Prediction method for cyclone-induced waves using Gaussian Process models

Chifaa Dahik, Olivier Roustant, Stéphane Puechmorel, Fabrice Gamboa

Mathematics Institute of Toulouse

March, 9 2023. IMT PhD seminar

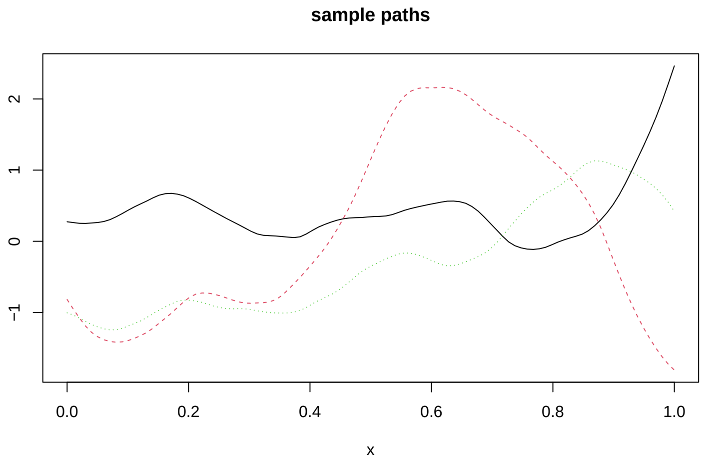
What is a Gaussian random process?

Gaussian random process

- A random process (or stochastic process) is a family of random variables , indexed by time $Z(t)_{t \in T}$ or space $Z(x)_{x \in X}$.
- A gaussian process is a random process such that every finite collection of those random variables has a multivariate normal distribution.

$$\rightarrow \mathcal{N}(\mu, \Sigma),$$
$$\rightarrow \mu = E[X] = (E[X_1], \ldots, E[X_n])^T,$$
$$\rightarrow \Sigma_{i,j} = E[(X_i - \mu_i)(X_j - \mu_j)] = Cov[X_i, X_j].$$

Example of a gaussian random process: figure



- Each curve represents 100 values: 1 representation for each $Z(x_i)$, $i = 1, \ldots, 100$.

Example: ingredients

Ingredients for a gaussian random process $Z(x)_{x \in X}$:

- A covariance function on $\mathbb{R} \times \mathbb{R}$: Matérn 5/2

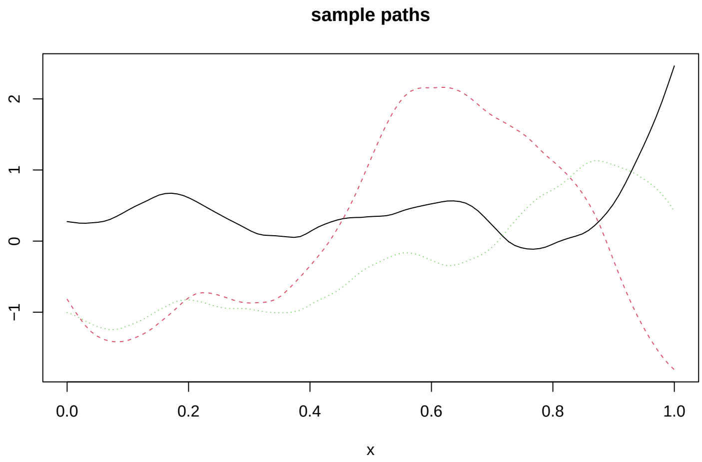
$$k(x, y) = \sigma^2 (1 + \frac{\sqrt{5}|x-y|}{\theta} + \frac{5|x-y|^2}{3\theta^2}) e^{-\frac{\sqrt{5}|x-y|}{\theta}}.$$

- Choice of covariance parameters: $\sigma = 1$, $\theta = 0.2$.
- A regular sequence $X = [x_1, \ldots, x_i, \ldots, x_n]$ of $n = 100$ points on $[0, 1]$.
- Calculate $k(x_i, x_j)$ $i, j = 1, \ldots, n$ to get a covariance matrix K with dimension $n \times n$.
- Generate 3 multivariate normal random samples.

Example: code

```r
mat5_2Kern <- function(x, y, param){
  sigma <- param[1]
  theta <- param[2]
  d <- abs(outer(x, y, '-')) * sqrt(5) / theta
  kern <- sigma^2*(1 + d + d^2 / 3)*exp(- d)
  return(kern)
}

## Simulating sample-paths  ##
n <- 100

x <- seq(0, 1, length.out = n) # regular sequence

param <- c(1, 0.2) # covariance parameters

K1 <- mat5_2Kern(x, x, param) # covariance matrix

## simulating some samples using the "mvrnorm" function
samples <- mvrnorm(n = 3, mu = rep(0, n), Sigma = K1)

# ?matplot # a function to plot the samples. The samples are indexed by columns
matplot(x, t(samples), type = "l",
        main = "sample paths", ylab = "")
```

Example: figure



What is a Gaussian random process regression?

Gaussian random process regression (Kriging/metamodeling)

- $x = (x_1, \ldots, x_d) \in D \subset \mathbb{R}^d$.
- Environmental simulator $f(.)$, $y = f(x)$.
- For the prediction, y is assumed to be a realization of a Gaussian process $(Y_x)_{x \in D}$ with a mean μ and a covariance function $k(.)$:

$$Y(x) = \mu(x) + Z(x). \quad (1)$$

$\rightarrow \mu : x \in D \rightarrow \mu(x) \in \mathbb{R}$ called trend function.
$\rightarrow Z$ is a centered gaussian process (with mean 0).
$\rightarrow k(x, x') = cov(Y(x), Y(x'))$.

Gaussian random process regression (Kriging/metamodeling)

- $X = (x^{(1)}, \ldots, x^{(n)})$ the points where y has already been evaluated ,
- $y = (y^{(1)}, \ldots, y^{(n)})^T$ the corresponding outputs.
- The prediction at a new observation x is the mean kriging

$$\begin{aligned}
&\hat{Y}(x) = E[Y(x) | Y(x^{(1)}) = y^{(1)}, \ldots, Y(x^{(n)}) = y^{(n)}] \\
&= \mu(x) + c(x)^T K^{-1}(y - \mu), \\
&K = (k(x^{(i)}, x^{(j)}))_{1 \leq i,j \leq n}, \\
&c(x) = (k(x, x^{(i)}))_{1 \leq i \leq n}.
\end{aligned}$$

- Best Linear Unbiased Predictor: minimizes

$$\text{MSE} := E[(Y(x) - \hat{Y}(x))^2]$$

under $E[\hat{Y}(x)] = E[Y(x)]$.

Gaussian random process regression (Kriging/metamodeling)

- The error estimate

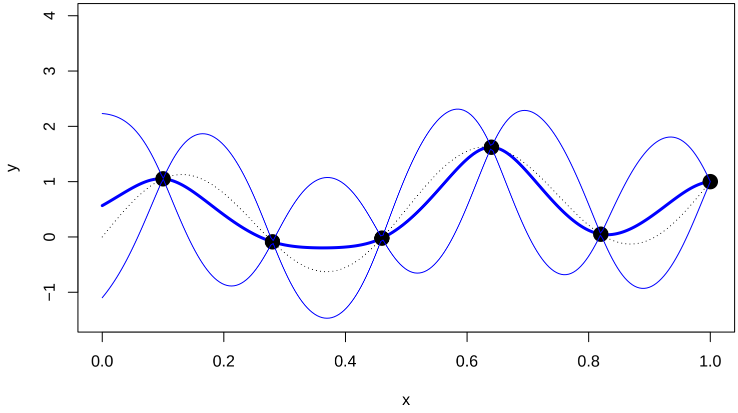
$$\hat{\sigma}^2(x) = k(x,x) - c(x)^T K^{-1} c(x).$$

- The 95% prediction interval is

$$[\hat{Y}(x) - 1.96\hat{\sigma}(x), \hat{Y}(x) + 1.96\hat{\sigma}(x)].$$

Example

Approximate the test function $f : x \in [0, 1] \rightarrow x + sin(4x)$ with Gaussian process regression.



Case study

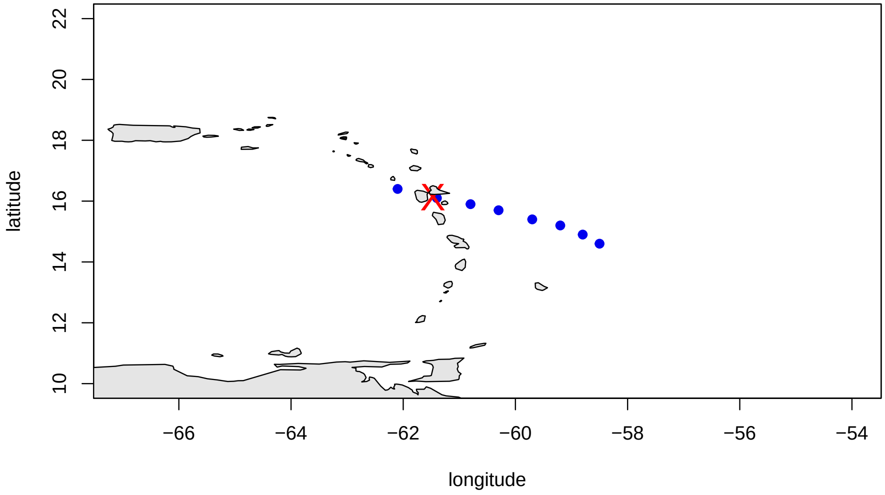
Context of the postdoc

Consortium's interest: Metamodels on non-Euclidean spaces- Application given by the partner BRGM.

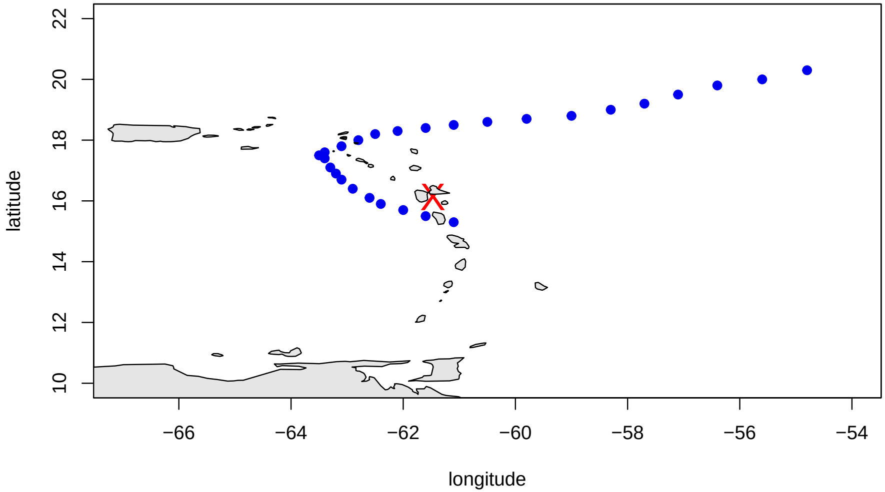
Data description

- A cyclone trajectory is a set of information in different positions of the cyclone, the duration between two consecutive positions is 3 hours.
- Each cyclone has a different number of positions.
- Each position has the following information: the longitude ϕ, latitude λ, the wind speed U, and the radius R from the cyclone eye.
- The goal is to predict the wave height H_S at a given position pt close to the coast of Guadeloupe island.

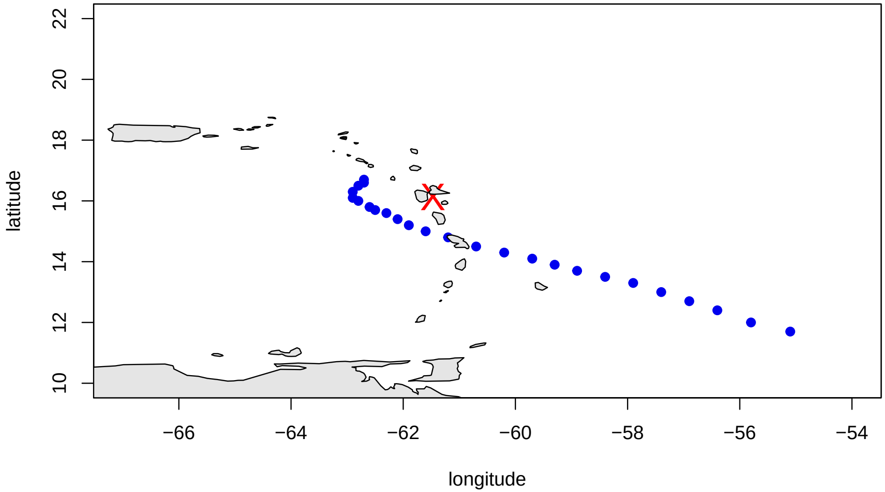
Examples of cyclone trajectory



Examples of cyclone trajectory



Examples of cyclone trajectory



Goals of the postdoc

Advisors said: Chifaa- You must do this

- Construction of graphs capturing the main characteristics of the trajectory,

And this

- Construction of adapted kernels on graphs, e.g. based on Laplacians,

And this

- Application to the cyclone case study.

Chifaa said: Ok!

Challenges

- Representation of the cyclone: include physical and topological features, each cyclone different size!
- Propose a kernel function for these complex objects.

Graph representation of the cyclones

Graph definition

- $G = (V, E)$, with $V = \{v_1, \ldots, v_n\}$ the set of nodes, and with edges $E = \{w_{ij}\}_{ij}$. The graph Laplacian L is a $n \times n$ matrix given by

$$L_{i,j} = \begin{cases} -w_{i,j} & \text{if } \{v_i, v_j\} \in \text{E}, \\ \sum_{k;\{v_i,v_k\}\in E} w_{ik} & \text{if i=j}, \\ 0 & \text{otherwise.} \end{cases}$$

- Each node v_i is only connected to v_{i-1} and v_{i+1} .
- The edge weights are chosen to be the geodesic distances between the nodes.
- The Laplacian matrix contains information about the connection between the nodes of a graph, but other information are important to consider.

Feature addition

- Thus we associate to each node a vector of features that can contain any topological or physical information.
- In our case, we consider to associate each vertex v_i to a vector of features $\psi = (U, R, d_{pt}, \alpha_l, \alpha_0)$.
- α_l is the local angle of curvature at the node v, which is the angle $v_{i-1}\widehat{v_i v_i}_{+1}$.
- α_0 is the angle on the node with a segment of reference $(\widehat{\overrightarrow{v_i v_{i+1}}, \overrightarrow{Ox}})$.

Useful formulas for the graph representation

For $P_1 = (\phi_1, \lambda_1)$, $P_2 = (\phi_2, \lambda_2)$,

$$d(P_1, P_2) =$$
$$R_{\text{terre}}(acos(sin(\lambda_1) * sin(\lambda_2) + cos(\lambda_1) * cos(\lambda_2) * cos(\phi_1 - \phi_2))).$$

For the angles computation, spherical coordinates → cartesian coordinates

$$P = (cos\lambda cos\phi, cos\lambda sin\phi, sin\lambda),$$

$$cos\widehat{P_1 P_2 P3} = \frac{\overrightarrow{P_2 P_1} \bullet \overrightarrow{P_2 P_3}}{\|\overrightarrow{P_2 P_1}\|\|\overrightarrow{P_2 P_3}\|}.$$

For 3 points A, B and C three points on earth, O center of the earth,

$$cos\widehat{BAC} = \frac{cos\widehat{BOC} - cos\widehat{AOC}cos\widehat{AOB}}{sin\widehat{AOC}sin\widehat{AOB}}.$$

Object transformation

$$k(\text{Cyclone } 1, \text{Cyclone } 2) = ?$$

with Cyclone= set of (?) nodes with 5 features over each node.

- Transformation from the vertex space variables to the feature space variables.
- $U_{ij} = \psi_i(v_j)$, with $i \in \{1, \ldots, 5\}$, and $j \in \{1, \ldots, N\}$.
- $X \sim \mathcal{N}(0, L^{-1}) \implies Y = U^T X \sim \mathcal{N}(0, UL^{-1}U^T)$.

$$k(\text{Cyclone } 1, \text{Cyclone } 2) = k(U_1 L_1^{-1} U_1^T, U_2 L_2^{-1} U_2^T).$$

Examples of kernels

- Kernels between positive semi-definite matrices

$$k(A, B) = \sigma^2 e^{-\lambda d_{LE}(A,B)},$$

where

$$d_{LE}(A, B) = \|logA - logB\|_F.$$

$$\|A\|_F = \sqrt{tr(A^T A)}.$$

- Kernels between distributions, etc.

To be continued . . .

Back-up slides

MVN distribution

How to generate random multivariate normal distribution?
$\Sigma = T^T T$, T upper triangular.
If $X \sim \mathcal{N}(0, I_n)$, then $T^T E \sim \mathcal{N}(0, \Sigma)$.

Example: code

```r
fun <- function(x) {
  x + sin(4*pi*x)
}
X <- seq(0.1, 1, length.out = 6)
Y <- fun(X)

## 5. Conditional mean and kernel
condMean <- function(x, X, Y, kern, param) {
  K <- kern(X, X, param)
  Kinv <- solve(K)
  kxX <- kern(x, X, param)
  return(kxX %*% Kinv %*% Y)
}

condCov <- function(x, X, kern, param) {
  K <- kern(X, X, param)
  Kinv <- solve(K)
  kxX <- kern(x, X, param)
  kxx <- kern(x, x, param)
  return(kxx - kxX %*% Kinv %*% t(kxX))
}
```

Example: code

```
x <- seq(0, 1, length.out = 250) # test points
param <- c(1, 0.1) # covariance params
kern <- mat5_2Kern   # default

mu <- condMean(x, X, Y, kern, param) # cond. mean
Sigma <- condCov(x, X, kern, param) # cond. covariance

varSigma <- diag(Sigma)
varSigma <- pmax(varSigma, 0)  # to avoid numerical negative values
plot(x, fun(x), type = "l", ylim = c(-1.5, 4), ylab = "y",
     lty = "dotted")
lines(x, mu, type = "l", col = "blue", lwd = 3)
points(X, Y, col = "black", pch = 20, cex = 3)
lines(x, mu + 1.96*sqrt(varSigma), col = "blue")
lines(x, mu - 1.96*sqrt(varSigma), col = "blue")
```