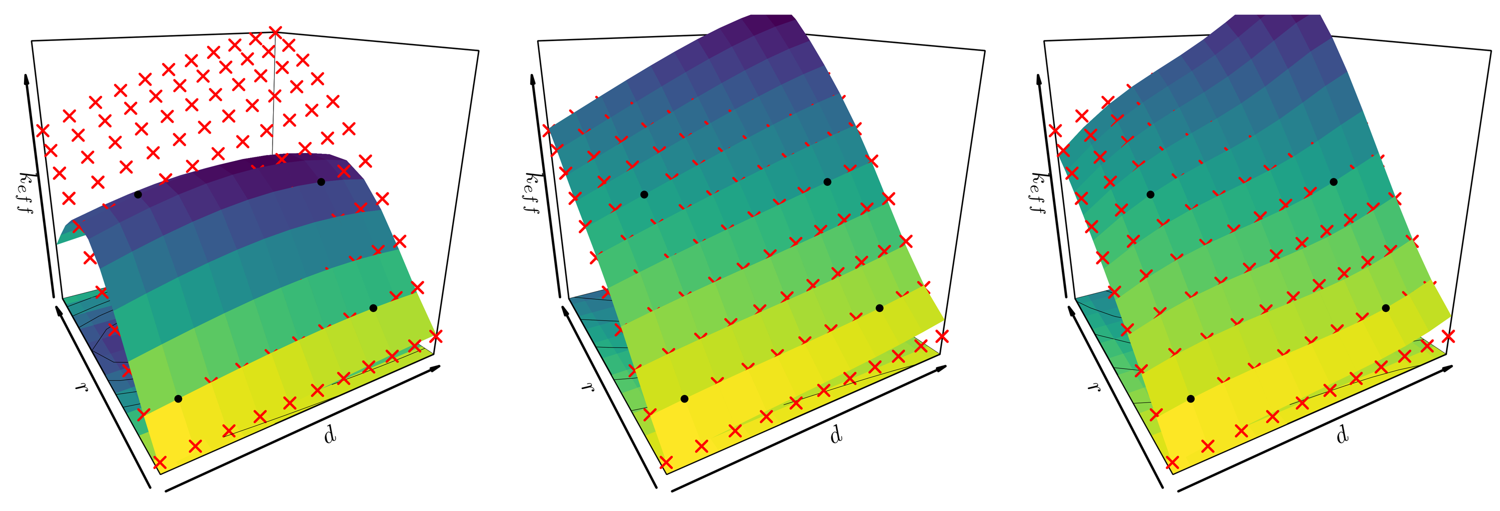
[7] F. Bachoc, A. Lagnoux, and A. F. López-Lopera, "Maximum likelihood estimation for Gaussian processes under inequality constraints," *Accepted for publication in Electronic Journal of Statistics*, 2019.

[8] A. Pakman and L. Paninski, "Exact Hamiltonian Monte Carlo for truncated multivariate Gaussians," *Journal of Computational and Graphical Statistics*, 2014.

2D Application: Nuclear Safety Criticality Assessment



Nuclear criticality safety dataset. k_{eff} is positive and non-decreasing.



(a) GP + ML (b) Constr. GP + ML (c) Constr. GP + cML

$n = 4$ training points. Maximum likelihood (ML): $\mathcal{L}_n(\boldsymbol{\theta}) = \log P_{\boldsymbol{\theta}}(\mathbf{Y}_n)$.

Constrained ML (cML): $\mathcal{L}_{n,c}(\boldsymbol{\theta}) = \log P_{\boldsymbol{\theta}}(\mathbf{Y}_n | Y \in \mathcal{E})$.

Performance of GPs for different n and using 20 random Latin hypercube designs. The accuracy is evaluated using the mean μ and the standard deviation σ of the Q^2 results.

n	GP + ML $\mu \pm \sigma$	Constr. GP + ML $\mu \pm \sigma$	Constr. GP + cML $\mu \pm \sigma$
4	0.558 ± 0.260	0.981 ± 0.014	$\mathbf{0.996 \pm 0.006}$
6	0.858 ± 0.139	0.940 ± 0.059	$\mathbf{0.995 \pm 0.004}$
8	0.962 ± 0.035	$\mathbf{0.995 \pm 0.003}$	0.981 ± 0.011

Additive GP under Monotonicity Constraints in 1000 Dimensions

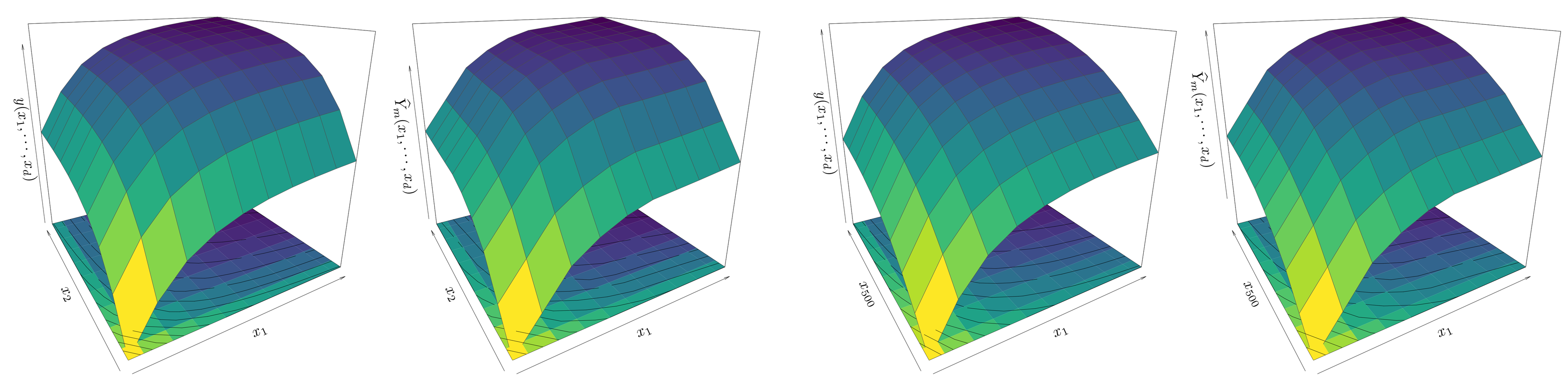
We consider the additive function:

$$y(\mathbf{x}) = \sum_{p=1}^{d} \arctan\left(5\left[1 - \frac{p}{d}\right] x_p\right), \tag{1}$$

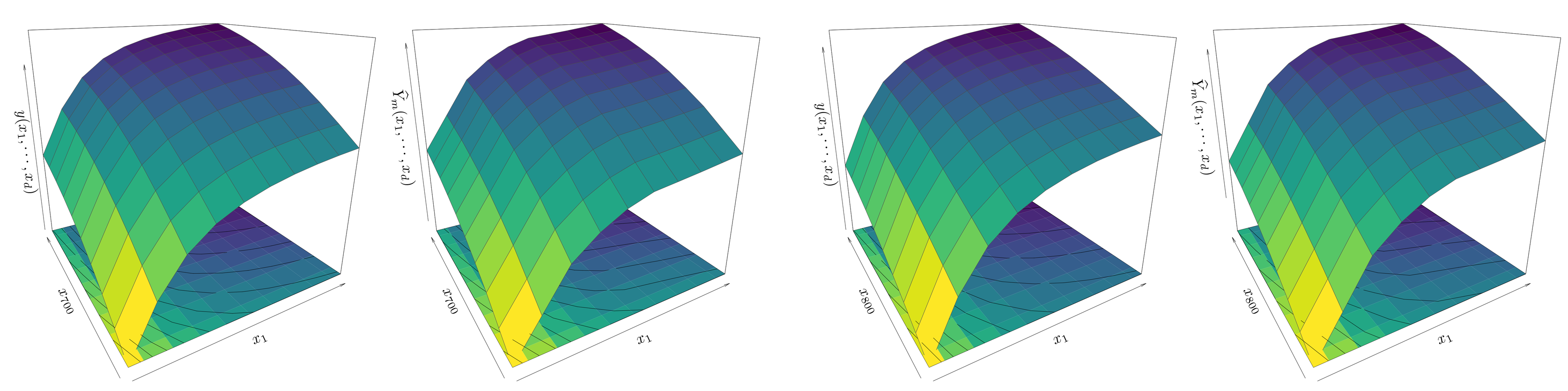
with $\mathbf{x} = (x_1, \cdots, x_d) \in [0,1]^d$. **Note:** y is completely monotone with different growth rates along each dimension. Small values of p leads to high growth rates.

Computational cost of predictions and simulations via Hamiltonian Monte Carlo (HMC) [8].

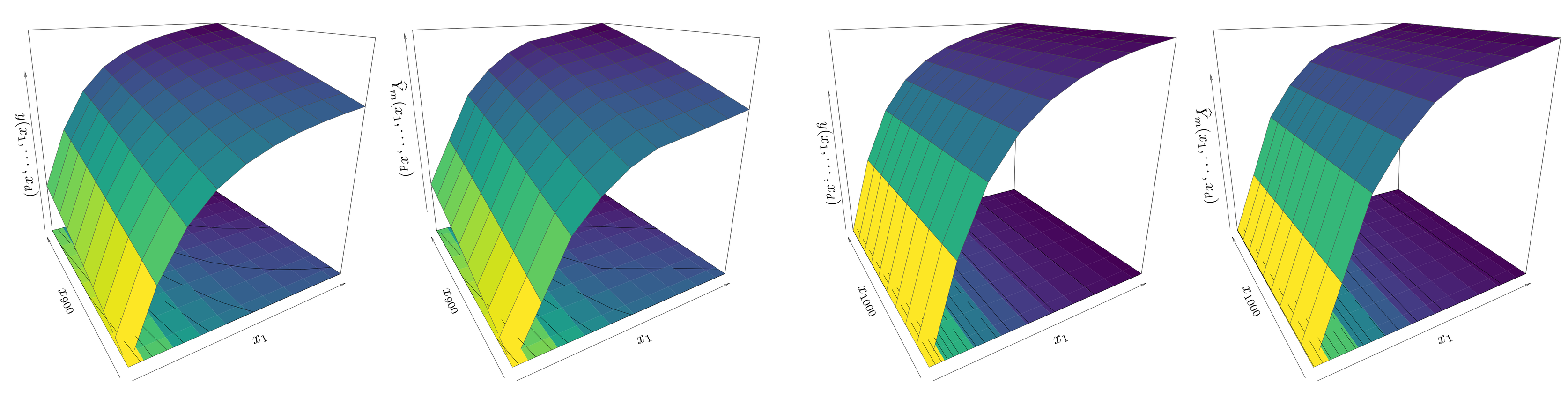
CPU Time [s]	d = 2	5	10	20	50	100	200	500	1000
Prediction	0.01	0.01	0.02	0.03	0.25	1.37	10.48	165.85	1364.54
HMC Sampling	0.05	0.11	0.22	0.72	7.14	2.91	2.73	5.28	10.83



(a) $y_1(x_1) + y_2(x_2)$ (b) $y_1(x_1) + y_{500}(x_{500})$



(c) $y_1(x_1) + y_{700}(x_{700})$ (d) $y_1(x_1) + y_{800}(x_{800})$



(e) $y_1(x_1) + y_{900}(x_{900})$ (f) $y_1(x_1) + y_{1000}(x_{1000})$

Each panel shows: the true (left) and predictive (right) mean profiles.

Further Comments

- Current version on CRAN, `lineqGPR v.0.0.4`, contains implementations from [2, 4]. Developments considering **additive models** have been added to a private beta version `lineqGPR v.0.1.0` (coming soon!).

andres-felipe.lopez@emse.fr