Metamodeling under Inequality Constraints

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

[1]École des Mines de Saint-Étienne (EMSE), France.
[2]Institut de Mathématiques de Toulouse (IMT), France.

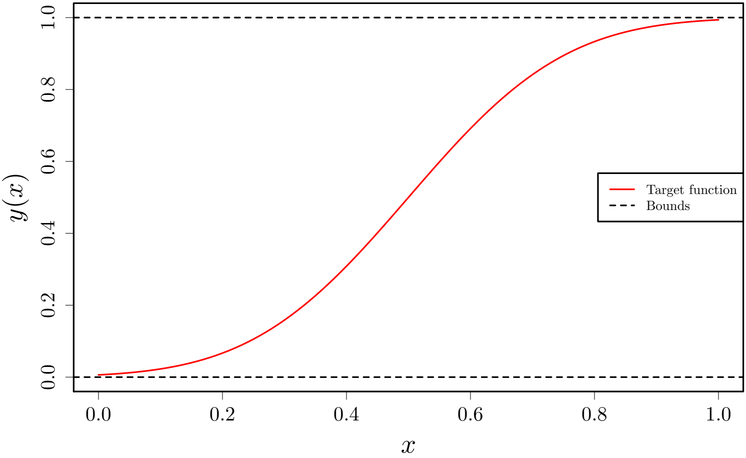
May 10, 2017

Table of Contents

Table of Contents

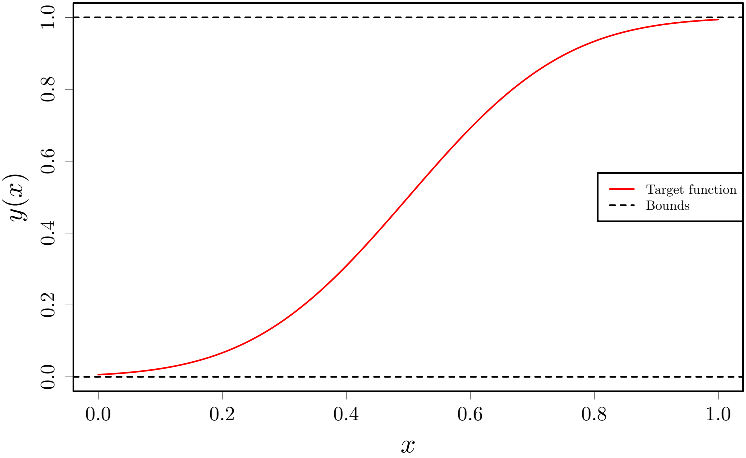
Motivation

Toy example.



Motivation

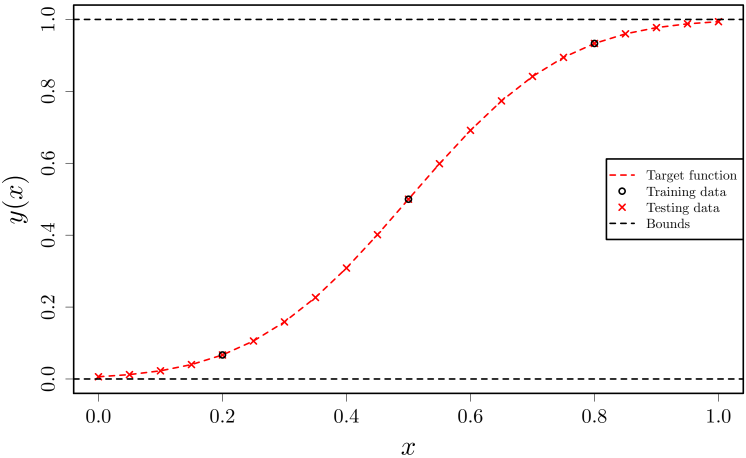
Toy example.



$\Rightarrow$ The target function is bounded and monotonic!!

Motivation

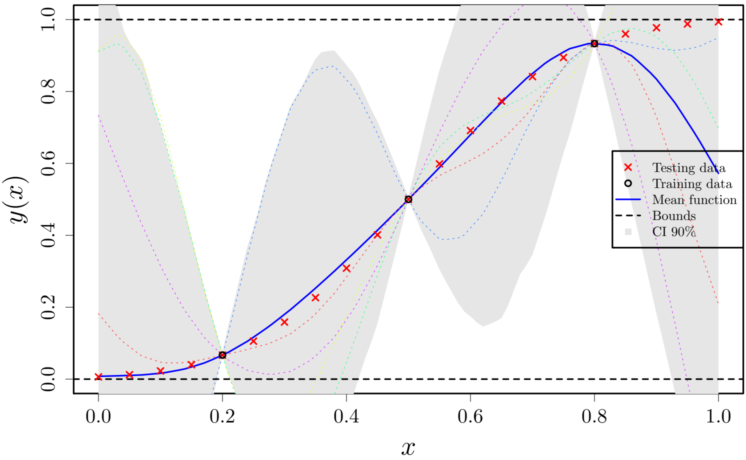
Toy example.



Data

Motivation

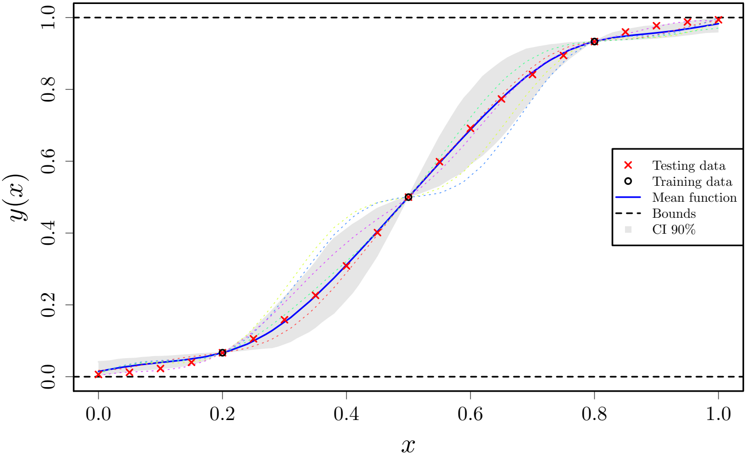
Toy example.



Standard Kriging

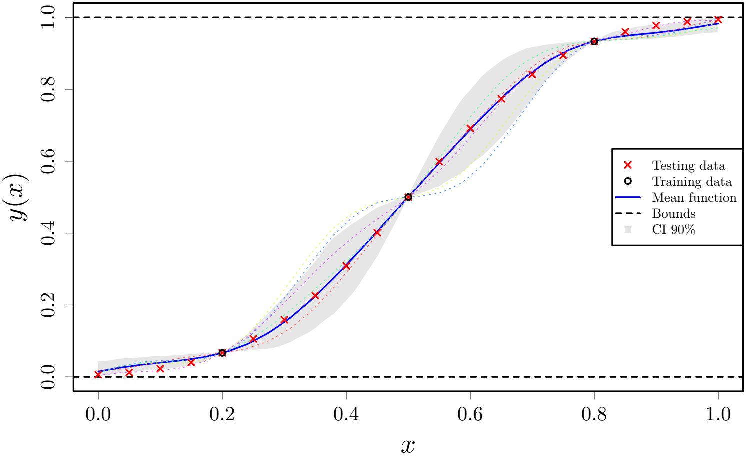
Motivation

Toy example.



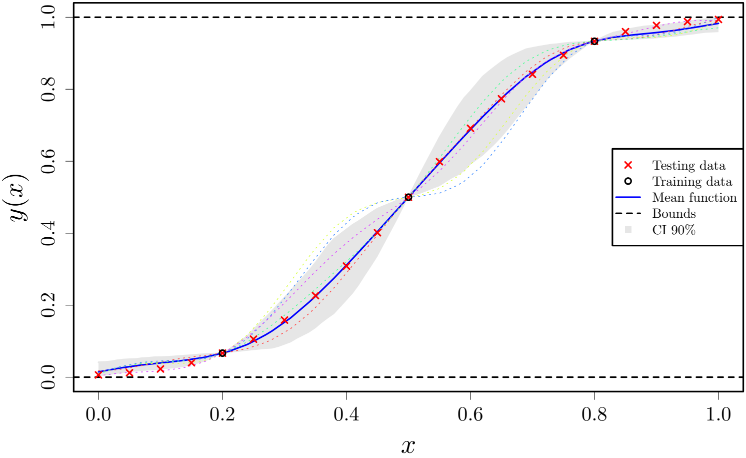
Motivation

Toy example. $y(x) = \frac{1}{2}\left[1 + \operatorname{erf}\left\{\frac{x-0.5}{0.2\sqrt{2}}\right\}\right].$



Motivation

Toy example. $y(x) = \Phi(\frac{x-0.5}{0.2})$ (Gaussian CDF).



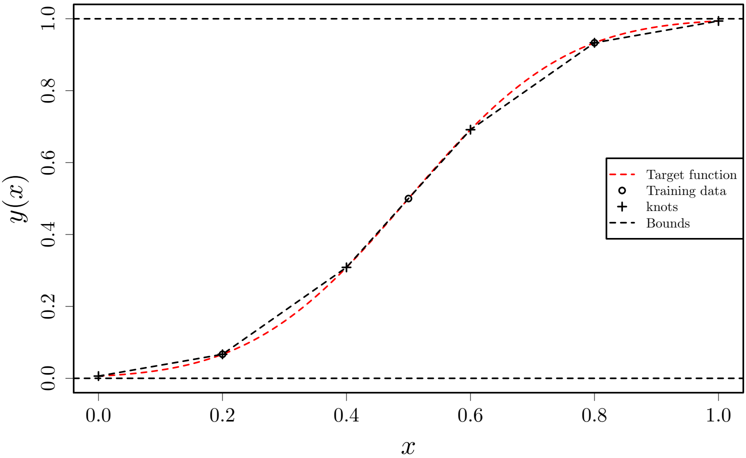
Constrained Kriging

Table of Contents

Table of Contents

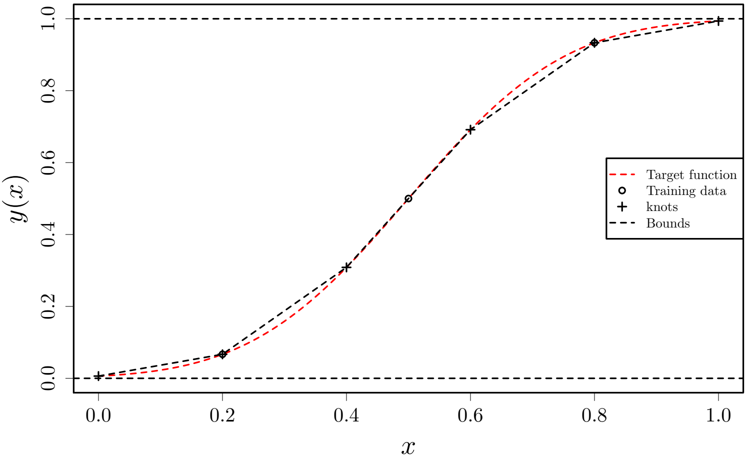
Constrained Kriging: finite approximation

Figure: toy example $y(x) = \Phi(\frac{x-0.5}{0.2})$ (Gaussian CDF).



Constrained Kriging: finite approximation

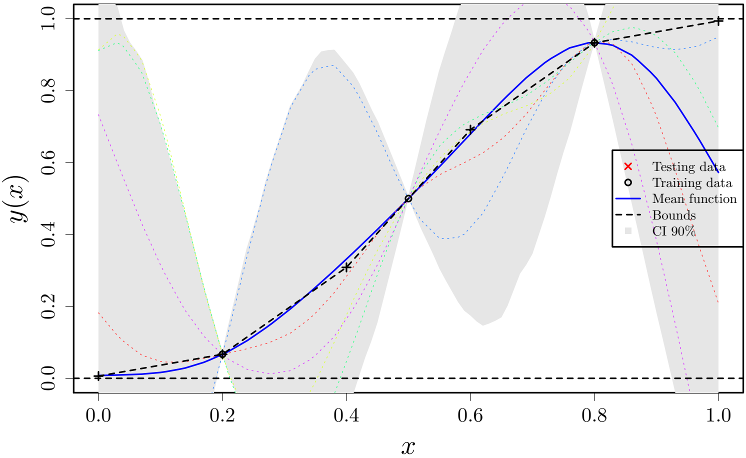
Figure: toy example $y(x) = \Phi(\frac{x-0.5}{0.2})$ (Gaussian CDF).



$\Rightarrow$ The finite approx. is also bounded and monotonic!!

Constrained Kriging: finite approximation

Figure: toy example $y(x) = \Phi(\frac{x-0.5}{0.2})$ (Gaussian CDF).



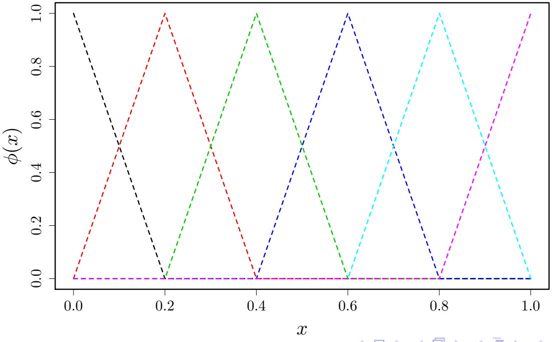
$\Rightarrow$ We can assume that the knots' images are random variables.

Constrained Kriging: Maatouk and Bay (2016)

Let the finite-one-dimensional GP-based approximation be defined as

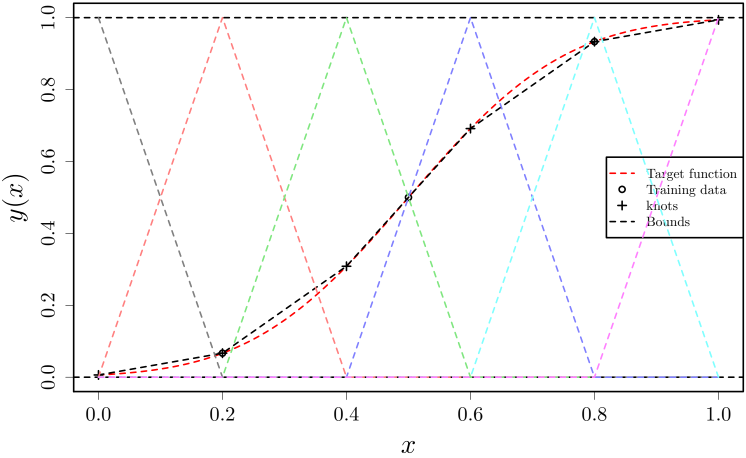
$$y(x) \approx \sum_{j=0}^{m-1} \xi_j \phi_j(x), \qquad x \in [0,1], \qquad \text{subject to} \quad \boldsymbol{\xi} \in \mathcal{C}, \qquad (1)$$

where $\boldsymbol{\xi} = \left[\xi_0,\ \xi_1,\ \cdots,\ \xi_{m-1}\right]^\top$ (knots' images) is a zero-mean Gaussian vector with covariance matrix $\boldsymbol{\Gamma} \in \mathbb{R}^{m \times m}$, and $\phi_j : [0,1] \to \mathbb{R}$ are hat functions.



Constrained Kriging: finite approximation

Figure: toy example $y(x) = \Phi(\frac{x-0.5}{0.2})$ (Gaussian CDF).



Constrained Kriging: Maatouk and Bay (2016)

- We are interested on convex sets of the form $\mathcal{C} : \boldsymbol{l} \leq \boldsymbol{\Lambda\xi} \leq \boldsymbol{u}$, where $\boldsymbol{\Lambda} \in \mathbb{R}^{p \times m}$, and $\boldsymbol{l}$ and $\boldsymbol{u}$ represent the lower and the upper bounds, respectively. Then, we have

$$\mathbf{y} \approx \boldsymbol{\Phi\xi}, \quad \boldsymbol{\xi} \sim \mathcal{N}\left(\mathbf{0}, \boldsymbol{\Gamma}\right), \quad \text{subject to} \quad \boldsymbol{l} \leq \boldsymbol{\Lambda\xi} \leq \boldsymbol{u}. \tag{2}$$

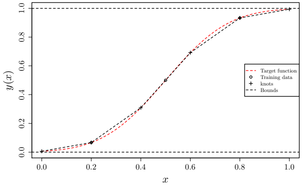
Constrained Kriging: Maatouk and Bay (2016)

✦ We are interested on convex sets of the form $\mathcal{C} : \boldsymbol{l} \leq \boldsymbol{\Lambda\xi} \leq \boldsymbol{u}$, where $\boldsymbol{\Lambda} \in \mathbb{R}^{p \times m}$, and $\boldsymbol{l}$ and $\boldsymbol{u}$ represent the lower and the upper bounds, respectively. Then, we have

$$\mathbf{y} \approx \boldsymbol{\Phi\xi}, \quad \boldsymbol{\xi} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Gamma}), \quad \text{subject to} \quad \boldsymbol{l} \leq \boldsymbol{\Lambda\xi} \leq \boldsymbol{u}. \tag{2}$$

Boundedness constraint

$$\underbrace{\begin{bmatrix}0\\0\\0\\0\\0\end{bmatrix}}_{\boldsymbol{l}_b} \leq \underbrace{\begin{bmatrix}1&0&0&0&0\\0&1&0&0&0\\0&0&1&0&0\\0&0&0&1&0\\0&0&0&0&1\end{bmatrix}}_{\boldsymbol{\Lambda}_b} \underbrace{\begin{bmatrix}\xi_1\\\xi_2\\\xi_3\\\xi_4\\\xi_5\end{bmatrix}}_{\boldsymbol{\xi}} \leq \underbrace{\begin{bmatrix}1\\1\\1\\1\\1\end{bmatrix}}_{\boldsymbol{u}_b}$$



The finite approximation is also bounded...

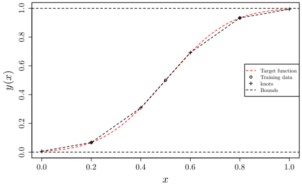
Constrained Kriging: Maatouk and Bay (2016)

✦ We are interested on convex sets of the form $\mathcal{C} : \boldsymbol{l} \leq \boldsymbol{\Lambda}\boldsymbol{\xi} \leq \boldsymbol{u}$, where $\boldsymbol{\Lambda} \in \mathbb{R}^{p \times m}$, and $\boldsymbol{l}$ and $\boldsymbol{u}$ represent the lower and the upper bounds, respectively. Then, we have

$$\mathbf{y} \approx \boldsymbol{\Phi}\boldsymbol{\xi}, \quad \boldsymbol{\xi} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Gamma}), \quad \text{subject to} \quad \boldsymbol{l} \leq \boldsymbol{\Lambda}\boldsymbol{\xi} \leq \boldsymbol{u}. \qquad (2)$$

Monotonicity constraint

$$\underbrace{\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}}_{\boldsymbol{l}_m} \leq \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 \\ 0 & 0 & 0 & -1 & 1 \end{bmatrix}}_{\boldsymbol{\Lambda}_m} \underbrace{\begin{bmatrix} \xi_1 \\ \xi_2 \\ \xi_3 \\ \xi_4 \\ \xi_5 \end{bmatrix}}_{\boldsymbol{\xi}} < \underbrace{\begin{bmatrix} \infty \\ \infty \\ \infty \\ \infty \\ \infty \end{bmatrix}}_{\boldsymbol{u}_m}$$



The finite approximation is also bounded and monotonic.

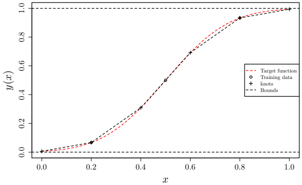
Constrained Kriging: Maatouk and Bay (2016)

✦ We are interested on convex sets of the form $\mathcal{C} : \boldsymbol{l} \leq \boldsymbol{\Lambda}\boldsymbol{\xi} \leq \boldsymbol{u}$, where $\boldsymbol{\Lambda} \in \mathbb{R}^{p \times m}$, and $\boldsymbol{l}$ and $\boldsymbol{u}$ represent the lower and the upper bounds, respectively. Then, we have

$$\mathbf{y} \approx \boldsymbol{\Phi}\boldsymbol{\xi}, \quad \boldsymbol{\xi} \sim \mathcal{N}\left(\mathbf{0}, \boldsymbol{\Gamma}\right), \quad \text{subject to} \quad \boldsymbol{l} \leq \boldsymbol{\Lambda}\boldsymbol{\xi} \leq \boldsymbol{u}. \tag{2}$$

Boundedness + Monotonicity

$$\boldsymbol{l} = \begin{bmatrix} \boldsymbol{l}_b \\ \boldsymbol{l}_m \end{bmatrix}, \quad \boldsymbol{\Lambda} = \begin{bmatrix} \boldsymbol{\Lambda}_b \\ \boldsymbol{\Lambda}_m \end{bmatrix}, \quad \boldsymbol{u} = \begin{bmatrix} \boldsymbol{u}_b \\ \boldsymbol{u}_m \end{bmatrix}.$$



The finite approximation is also bounded and monotonic.

Constrained Kriging: Maatouk and Bay (2016)

Finally, we obtain

$$\boldsymbol{\xi}_{\mathcal{C}} | \mathbf{y} \sim \mathcal{TN}\left(\boldsymbol{\Lambda}\boldsymbol{\mu}_{\boldsymbol{\xi}|\mathbf{y}}, \boldsymbol{\Lambda}\boldsymbol{\Sigma}_{\boldsymbol{\xi}|\mathbf{y}}\boldsymbol{\Lambda}^\top, \boldsymbol{l}, \boldsymbol{u}\right),$$

where $\boldsymbol{\xi}_{\mathcal{C}} : \boldsymbol{l} \leq \boldsymbol{\Lambda}\boldsymbol{\xi}_{\mathcal{C}} \leq \boldsymbol{u}$, and

$$\boldsymbol{\mu}_{\boldsymbol{\xi}|\mathbf{y}} = \boldsymbol{\Gamma}\boldsymbol{\Phi}^\top(\boldsymbol{\Phi}\boldsymbol{\Gamma}\boldsymbol{\Phi}^\top)^{-1}\mathbf{y},$$
$$\boldsymbol{\Sigma}_{\boldsymbol{\xi}|\mathbf{y}} = \boldsymbol{\Gamma} - \boldsymbol{\Gamma}\boldsymbol{\Phi}^\top(\boldsymbol{\Phi}\boldsymbol{\Gamma}\boldsymbol{\Phi}^\top)^{-1}\boldsymbol{\Phi}\boldsymbol{\Gamma}.$$

Constrained Kriging: Maatouk and Bay (2016)

Finally, we obtain

$$\boldsymbol{\xi}_{\mathcal{C}}|\mathbf{y} \sim \mathcal{TN}\left(\boldsymbol{\Lambda}\boldsymbol{\mu}_{\boldsymbol{\xi}|\mathbf{y}}, \boldsymbol{\Lambda}\boldsymbol{\Sigma}_{\boldsymbol{\xi}|\mathbf{y}}\boldsymbol{\Lambda}^\top, \boldsymbol{l}, \boldsymbol{u}\right), \checkmark$$

where $\boldsymbol{\xi}_{\mathcal{C}} : \boldsymbol{l} \leq \boldsymbol{\Lambda}\boldsymbol{\xi}_{\mathcal{C}} \leq \boldsymbol{u}$, and

$$\boldsymbol{\mu}_{\boldsymbol{\xi}|\mathbf{y}} = \boldsymbol{\Gamma}\boldsymbol{\Phi}^\top(\boldsymbol{\Phi}\boldsymbol{\Gamma}\boldsymbol{\Phi}^\top)^{-1}\mathbf{y},$$
$$\boldsymbol{\Sigma}_{\boldsymbol{\xi}|\mathbf{y}} = \boldsymbol{\Gamma} - \boldsymbol{\Gamma}\boldsymbol{\Phi}^\top(\boldsymbol{\Phi}\boldsymbol{\Gamma}\boldsymbol{\Phi}^\top)^{-1}\boldsymbol{\Phi}\boldsymbol{\Gamma}.$$

$\Rightarrow$ Maatouk and Bay (2016) proposed an MC approach based on rejection sampling known as *Rejection Sampling from the Mode* (RSM).

Constrained Kriging: Maatouk and Bay (2016)

Maatouk and Bay (2016) proposed an MC approach based on rejection sampling known as *Rejection Sampling from the Mode* (RSM).

✓ **Advantages**

- It is an exact approach for sampling (uncorrelated samples).
- It is easy to implement.

Constrained Kriging: Maatouk and Bay (2016)

Maatouk and Bay (2016) proposed an MC approach based on rejection sampling known as *Rejection Sampling from the Mode* (RSM).

✓ **Advantages**

- It is an exact approach for sampling (uncorrelated samples).
- It is easy to implement.

✗ **Disadvantages**

- However, simulating from a truncated multivariate Gaussian is required: for higher dimensions the acceptance rate is smaller!

Constrained Kriging: Maatouk and Bay (2016)

✦ Aim

- To simulate from truncated high dimensional Gaussians!

Constrained Kriging: Maatouk and Bay (2016)

✦ **Aim**

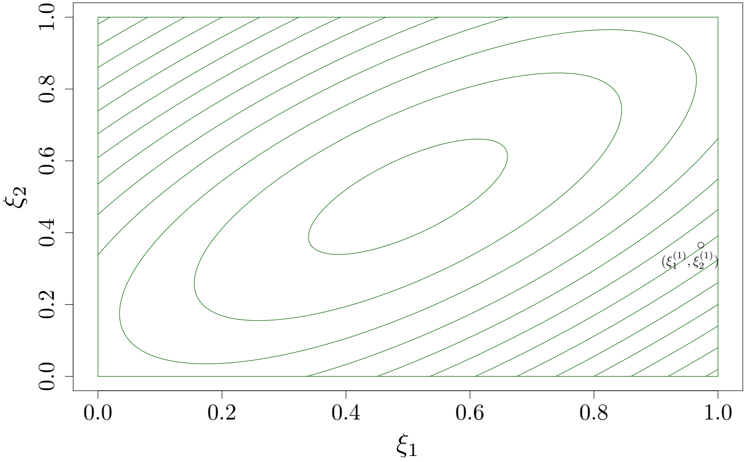
- To simulate from truncated high dimensional Gaussians!

⇒ Let's call some MCMC techniques! (Bishop, 2007; Murphy, 2012)
 - Gibbs sampling.
 - Metropolis-Hastings (MH) algorithm.
 - Hamiltonian Monte Carlo (HMC).

Table of Contents

MCMC: Gibbs Sampling (Gibbs)

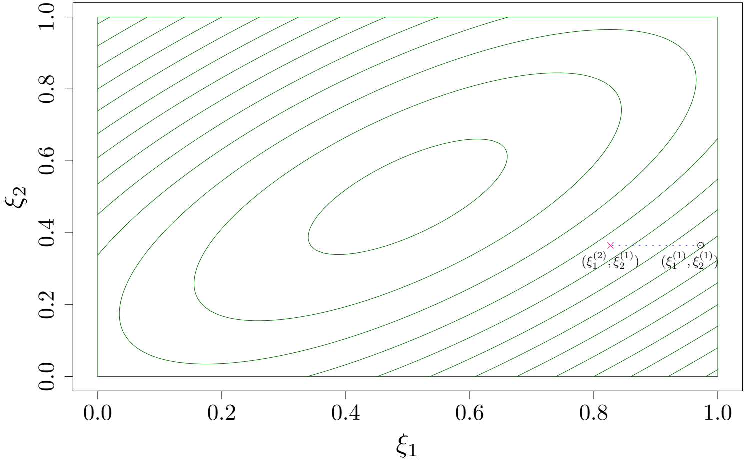
Figure: Solid red lines represent the contour lines of the bivariate Gaussian $(\xi_1, \xi_2) \sim \mathcal{TN}([0.5\ 0.5], [1.0\ 0.7; 0.7\ 1.0], [0.0\ 0.0], [1.0\ 1.0])$.



Iteration 1

MCMC: Gibbs Sampling (Gibbs)

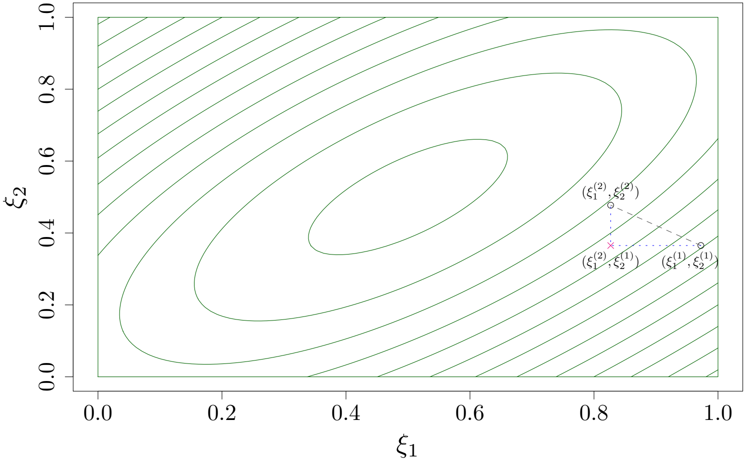
Figure: Solid red lines represent the contour lines of the bivariate Gaussian $(\xi_1, \xi_2) \sim \mathcal{TN}([0.5\ 0.5], [1.0\ 0.7; 0.7\ 1.0], [0.0\ 0.0], [1.0\ 1.0])$.



Iteration 2: simulating the 1st coordinate using $p(\xi_1^{(2)}|\xi_2^{(1)})$

MCMC: Gibbs Sampling (Gibbs)

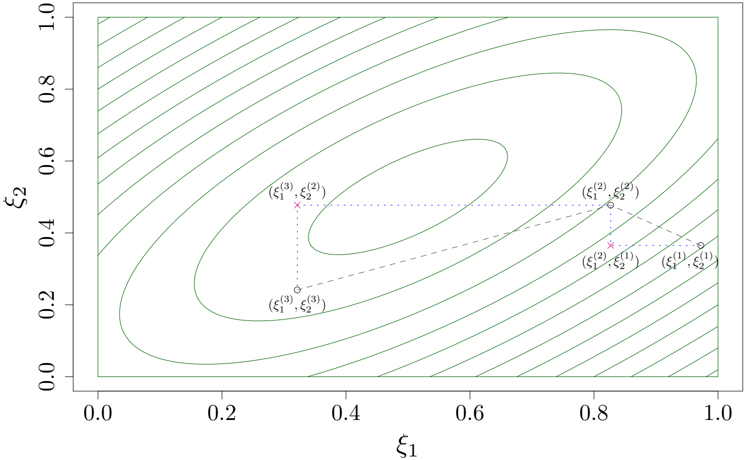
Figure: Solid red lines represent the contour lines of the bivariate Gaussian $(\xi_1, \xi_2) \sim \mathcal{TN}([0.5\ 0.5], [1.0\ 0.7; 0.7\ 1.0], [0.0\ 0.0], [1.0\ 1.0])$.



Iteration 2: simulating the 2nd coordinate using $p(\xi_2^{(2)} | \xi_1^{(2)})$

MCMC: Gibbs Sampling (Gibbs)

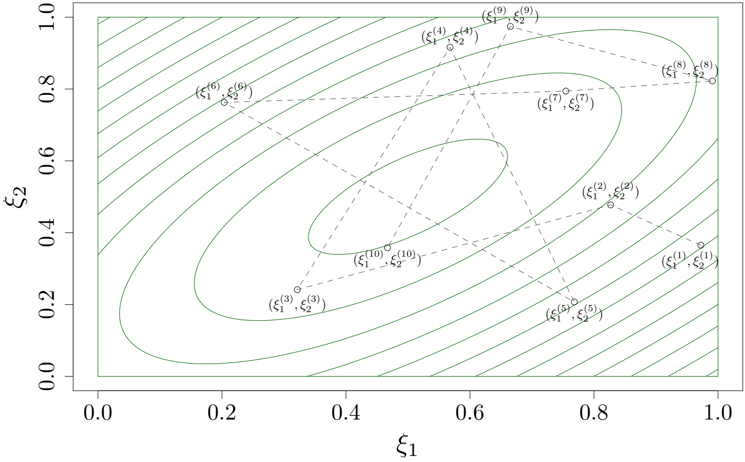
Figure: Solid red lines represent the contour lines of the bivariate Gaussian $(\xi_1, \xi_2) \sim \mathcal{TN}\left([0.5\ 0.5], [1.0\ 0.7; 0.7\ 1.0], [0.0\ 0.0], [1.0\ 1.0]\right)$.



Iteration 3

MCMC: Gibbs Sampling (Gibbs)

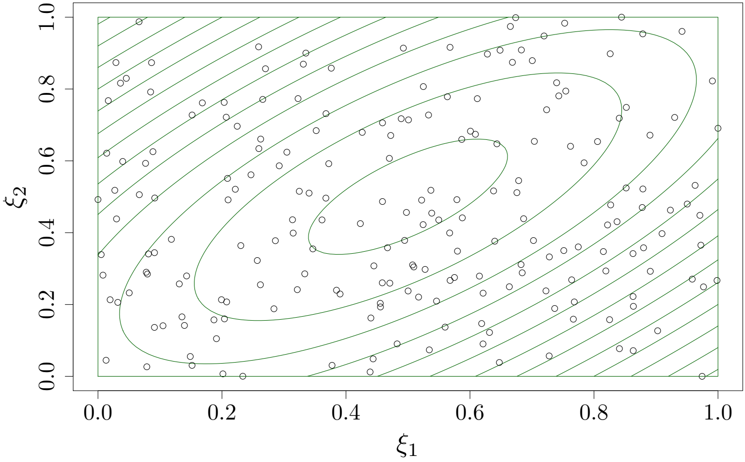
Figure: Solid red lines represent the contour lines of the bivariate Gaussian $(\xi_1, \xi_2) \sim \mathcal{TN}\,([0.5\ 0.5], [1.0\ 0.7; 0.7\ 1.0], [0.0\ 0.0], [1.0\ 1.0])$.



Iteration 10

MCMC: Gibbs Sampling (Gibbs)

Figure: Solid red lines represent the contour lines of the bivariate Gaussian $(\xi_1, \xi_2) \sim \mathcal{TN}([0.5\ 0.5], [1.0\ 0.7; 0.7\ 1.0], [0.0\ 0.0], [1.0\ 1.0])$.



Iteration 200

MCMC: Gibbs Sampling (Gibbs)

✓ **Advantages**

- Unlike RSM, there is no rejection step.
- Simulating from a truncated multivariate Gaussian is reduced to a sequential sampling from truncated univariate Gaussians.
- There are efficient implementations in R (e.g `tmvtnorm` package).

MCMC: Gibbs Sampling (Gibbs)

✓ **Advantages**

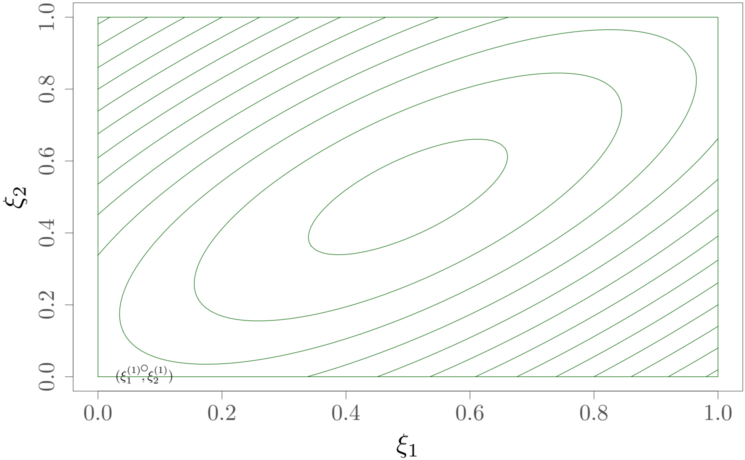
- Unlike RSM, there is no rejection step.
- Simulating from a truncated multivariate Gaussian is reduced to a sequential sampling from truncated univariate Gaussians.
- There are efficient implementations in R (e.g `tmvtnorm` package).

✗ **Disadvantages**

- It is necessary to discard intermediate simulations to obtain less correlated samples (*thinning* effect).

MCMC: Metropolis-Hastings (MH)

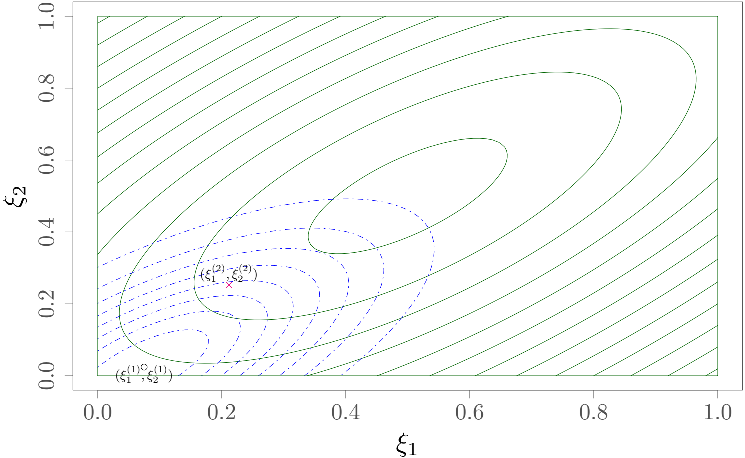
Figure: Solid red lines represent the contour lines of the bivariate Gaussian $(\xi_1, \xi_2) \sim \mathcal{TN}\left([0.5\ 0.5], [1.0\ 0.7; 0.7\ 1.0], [0.0\ 0.0], [1.0\ 1.0]\right)$.



Iteration 1

MCMC: Metropolis-Hastings (MH)

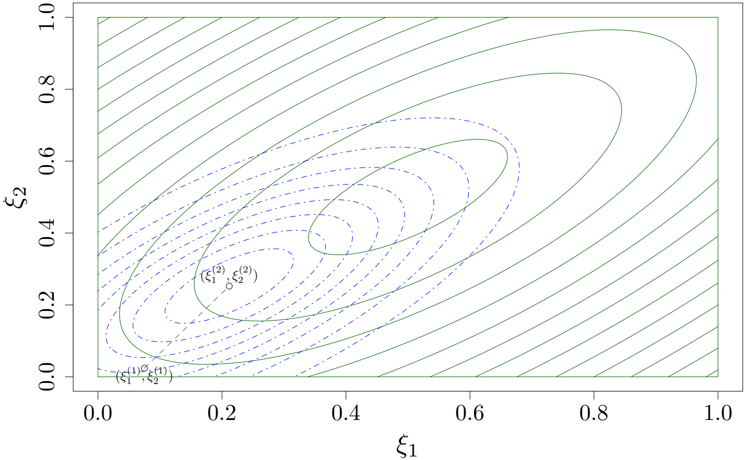
Figure: Solid red lines represent the contour lines of the bivariate Gaussian $(\xi_1, \xi_2) \sim \mathcal{TN}([0.5\ 0.5], [1.0\ 0.7; 0.7\ 1.0], [0.0\ 0.0], [1.0\ 1.0])$.



Iteration 2: simulating from $q(\boldsymbol{\xi}^{(2)}|\boldsymbol{\xi}^{(1)}) = \mathcal{N}\left(\boldsymbol{\xi}^{(2)}|\boldsymbol{\xi}^{(1)}, \eta\boldsymbol{\Sigma}\right)$

MCMC: Metropolis-Hastings (MH)

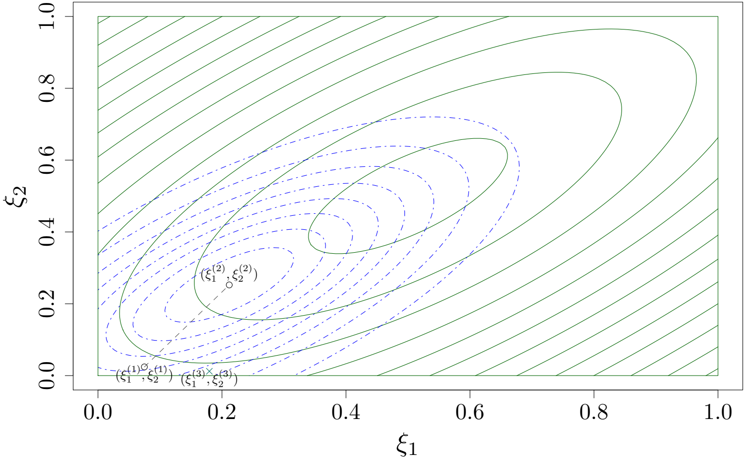
Figure: Solid red lines represent the contour lines of the bivariate Gaussian $(\xi_1, \xi_2) \sim \mathcal{TN}([0.5\ 0.5], [1.0\ 0.7; 0.7\ 1.0], [0.0\ 0.0], [1.0\ 1.0])$.



Iteration 2: updating according to $\alpha = \tilde{p}(\boldsymbol{\xi}^{(2)})/\tilde{p}(\boldsymbol{\xi}^{(1)})$

MCMC: Metropolis-Hastings (MH)

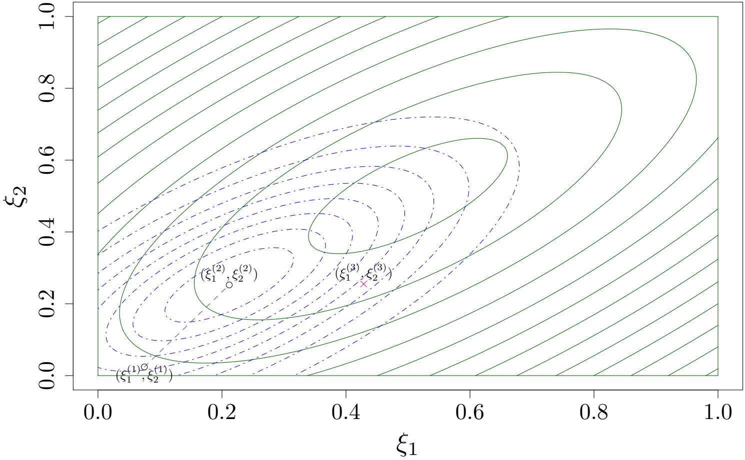
Figure: Solid red lines represent the contour lines of the bivariate Gaussian $(\xi_1, \xi_2) \sim \mathcal{TN}\left([0.5\ 0.5], [1.0\ 0.7; 0.7\ 1.0], [0.0\ 0.0], [1.0\ 1.0]\right)$.



Iteration 3: simulating from $q(\boldsymbol{\xi}^{(3)}|\boldsymbol{\xi}^{(2)}) = \mathcal{N}\left(\boldsymbol{\xi}^{(3)}|\boldsymbol{\xi}^{(2)}, \eta\boldsymbol{\Sigma}\right)$

MCMC: Metropolis-Hastings (MH)

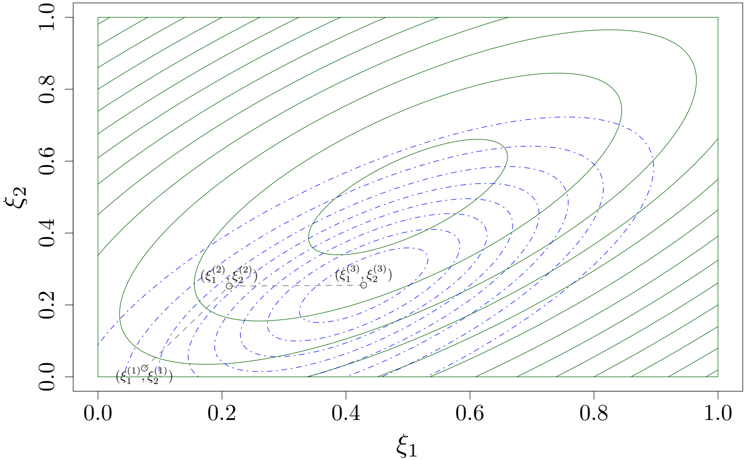
Figure: Solid red lines represent the contour lines of the bivariate Gaussian $(\xi_1, \xi_2) \sim \mathcal{TN}\left([0.5\ 0.5], [1.0\ 0.7; 0.7\ 1.0], [0.0\ 0.0], [1.0\ 1.0]\right)$.



Iteration 3: resampling from $q(\boldsymbol{\xi}^{(3)}|\boldsymbol{\xi}^{(2)}) = \mathcal{N}\left(\boldsymbol{\xi}^{(3)}|\boldsymbol{\xi}^{(2)}, \eta\boldsymbol{\Sigma}\right)$

MCMC: Metropolis-Hastings (MH)

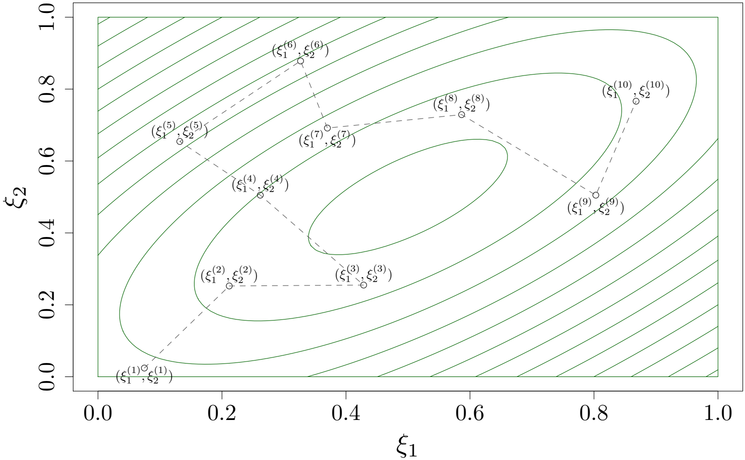
Figure: Solid red lines represent the contour lines of the bivariate Gaussian $(\xi_1, \xi_2) \sim \mathcal{TN}\left([0.5\ 0.5], [1.0\ 0.7; 0.7\ 1.0], [0.0\ 0.0], [1.0\ 1.0]\right)$.



Iteration 3: updating according to $\alpha = \tilde{p}(\boldsymbol{\xi}^{(2)})/\tilde{p}(\boldsymbol{\xi}^{(1)})$

MCMC: Metropolis-Hastings (MH)

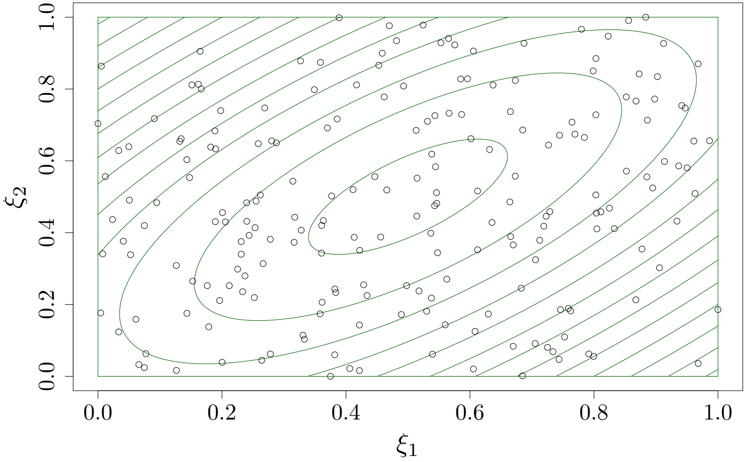
Figure: Solid red lines represent the contour lines of the bivariate Gaussian $(\xi_1, \xi_2) \sim \mathcal{TN}\,([0.5\ 0.5], [1.0\ 0.7; 0.7\ 1.0], [0.0\ 0.0], [1.0\ 1.0])$.



Iteration 10

MCMC: Metropolis-Hastings (MH)

Figure: Solid red lines represent the contour lines of the bivariate Gaussian $(\xi_1, \xi_2) \sim \mathcal{TN}([0.5\ 0.5], [1.0\ 0.7; 0.7\ 1.0], [0.0\ 0.0], [1.0\ 1.0])$.



Iteration 200

MCMC: Metropolis-Hastings (MH)

✓ **Advantages**

- By properly tuning the value of the scale factor η, it is possible to simulate less correlated samples.

MCMC: Metropolis-Hastings (MH)

✓ **Advantages**

- By properly tuning the value of the scale factor η, it is possible to simulate less correlated samples.

✗ **Disadvantages**

- The scale factor η has to be tuned.
- Like RSM, simulating from a truncated multivariate Gaussian is required: higher dimensions smaller acceptance rate!

MCMC: Hamiltonian Monte Carlo (HMC)

- Duane et al. (1987) introduced an efficient hybrid approach using the properties of the Hamiltonian dynamics.
- According to physical systems, HMC is employed to simulate

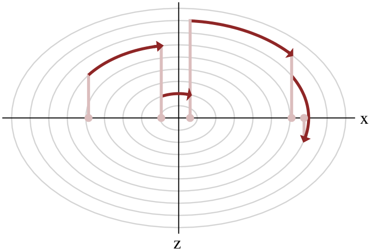
$$p(\mathbf{x}) \propto \exp\left\{-\frac{U(\mathbf{x})}{T}\right\}, \tag{3}$$

where $U(\mathbf{x})$ is the *energy* of the state $\mathbf{x}$, and T is the temperature.
- For the Gaussian case, $p(\mathbf{x}) = \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$,

$$U(\mathbf{x}) = \frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu}) \quad \text{and} \quad T = 1.$$

MCMC: Hamiltonian Monte Carlo (HMC)

Figure: HMC in 1-dimensional example. Figures from (Betancourt, 2017).

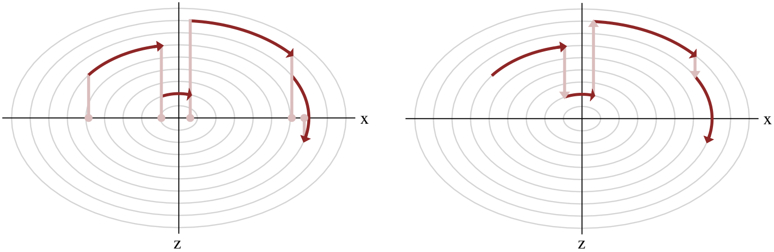


$$H = U(\mathbf{x}) + K(\mathbf{z}) = \frac{1}{2}\mathbf{x}^\top \boldsymbol{\Sigma}^{-1}\mathbf{x} + \frac{1}{2}\mathbf{z}^\top \mathbf{M}^{-1}\mathbf{z},$$

where $\mathbf{z}$ is a vector with the auxiliary *momentum* variables.

MCMC: Hamiltonian Monte Carlo (HMC)

Figure: HMC in 1-dimensional example. Figures from (Betancourt, 2017).



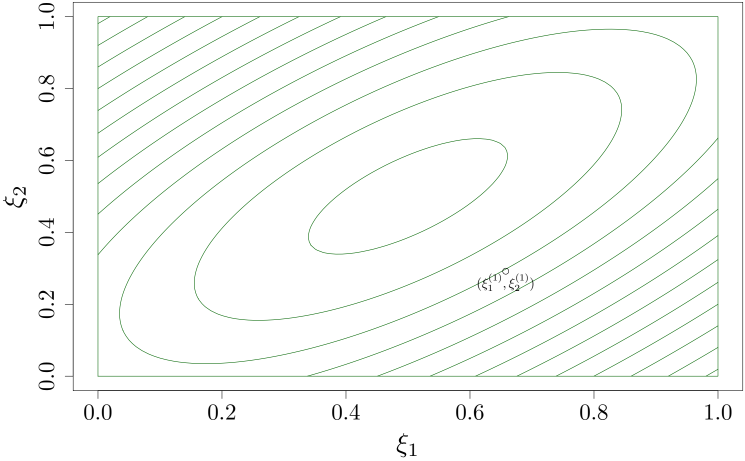
$$H = U(\mathbf{x}) + K(\mathbf{z}) = \frac{1}{2}\mathbf{x}^\top \boldsymbol{\Sigma}^{-1}\mathbf{x} + \frac{1}{2}\mathbf{z}^\top \mathbf{M}^{-1}\mathbf{z},$$

where $\mathbf{z}$ is a vector with the auxiliary *momentum* variables.

Toy example on-line: [url]

MCMC: Hamiltonian Monte Carlo (HMC)

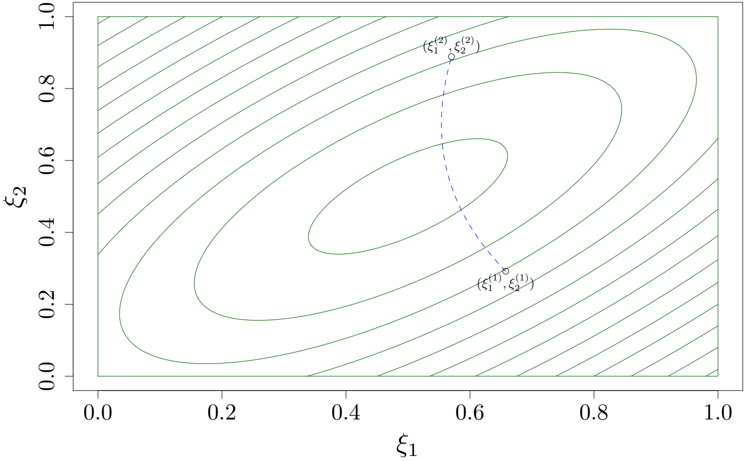
Figure: Solid red lines represent the contour lines of the bivariate Gaussian $(\xi_1, \xi_2) \sim \mathcal{TN}\left([0.5\ 0.5], [1.0\ 0.7; 0.7\ 1.0], [0.0\ 0.0], [1.0\ 1.0]\right)$.



Iteration 1

MCMC: Hamiltonian Monte Carlo (HMC)

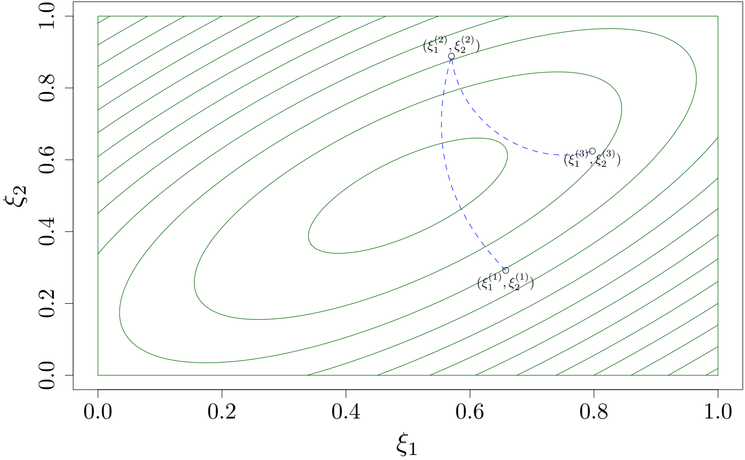
Figure: Solid red lines represent the contour lines of the bivariate Gaussian $(\xi_1, \xi_2) \sim \mathcal{TN}([0.5\ 0.5], [1.0\ 0.7; 0.7\ 1.0], [0.0\ 0.0], [1.0\ 1.0])$.



Iteration 2

MCMC: Hamiltonian Monte Carlo (HMC)

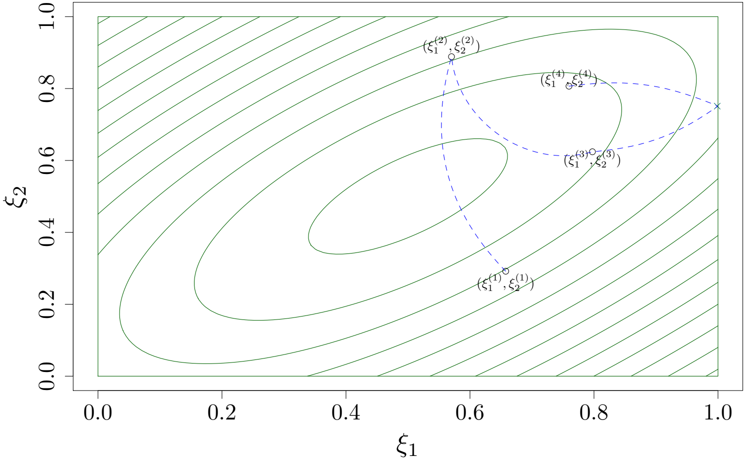
Figure: Solid red lines represent the contour lines of the bivariate Gaussian $(\xi_1, \xi_2) \sim \mathcal{TN}([0.5\ 0.5], [1.0\ 0.7; 0.7\ 1.0], [0.0\ 0.0], [1.0\ 1.0])$.



Iteration 3

MCMC: Hamiltonian Monte Carlo (HMC)

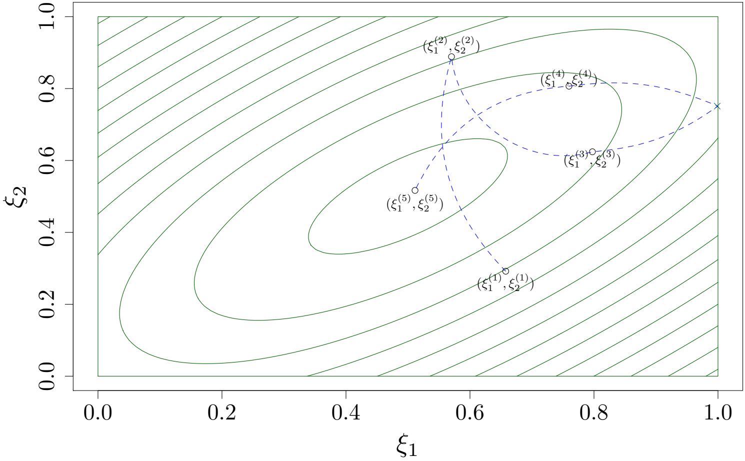
Figure: Solid red lines represent the contour lines of the bivariate Gaussian $(\xi_1, \xi_2) \sim \mathcal{TN}([0.5\ 0.5], [1.0\ 0.7; 0.7\ 1.0], [0.0\ 0.0], [1.0\ 1.0])$.



Iteration 4

MCMC: Hamiltonian Monte Carlo (HMC)

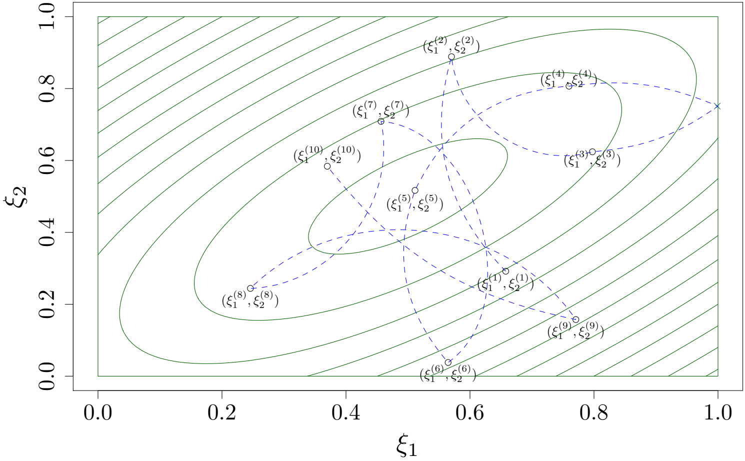
Figure: Solid red lines represent the contour lines of the bivariate Gaussian $(\xi_1, \xi_2) \sim \mathcal{TN}([0.5\ 0.5], [1.0\ 0.7; 0.7\ 1.0], [0.0\ 0.0], [1.0\ 1.0])$.



Iteration 5

MCMC: Hamiltonian Monte Carlo (HMC)

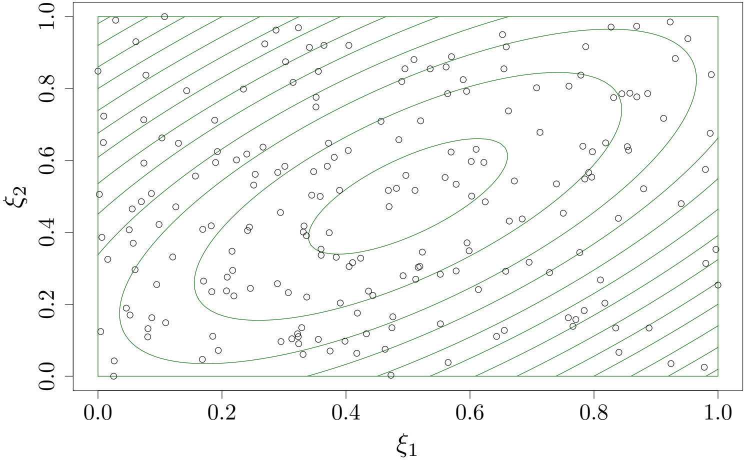
Figure: Solid red lines represent the contour lines of the bivariate Gaussian $(\xi_1, \xi_2) \sim \mathcal{TN}([0.5\ 0.5], [1.0\ 0.7; 0.7\ 1.0], [0.0\ 0.0], [1.0\ 1.0])$.



Iteration 10

MCMC: Hamiltonian Monte Carlo (HMC)

Figure: Solid red lines represent the contour lines of the bivariate Gaussian $(\xi_1, \xi_2) \sim \mathcal{TN}\left([0.5\ 0.5], [1.0\ 0.7; 0.7\ 1.0], [0.0\ 0.0], [1.0\ 1.0]\right)$.



Iteration 200

MCMC: Hamiltonian Monte Carlo (HMC)

✓ Advantages

- It is an efficient approach to simulate less correlated samples.
- There is an exact HMC for sampling from truncated Gaussians with linear inequality constraints (Pakman and Paninski, 2014).
- The approach from (Pakman and Paninski, 2014) is already implemented in R: `tmg` package.
- It is the trending topic in the state-of-the-art.

MCMC: Hamiltonian Monte Carlo (HMC)

✓ **Advantages**

- It is an efficient approach to simulate less correlated samples.
- There is an exact HMC for sampling from truncated Gaussians with linear inequality constraints (Pakman and Paninski, 2014).
- The approach from (Pakman and Paninski, 2014) is already implemented in R: `tmg` package.
- It is the trending topic in the state-of-the-art.

✗ **Disadvantages**

- Some parameters have to be tunned (e.g. travel time between samples).
- Sometimes it requires some expensive computations (e.g. computing hitting instants and reflection velocities) (Pakman and Paninski, 2014).

MCMC: Hamiltonian Monte Carlo (HMC)

Table: Comparison between proposed MCMC techniques w.r.t. the RSM approach

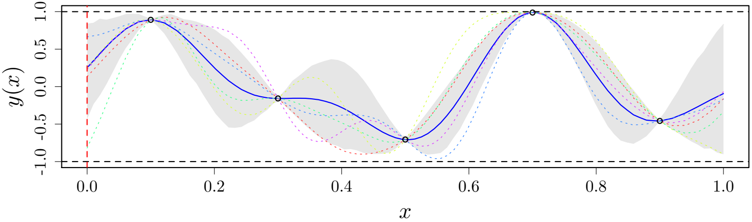
	RSM	Gibbs	MH	HMC
Exact method	✓	✗	✗	✗
Non parametric	✓	✓	✗	✗✓
Acceptance rate	✗	✓✓	✗-✓	✓
Speed	✗	✓	✗-✓	✓
Uncorrelated samples	-	✗	✗-✓	✓
Package	-	`tmvtnorm`	-	`tmg`

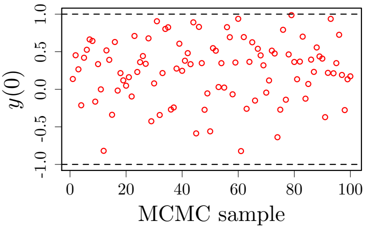
RSM: Rejection Sampling from the Mode
MH: Metropolis-Hastings
HMC: Hamiltonian Monte Carlo

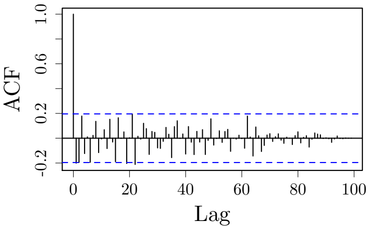
Partial results: simulating from the model

Toy example 1

Performance of the proposed MCMC approaches for constrained Kriging w.r.t. the one obtained using RSM.





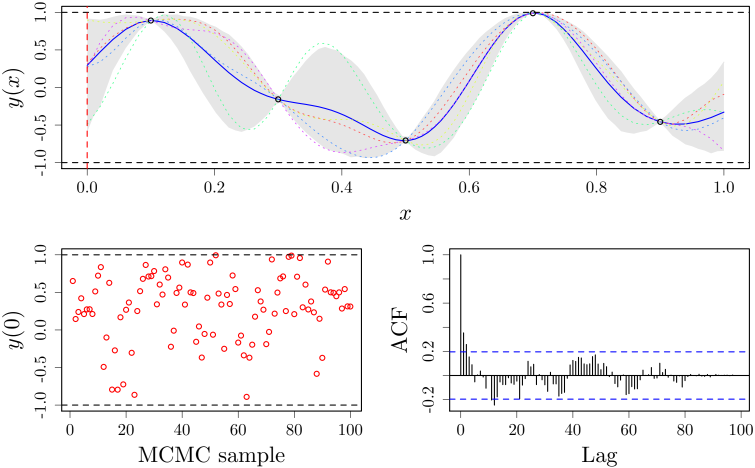


Rejection Sampling from the Mode (RSM) - Gold Standard

Partial results: simulating from the model

Toy example 1

Performance of the proposed MCMC approaches for constrained Kriging w.r.t. the one obtained using RSM.

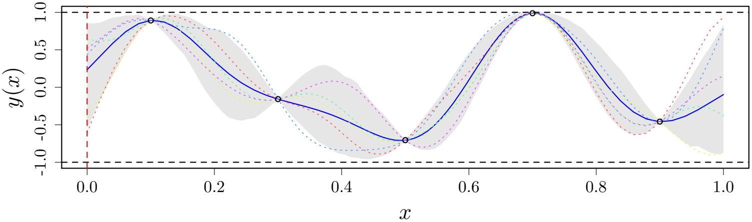


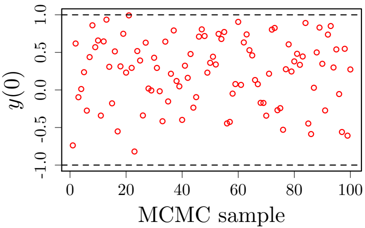
Gibbs sampling with thinning = 100

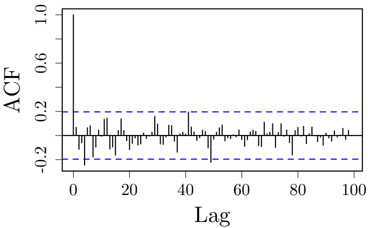
Partial results: simulating from the model

Toy example 1

Performance of the proposed MCMC approaches for constrained Kriging w.r.t. the one obtained using RSM.





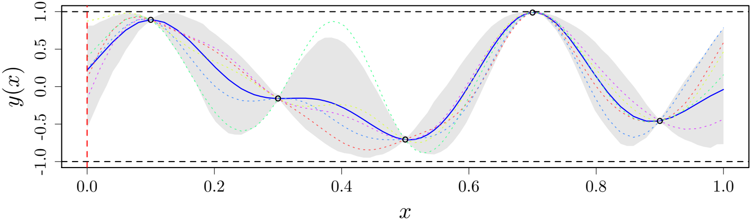


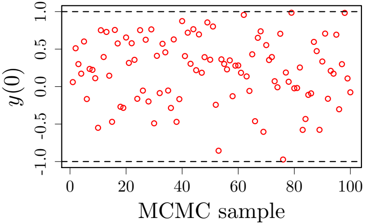
Random walk Metropolis algorithm with $\eta = 1$

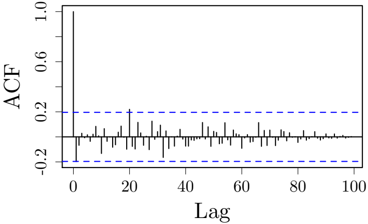
Partial results: simulating from the model

Toy example 1

Performance of the proposed MCMC approaches for constrained Kriging w.r.t. the one obtained using RSM.





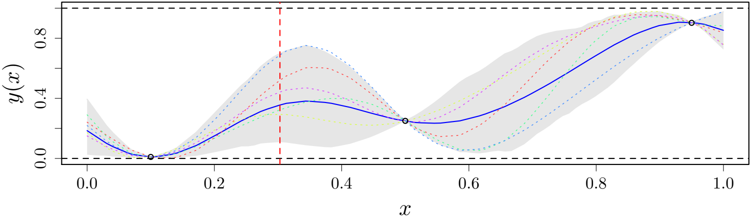


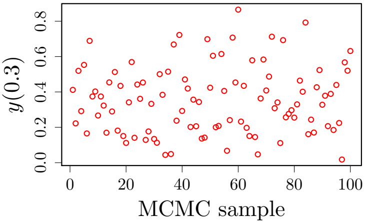
Hamiltonian Monte Carlo (HMC)

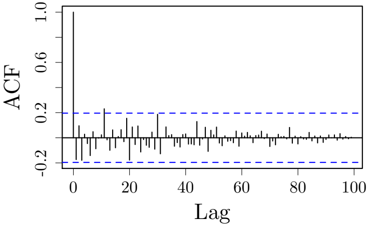
Partial results: simulating from the model

Toy example 2

Performance of the proposed model for Kriging under different types of constraints using HMC. Target function: $y = x^2$.





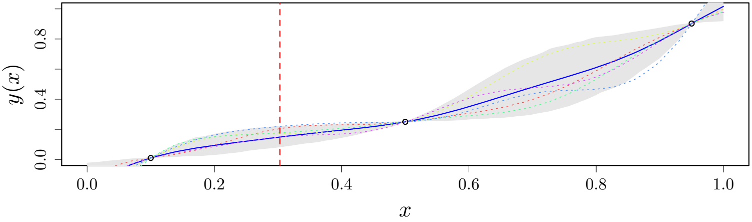


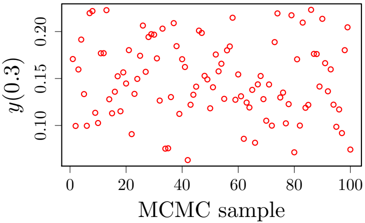
Boundedness

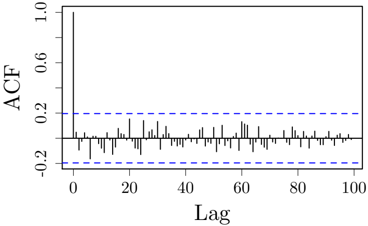
Partial results: simulating from the model

Toy example 2

Performance of the proposed model for Kriging under different types of constraints using HMC. Target function: $y = x^2$.





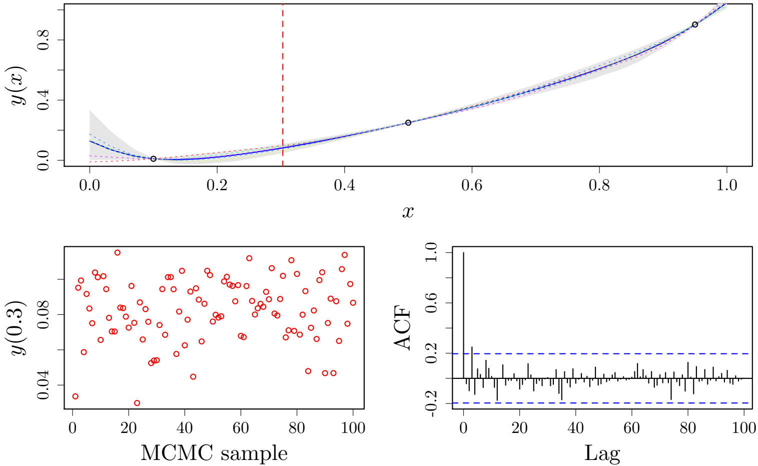


Monotonicity

Partial results: simulating from the model

Toy example 2

Performance of the proposed model for Kriging under different types of constraints using HMC. Target function: $y = x^2$.

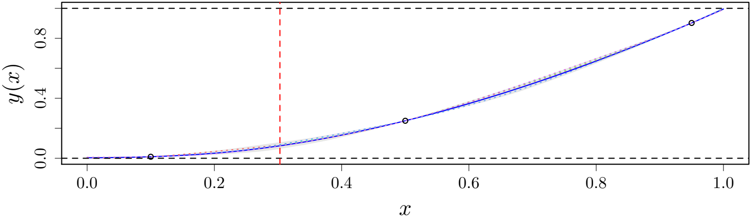


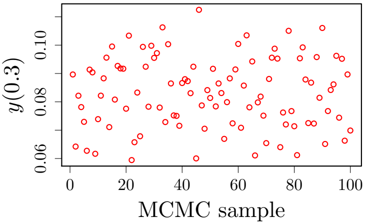
Convexity

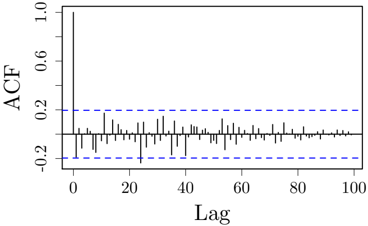
Partial results: simulating from the model

Toy example 2

Performance of the proposed model for Kriging under different types of constraints using HMC. Target function: $y = x^2$.







Boundedness + Monotonicity + Convexity

Table of Contents

Proposal: Maximum Likelihood (ML)

Commonly, the hyperparameters $\boldsymbol{\Theta}$ of a Gaussian process (GP) are estimated by minimizing $\mathcal{L} = -\log p(\mathbf{y}|\boldsymbol{\Theta})$ known as the *negative log likelihood* (NLL). Let $\mathbf{y}|\boldsymbol{\Theta} \sim \mathcal{GP}(\mathbf{0}, \boldsymbol{\Sigma}_{\boldsymbol{\Theta}})$, the NLL follows

$$\mathcal{L} = -\log p(\mathbf{y}|\boldsymbol{\Theta}) = \frac{d}{2}\log(2\pi) + \frac{1}{2}\log|\boldsymbol{\Sigma}_{\boldsymbol{\Theta}}| + \frac{1}{2}\mathbf{y}^\top \boldsymbol{\Sigma}_{\boldsymbol{\Theta}}^{-1}\mathbf{y}, \tag{4}$$

where the covariance $\boldsymbol{\Sigma}_{\boldsymbol{\Theta}}$ depends on the hyperparameters $\boldsymbol{\Theta}$.

Proposal: Maximum Likelihood (ML)

Commonly, the hyperparameters $\boldsymbol{\Theta}$ of a Gaussian process (GP) are estimated by minimizing $\mathcal{L} = -\log p(\mathbf{y}|\boldsymbol{\Theta})$ known as the *negative log likelihood* (NLL). Let $\mathbf{y}|\boldsymbol{\Theta} \sim \mathcal{GP}(\mathbf{0}, \boldsymbol{\Sigma}_{\boldsymbol{\Theta}})$, the NLL follows

$$\mathcal{L} = -\log p(\mathbf{y}|\boldsymbol{\Theta}) = \frac{d}{2}\log(2\pi) + \frac{1}{2}\log|\boldsymbol{\Sigma}_{\boldsymbol{\Theta}}| + \frac{1}{2}\mathbf{y}^\top \boldsymbol{\Sigma}_{\boldsymbol{\Theta}}^{-1}\mathbf{y}, \checkmark \tag{4}$$

where the covariance $\boldsymbol{\Sigma}_{\boldsymbol{\Theta}}$ depends on the hyperparameters $\boldsymbol{\Theta}$.

$\Rightarrow$ Equation (4) can be optimized using standard gradient algorithms.

Proposal: Maximum Likelihood (ML)

Commonly, the hyperparameters $\boldsymbol{\Theta}$ of a Gaussian process (GP) are estimated by minimizing $\mathcal{L} = -\log p(\mathbf{y}|\boldsymbol{\Theta})$ known as the *negative log likelihood* (NLL). Let $\mathbf{y}|\boldsymbol{\Theta} \sim \mathcal{GP}(\mathbf{0}, \boldsymbol{\Sigma}_{\boldsymbol{\Theta}})$, the NLL follows

$$\mathcal{L} = -\log p(\mathbf{y}|\boldsymbol{\Theta}) = \frac{d}{2}\log(2\pi) + \frac{1}{2}\log|\boldsymbol{\Sigma}_{\boldsymbol{\Theta}}| + \frac{1}{2}\mathbf{y}^\top \boldsymbol{\Sigma}_{\boldsymbol{\Theta}}^{-1}\mathbf{y}, \quad ✓ \; ✗ \tag{4}$$

where the covariance $\boldsymbol{\Sigma}_{\boldsymbol{\Theta}}$ depends on the hyperparameters $\boldsymbol{\Theta}$.

$\Rightarrow$ Equation (4) can be optimized using standard gradient algorithms.

$\Rightarrow$ It does not take into account the inequalities $\mathcal{C} : \boldsymbol{l} \leq \boldsymbol{\Lambda}\boldsymbol{\xi} \leq \boldsymbol{u}$.

MINES Saint-Étienne
INSTITUT DE MATHÉMATIQUES DE TOULOUSE

Proposal: Maximum Likelihood (ML)

In order to take into account a prior the fact that $\boldsymbol{\xi}_{\mathcal{C}} : \boldsymbol{l} \leq \boldsymbol{\Lambda}\boldsymbol{\xi} \leq \boldsymbol{u}$, we propose a constrained version of the NLL given by

$$\begin{aligned} \mathcal{L}^* &= -\log p(\mathbf{y}|\boldsymbol{\xi}_{\mathcal{C}}, \boldsymbol{\Theta}) \\ &= -\log \frac{p(\mathbf{y}|\boldsymbol{\Theta}) p(\boldsymbol{\xi}_{\mathcal{C}}|\mathbf{y}, \boldsymbol{\Theta})}{p(\boldsymbol{\xi}_{\mathcal{C}}|\boldsymbol{\Theta})} \\ &= -\log p(\mathbf{y}|\boldsymbol{\Theta}) - \log p(\boldsymbol{\xi}_{\mathcal{C}}|\mathbf{y}, \boldsymbol{\Theta}) + \log p(\boldsymbol{\xi}_{\mathcal{C}}|\boldsymbol{\Theta}). \end{aligned} \tag{5}$$

Proposal: Maximum Likelihood (ML)

In order to take into account a prior the fact that $\boldsymbol{\xi}_{\mathcal{C}} : \boldsymbol{l} \leq \boldsymbol{\Lambda}\boldsymbol{\xi} \leq \boldsymbol{u}$, we propose a constrained version of the NLL given by

$$\begin{aligned} \mathcal{L}^* &= -\log p(\mathbf{y}|\boldsymbol{\xi}_{\mathcal{C}}, \boldsymbol{\Theta}) \\ &= -\log \frac{p(\mathbf{y}|\boldsymbol{\Theta})p(\boldsymbol{\xi}_{\mathcal{C}}|\mathbf{y}, \boldsymbol{\Theta})}{p(\boldsymbol{\xi}_{\mathcal{C}}|\boldsymbol{\Theta})} \\ &= -\log p(\mathbf{y}|\boldsymbol{\Theta}) - \log p(\boldsymbol{\xi}_{\mathcal{C}}|\mathbf{y}, \boldsymbol{\Theta}) + \log p(\boldsymbol{\xi}_{\mathcal{C}}|\boldsymbol{\Theta}). \checkmark \end{aligned} \quad (5)$$

$\Rightarrow$ Good news!! Green terms can be properly estimated using the theory behind Gaussian orthant probabilities (e.g. `mvtnorm` package (Genz, 1992); `TruncatedNormal` package, (Botev, 2017)).

Partial results: estimating the hyperparameters

1-dimensional example

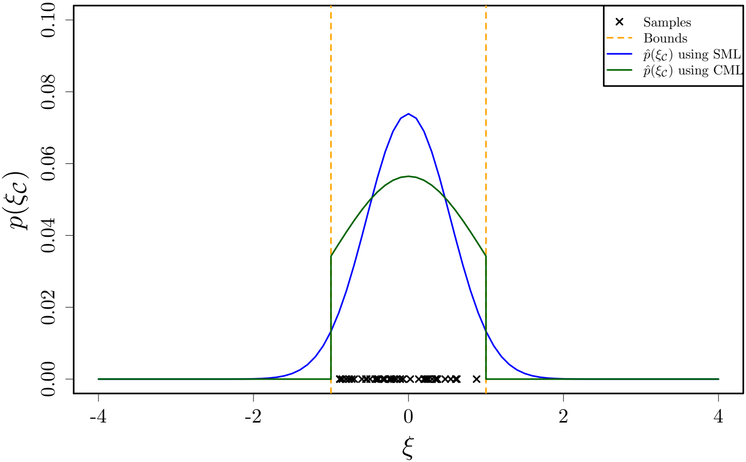
In this example, we estimate $\hat{\sigma}$ given different nb of samples.

- Simulations: $\xi_{\mathcal{C}_i} \sim \mathcal{TN}\left(0, \sigma^2 = 1^2, -1, 1\right)$ for $i = 1, \cdots n$.
- Target function: $y_i = \xi_{\mathcal{C}_i}$.
- We estimate $\hat{\sigma}$ by maximizing each type of likelihood
    - Standard ML (SML): $\hat{\sigma} = \arg\max_\sigma \log p(y|\sigma)$.
    - Constrained ML (CML): $\hat{\sigma} = \arg\max_\sigma \log p(y|\xi_{\mathcal{C}}, \sigma)$.
- We repeat the experiment 100 times.

Partial results: estimating the hyperparameters

1-dimensional example

Estimating $\hat{\sigma}$ given different nb of samples.

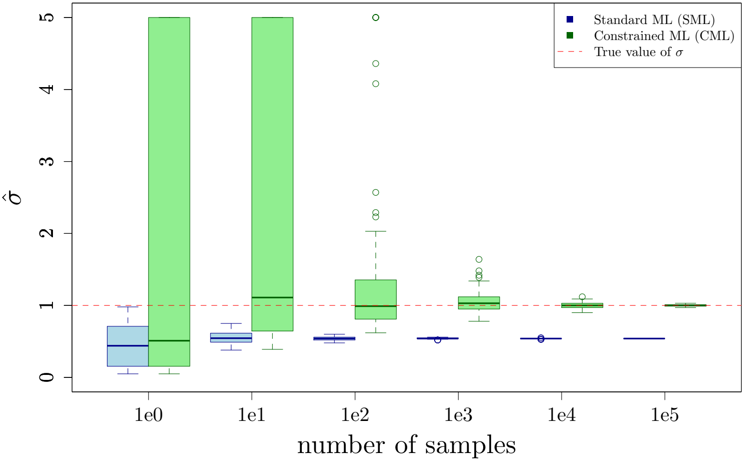


■ Standard ML (SML) ■ Constrained ML (CML)

Partial results: estimating the hyperparameters

1-dimensional example

Estimating $\hat{\sigma}$ given different nb of samples.



Standard ML (SML) Constrained ML (CML)

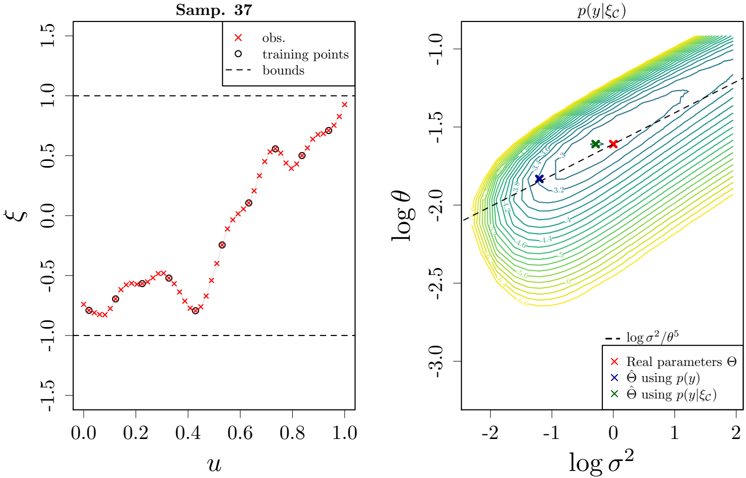
Proposal: estimating the hyperparameters

Multi-dimensional example.
We estimate $\hat{\mathbf{\Theta}} = (\hat{\sigma}^2, \hat{\theta})$ using different GP simulations.

Proposal: estimating the hyperparameters

Multi-dimensional example.

We estimate $\hat{\boldsymbol{\Theta}} = (\hat{\sigma}^2, \hat{\theta})$ using different GP simulations.

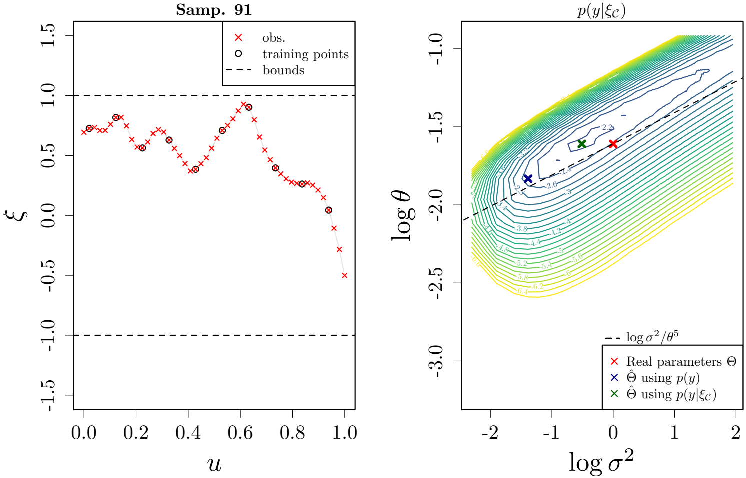


× Standard ML (SML) × Constrained ML (CML)

Proposal: estimating the hyperparameters

Multi-dimensional example.

We estimate $\hat{\boldsymbol{\Theta}} = (\hat{\sigma}^2, \hat{\theta})$ using different GP simulations.

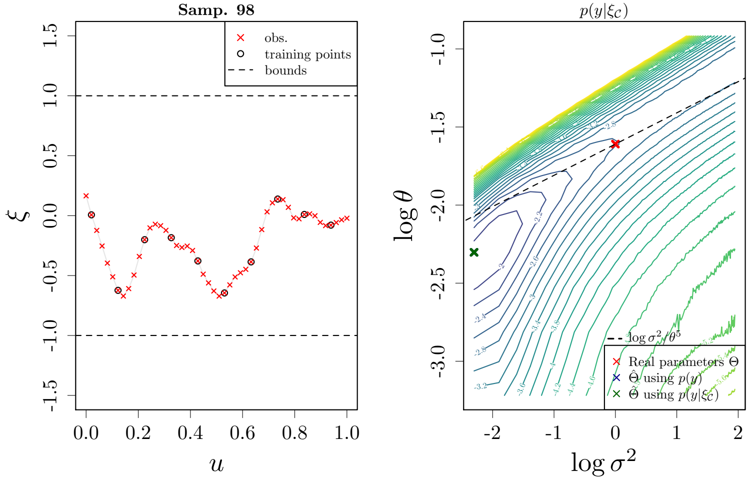


× Standard ML (SML) × Constrained ML (CML)

Proposal: estimating the hyperparameters

Multi-dimensional example.

We estimate $\hat{\boldsymbol{\Theta}} = (\hat{\sigma}^2, \hat{\theta})$ using different GP simulations.

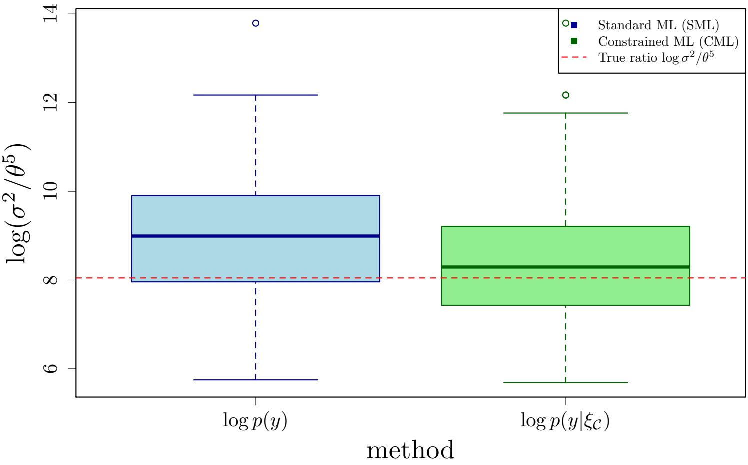


× Standard ML (SML) × Constrained ML (CML)

Proposal: estimating the hyperparameters

Multi-dimensional example.

We estimate $\hat{\mathbf{\Theta}} = (\hat{\sigma}^2, \hat{\theta})$ using different GP simulations.

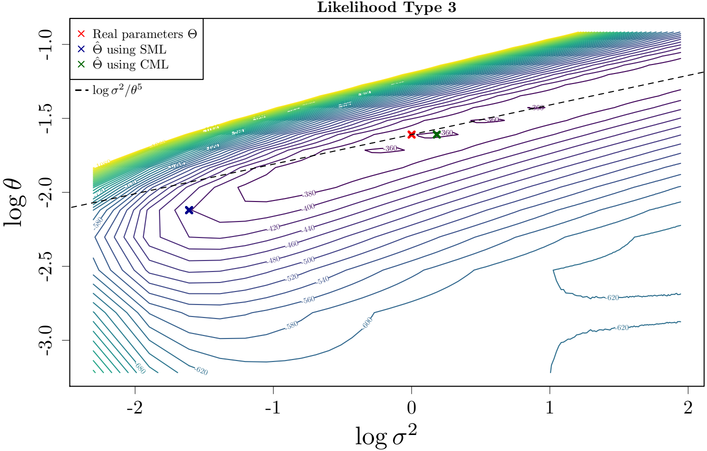


Real value Standard ML (SML) Constrained ML (CML)

Proposal: estimating the hyperparameters

Multi-dimensional example.

We estimate $\hat{\boldsymbol{\Theta}} = (\hat{\sigma}^2, \hat{\theta})$ using different GP simulations.

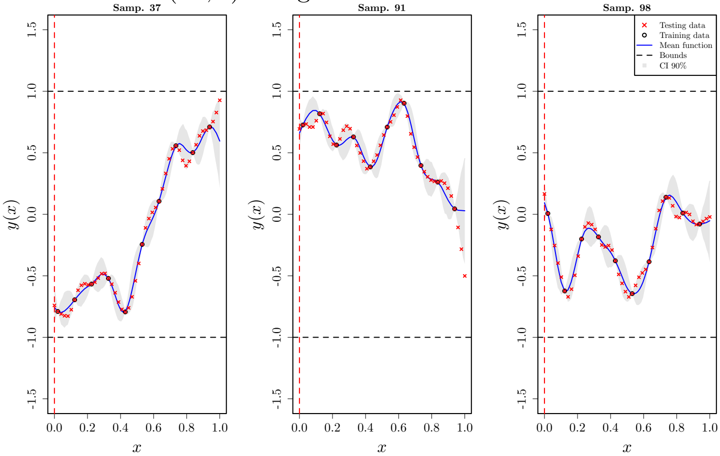


× Real value × Standard ML (SML) × Constrained ML (CML)

Proposal: estimating the hyperparameters

Multi-dimensional example.

We estimate $\hat{\mathbf{\Theta}} = (\hat{\sigma}^2, \hat{\theta})$ using different GP simulations.



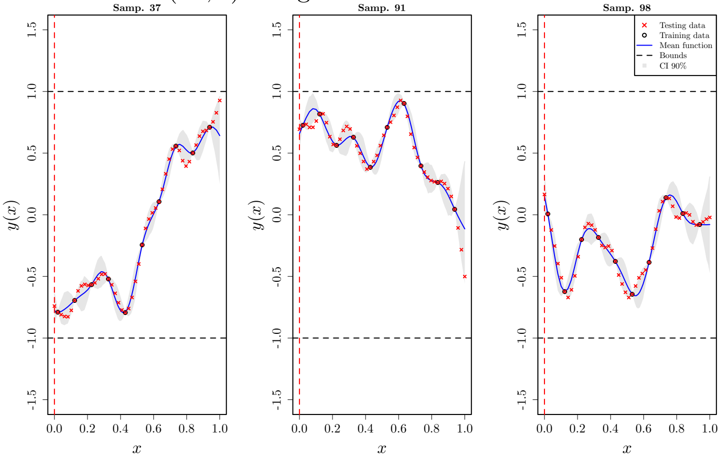
SMSE: (left) 0.00778823 (centre) 0.08241287 (right) 0.0542675
MSLL: (left) -3.935441 (centre) -3.184300 (right) -2.929120

Unconstrained ML.

Proposal: estimating the hyperparameters

Multi-dimensional example.

We estimate $\hat{\mathbf{\Theta}} = (\hat{\sigma}^2, \hat{\theta})$ using different GP simulations.



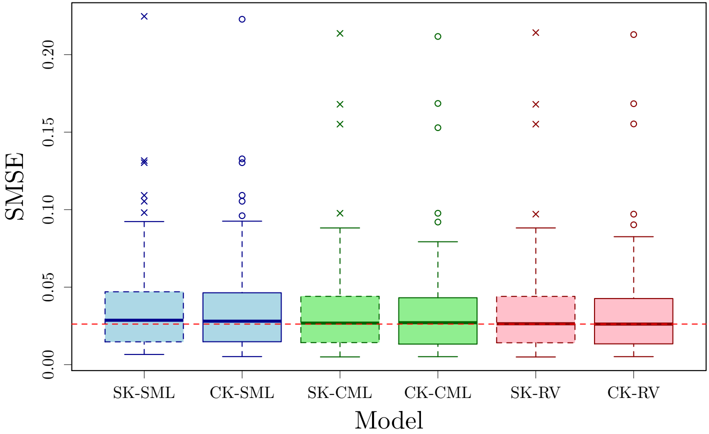
SMSE: (left) 0.0106797 (centre) 0.07928962 (right) 0.0542675
MSLL: (left) -3.960415 (centre) -3.218316 (right) -2.929120

Constrained ML.

Proposal: estimating the hyperparameters

Multi-dimensional example.

We estimate $\hat{\boldsymbol{\Theta}} = (\hat{\sigma}^2, \hat{\theta})$ using different GP simulations.



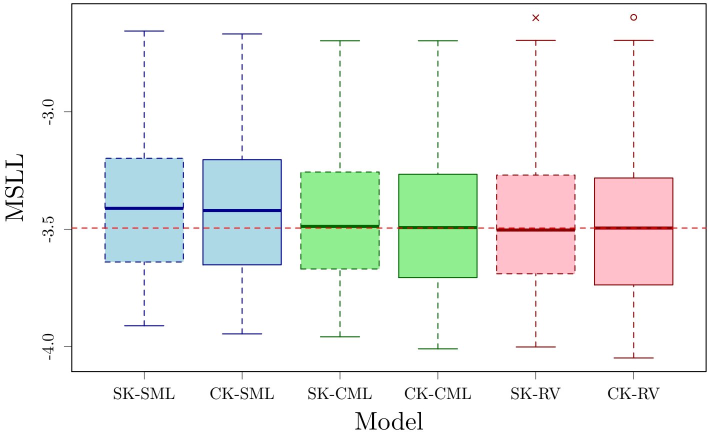
■ Standard ML (SML) ■ Constrained ML (CML) ■ Real $\boldsymbol{\Theta}$

× Standard Kriging (SK) ∘ Constrained Kriging (CK)

Proposal: estimating the hyperparameters

Multi-dimensional example.

We estimate $\hat{\mathbf{\Theta}} = (\hat{\sigma}^2, \hat{\theta})$ using different GP simulations.



■ Standard ML (SML) ■ Constrained ML (CML) ■ Real $\mathbf{\Theta}$

$\times$ Standard Kriging (SK) $\circ$ Constrained Kriging (CK)

Table of Contents

Conclusions and Future Works

Conclusions

- We further investigated the approach proposed in (Maatouk and Bay, 2016): now it works for any linear set of inequality constraints.
- We implemented several simulation methods based on MCMC approaches.
- We investigated a proper likelihood for hyperparameters estimation.
- We implemented the codes in R (they are almost an R package).

Conclusions and Future Works

Future works

- To implement a gradient-based method to estimate automatically the hyperparameters of the model.
- To investigate theoretical estimation properties from the constrained likelihood.
- To evaluate the proposed approach with real problems.
- To build the R package.
- To extend this approach when the input space is multi-dimensional, i.e. $\mathbf{x} \in [0\ 1]^d$.

Table of Contents

Table of Contents

MINES Saint-Étienne

INSTITUT DE MATHÉMATIQUES DE TOULOUSE

References I

Altmann, Y., McLaughlin, S., and Dobigeon, N. (2014). Sampling from a multivariate gaussian distribution truncated on a simplex: A review. In *2014 IEEE Workshop on Statistical Signal Processing (SSP)*, pages 113–116.

Azzimonti, D. and Ginsbourger, D. (2016). Estimating orthant probabilities of high dimensional Gaussian vectors with an application to set estimation. *ArXiv e-prints*.

Bachoc, F. (2013). Cross validation and maximum likelihood estimations of hyper-parameters of Gaussian processes with model misspecification. *Computational Statistics & Data Analysis*, 66:55 – 69.

Betancourt, M. (2017). A Conceptual Introduction to Hamiltonian Monte Carlo. *ArXiv e-prints*.

Bishop, C. M. (2007). *Pattern Recognition And Machine Learning (Information Science And Statistics)*. Springer.

Botev, Z. I. (2017). The normal law under linear restrictions: simulation and estimation via minimax tilting. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 79(1):125–148.

Brooks, S., Gelman, A., Jones, G., and Meng, X. (2011). *Handbook of Markov Chain Monte Carlo*. Chapman & Hall/CRC Handbooks of Modern Statistical Methods. CRC Press.

Cramér, H. and Leadbetter, M. (1967). *Stationary and related stochastic processes: sample function properties and their applications*. Wiley series in probability and mathematical statistics. Tracts on probability and statistics. Wiley.

Da Veiga, S. and Marrel, A. (2012). Gaussian process modeling with inequality constraints. *Annales de la faculté des sciences de Toulouse Mathématiques*, 21(3):529–555.

References II

Duane, S., Kennedy, A. D., Pendleton, B. J., and Roweth, D. (1987). Hybrid monte carlo. *Physics letters B*, 195(2):216–222.

Genton, M. G. (2001). Classes of kernels for machine learning: A statistics perspective. *Journal of Machine Learning Research*, 2:299–312.

Genz, A. (1992). Numerical computation of multivariate normal probabilities. *Journal of Computational and Graphical Statistics*, 1:141–150.

Genz, A., Bretz, F., Miwa, T., and Mi, X. (2017). mvtnorm: Multivariate normal and t distributions. `https://cran.r-project.org/web/packages/mvtnorm/mvtnorm.pdf`. [Online; accessed 04-April-2017].

Golchi, S., Bingham, D. R., Chipman, H., and Campbell, D. A. (2015). Monotone emulation of computer experiments. *SIAM/ASA Journal on Uncertainty Quantification*, 3(1):370–392.

Gramacy, R. B. and Apley, D. W. (2015). Local Gaussian process approximation for large computer experiments. *Journal of Computational and Graphical Statistics*, 24(2):561–578.

L'Ecuyer, P., Blanchet, J. H., Tuffin, B., and Glynn, P. W. (2010). Asymptotic robustness of estimators in rare-event simulation. *ACM Trans. Model. Comput. Simul.*, 20(1):6:1–6:41.

Liu, J. S. (2008). *Monte Carlo Strategies in Scientific Computing*. Springer Publishing Company, Incorporated.

Maatouk, H. and Bay, X. (2016). Gaussian process emulators for computer experiments with inequality constraints. *ArXiv e-prints*.

Maatouk, H. and Bay, X. (2016). *A New Rejection Sampling Method for Truncated Multivariate Gaussian Random Variables Restricted to Convex Sets*, pages 521–530. Springer International Publishing, Cham.

References III

Maatouk, H., Roustant, O., and Richet, Y. (2015). Cross-validation estimations of hyper-parameters of Gaussian processes with inequality constraints. *Procedia Environmental Sciences*, 27:38 – 44.

Murphy, K. P. (2012). *Machine Learning: A Probabilistic Perspective (Adaptive Computation And Machine Learning Series)*. The MIT Press.

Neal, R. M. (1996). *Bayesian Learning for Neural Networks*. Springer-Verlag New York, Inc., Secaucus, NJ, USA.

Neal, R. M. (2012). MCMC using Hamiltonian dynamics. *ArXiv e-prints*.

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)*. The MIT Press.

Ripley, B. D. (1987). *Stochastic Simulation*. John Wiley & Sons, Inc., New York, NY, USA.

Rogozhnikov, A. (2016). Brilliantly wrong thoughts on science and programming: Hamiltonian Monte Carlo explained. `http://arogozhnikov.github.io/2016/12/19/markov_chain_monte_carlo.html`. [Online; accessed 04-April-2017].

Stein, M. (1999). *Interpolation of Spatial Data: Some Theory for Kriging*. Springer Series in Statistics. Springer New York.

Taylor, J. and Benjamini, Y. (2017). RestrictedMVN: multivariate normal restricted by affine constraints. `https://cran.r-project.org/web/packages/restrictedMVN/index.html`. [Online; accessed 02-February-2017].