An Introduction to Uncertainty Quantification

Josselin Garnier

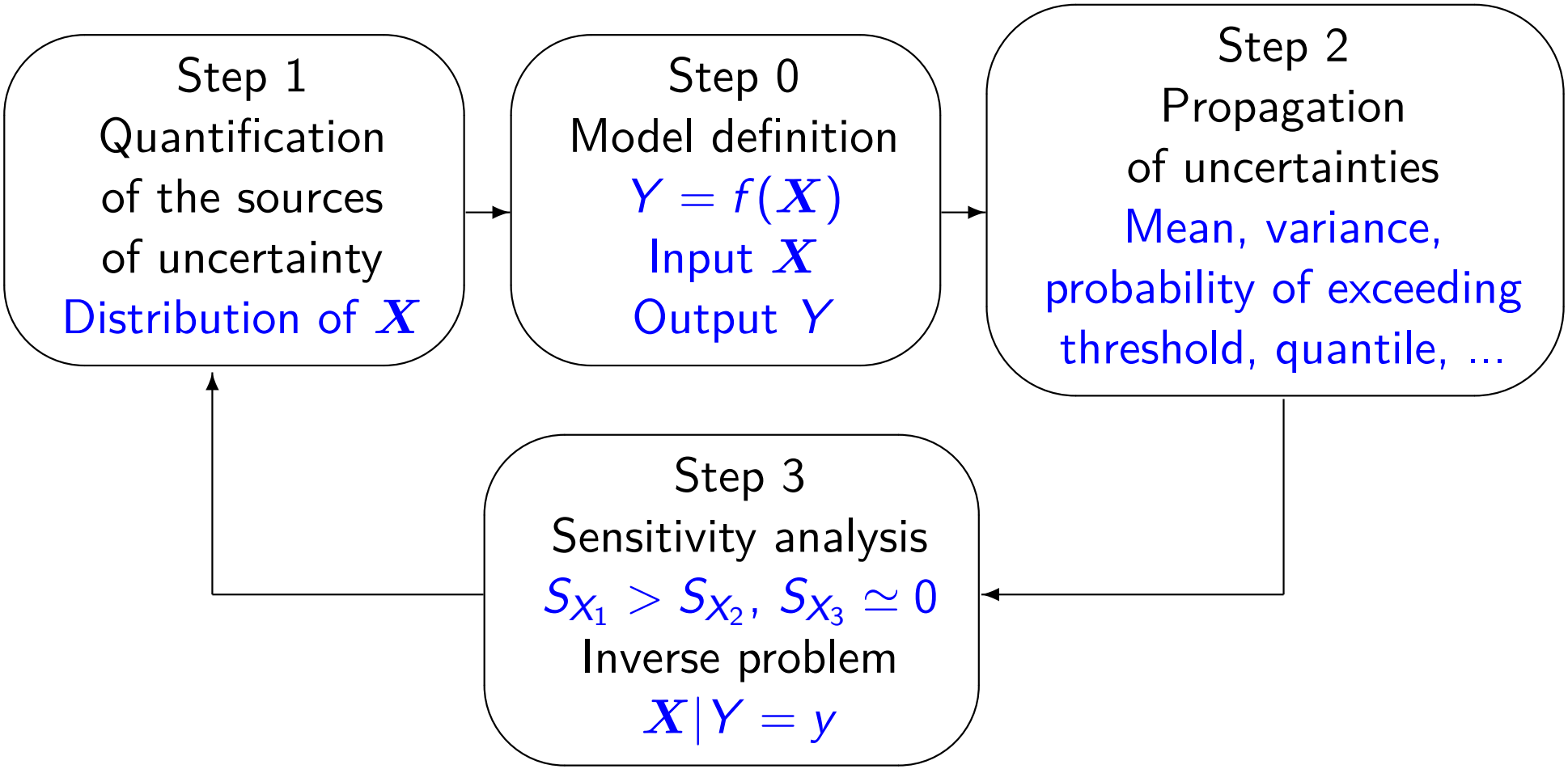
Ecole polytechnique

IHP - November 12, 2025

Why uncertainty quantification in climate science?

- Climate science deals with complex, multi-scale, nonlinear systems.
- Predictions are uncertain due to:
  - incomplete knowledge of parameters and processes,
  - variability in natural systems,
  - limitations of computational models.
- Uncertainty quantification provides tools to characterize, propagate (and reduce) these uncertainties.
- Goal: quantify confidence in model outputs and support decision-making.

Uncertainty quantification pipeline



$\hookrightarrow$ Use of surrogate models for Step 2 and Step 3 when f is costly.

Step 1. Sources of uncertainty

- Two types of "uncertain" inputs:
  - Stochastic variables (irreducible):
    - These variables have natural variability resulting from random phenomena.
    - Typically, an external (solar) forcing term.
  - Epistemic variables (reducible):
    - These variables have a value, but it is unknown to us due to a lack of knowledge.
    - Typically, a constant in a physical/biological law.
- Input variables are treated as random variables.

Their distribution can be determined by expert judgments, prior information, data.

Sources of uncertainty in climate models

- **Initial conditions** $\rightarrow$ chaotic sensitivity.
- **Parametric uncertainty** $\rightarrow$ e.g., cloud albedo.
- **Structural uncertainty** $\rightarrow$ incomplete, approximate, or stochastic equations (stochastic climate modeling†).
- **Forcing uncertainty** $\rightarrow$ future emissions, solar forcing.
- **Observation uncertainty** $\rightarrow$ sparse or noisy data.

†: the fast fluctuations in a climate model can be represented by a stochastic process in an effective climate model which only simulates the slow effects (2021 Nobel prize for S. Manabe, K. Hasselmann, G. Parisi).

Step 2. Propagation of uncertainties

- Context: computer code (or experiment) modeled by

$$Y = f(\boldsymbol{X})$$

with $Y =$ output variable
$\boldsymbol{X} = (X_j)_{j=1}^d$ input variables, with given distribution
$f =$ black box

- Goal: estimation of an expectation (quantity of interest)

$$\mathbb{E}[\psi(Y)]$$

with an error bar and a small number of simulations/experiments.

- Examples (when Y is real-valued):
- $\psi(y) = y \rightarrow$ mean of Y, i.e. $\mathbb{E}[Y]$
- $\psi(y) = y^2 \rightarrow$ variance of Y, i.e. $\mathrm{Var}(Y) = \mathbb{E}[Y^2] - \mathbb{E}[Y]^2$
- $\psi(y) = \mathbf{1}_{[a,\infty)}(y) \rightarrow$ probability to exceed the threshold a, i.e. $\mathbb{P}(Y \geq a)$

Quadratic sum method

- The first method developed by engineers ("sandwich method").
- Goal: For real-valued output, estimate $\mathbb{E}[Y]$ and $\mathrm{Var}(Y)$ from $\boldsymbol{\mu} = (\mathbb{E}[X_j])_{j=1}^d$ and $\mathbf{C} = (\mathrm{Cov}(X_i, X_j))_{i,j=1}^d$.
- Assuming C_{jj} are small and f is smooth:

$$\mathbb{E}[Y] \simeq f(\boldsymbol{\mu}), \qquad \mathrm{Var}(Y) \simeq \nabla f(\boldsymbol{\mu})^T \mathbf{C} \nabla f(\boldsymbol{\mu}).$$

Proof: If $f \in \mathcal{C}^2$, then

$$f(\boldsymbol{x}) = f(\boldsymbol{\mu}) + \nabla f(\boldsymbol{\mu}) \cdot (\boldsymbol{x} - \boldsymbol{\mu}) + O(|\boldsymbol{x} - \boldsymbol{\mu}|^2).$$

□

- We only need to know f and ∇f at $\boldsymbol{\mu}$.
- Fast ($d + 1$ calls to the code, or even 2 if an adjoint code is available).
- Suitable for small variations in input parameters and a smooth model (which can be linearized).

Quadrature methods

- When the random vector $\boldsymbol{X}$ has a pdf $p(\boldsymbol{x})$, the quantity of interest $I = \mathbb{E}[\psi(Y)]$, $Y = f(\boldsymbol{X})$, is a d-dimensional integral:

$$I = \int_{\mathbb{R}^d} \psi(f(\boldsymbol{x}))p(\boldsymbol{x})d\boldsymbol{x}$$

- Quadrature or cubature methods:

$$I_n = \sum_{i=1}^{n} w_i \psi(f(\boldsymbol{x}_i))$$

for well-chosen nodes $(\boldsymbol{x}_i)_{i=1}^n$ and weights $(w_i)_{i=1}^n$.
They require:
- regularity conditions on $\boldsymbol{x} \to \psi(f(\boldsymbol{x}))$,
- a small dimension d (even with sparse grids of the Smolyak type),
- many calls to the code.

Monte Carlo methods

- The quantity of interest $I = \mathbb{E}[\psi(Y)]$ is an expectation.
- Monte Carlo method:

$$\hat{I}_n = \frac{1}{n}\sum_{i=1}^{n} \psi(f(\boldsymbol{X}^{(i)}))$$

where the $(\boldsymbol{X}^{(i)})_{i=1}^n$ are n i.i.d. copies of $\boldsymbol{X}$.

- Unbiased estimation:

$$\mathbb{E}[\hat{I}_n] = I$$

with the mean squared error:

$$\mathbb{E}\left[(\hat{I}_n - I)^2\right]^{1/2} = \frac{1}{\sqrt{n}} \mathrm{Var}(\psi(f(\boldsymbol{X})))^{1/2}$$

- Advantages:

- no regularity required on f, ψ,
- convergence rate independent of dimension d (but slow $O(n^{-1/2})$),
- confidence intervals.

Advanced Monte Carlo methods

$$\mathbb{E}\left[(\hat{I}_n - I)^2\right]^{1/2} = \frac{1}{\sqrt{n}} \mathrm{Var}(\psi(f(\boldsymbol{X})))^{1/2}$$

- **Variance reduction techniques**

We seek to reduce the "constant" while keeping $1/\sqrt{n}$.
Various methods are possible (importance sampling, control variate, ...).
Particularly useful for estimating probabilities of rare events.

- **Quasi Monte Carlo**

Low discrepancy sequences : The sample is drawn in a less random manner than Monte Carlo (mainly to fill in the gaps).
This technique:
- reduces variance if $\boldsymbol{x} \to \psi(f(\boldsymbol{x}))$ has some regularity and/or monotonicity; we can expect an error of $C_d(\log n)^d/n$,
- works in dimension d that is not too large,
- represents an intermediate between Monte Carlo and quadrature.
- does not give directly a confidence interval.

Propagation of uncertainties by surrogate model

- We replace f with a surrogate model (reduced model, response surface) and apply one of the above techniques.

$\rightarrow$ We can make many calls to the surrogate model.

$\rightarrow$ The choice of surrogate model is critical.

$\rightarrow$ The quadratic sum method is a special case (affine surrogate model).

- Construction of the surrogate model:
  - The surrogate model must be constructed by calling the code f at a limited number of points.
  - The choice of surrogate model depends on the quantity of interest.
  - Error control is not simple but possible.
- Surrogate models:
  - Linear regression
  - Gaussian Process (GP) regression
  - Polynomial Chaos Expansion (PCE)
  - Neural Network (NN)

Step 3. Sensitivity analysis

- Context: computer code (or experiment) modeled by

$$Y = f(\boldsymbol{X})$$

with $Y =$ output variable,
$\boldsymbol{X} = (X_j)_{j=1}^d$ input variables, with given distribution,
$f =$ black box.
d can be large, f can be costly.

- Objective: explain the variability of the output Y as a function of the X_j's.

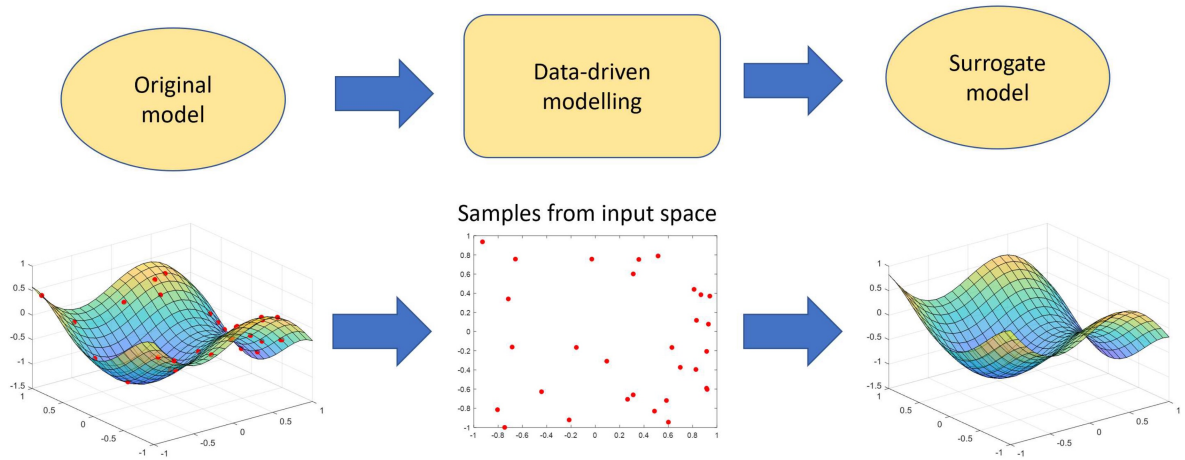
- Purpose of sensitivity analysis:
- qualitative: Identify and rank influential parameters (screening).
- quantitative: Quantify the impact of each input variable on output uncertainty, for instance, determine the proportion of the **variance** of the output Y due to an input variable X_j (or a subset of input variables).

- Useful applications: Reduce dimensionality for surrogate construction or calibration.

Surrogate modeling

Surrogate modeling

- Surrogate modeling is an application of supervised machine learning.
- The objective is to build a metamodel/surrogate model/surface response of a complex process $\boldsymbol{x} \mapsto y$ based on training data $\{(\boldsymbol{x}^{(i)}, y^{(i)})\}_{i=1}^{n}$.



- Popular ML approaches, such as linear or polynomial regressions, support vector machines, Gaussian process regression, and neural networks are commonly used as metamodels.
- Special focus on methods that can work with limited data and which allow for uncertainty quantification.

Surrogate modeling with uncertainty quantification

1. Construction of a surrogate model by linear regression.
2. Gaussian process regression: improve linear regression by the introduction of a model error.
3. Polynomial chaos expansion: improve linear regression by an appropriate choice of the basis functions.
4. Kalman filter: assimilate information from noisy measurements and approximate models to produce accurate estimates (cf E Blayo's talk).

References

- R. B. Gramacy, *Surrogates: Gaussian Process Modeling, Design, and Optimization for the Applied Sciences*, Chapman and Hall, 2020.
- T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*, Springer, 2009.
- C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*, MIT Press, 2006.
- T. Sullivan, *Introduction to Uncertainty Quantification*, Springer, 2015.

Linear regression

Linear regression problem

- The *true output* is:

$$f(\boldsymbol{x})$$

- The *observed output* is:

$$Y_{\text{obs}}(\boldsymbol{x}) = f(\boldsymbol{x}) + \epsilon_{\text{meas}}(\boldsymbol{x})$$

where $\epsilon_{\text{meas}}(\boldsymbol{x})$ is the measurement error (independent random variables).

- The *output of the metamodel* is:

$$Y_{\text{meta}}(\boldsymbol{x}) = \sum_{j=1}^{p} \beta_j \phi_j(\boldsymbol{x}),$$

where $\Phi_j(\boldsymbol{x})$ are given functions (e.g. monomials) and $\boldsymbol{\beta} = (\beta_j)_{j=1}^p$ is a calibration parameter.

- Given $(\boldsymbol{x}^{(i)})_{i=1}^n$ and a set $\boldsymbol{y}_{\text{obs}} = (Y_{\text{obs}}(\boldsymbol{x}^{(i)}))_{i=1}^n$ of observations.
- Calibration. Determine the best $\boldsymbol{\beta}$, with uncertainty quantification.
- Prediction. Determine the value at a new point $\boldsymbol{x}^{(0)}$ of $f(\boldsymbol{x}^{(0)})$ or $Y_{\text{obs}}(\boldsymbol{x}^{(0)})$, with uncertainty quantification.

Linear regression model

- Introduce the vector $\boldsymbol{y}_{\rm obs} = \left(Y_{\rm obs}(\boldsymbol{x}^{(i)})\right)_{i=1}^n$ and the matrix $\mathbf{H}$:

$$H_{ij} = \phi_j(\boldsymbol{x}^{(i)}), \quad i = 1, \ldots, n, \quad j = 1, \ldots, p,$$

the minimization problem can be written as:

$$\widehat{\boldsymbol{\beta}} = \underset{\boldsymbol{\beta}\in\mathbb{R}^p}{\rm argmin} \left\{ \sum_{i=1}^n \left| Y_{\rm obs}(\boldsymbol{x}^{(i)}) - \sum_{j=1}^p \beta_j \phi_j(\boldsymbol{x}^{(i)}) \right|^2 \right\} = \underset{\boldsymbol{\beta}\in\mathbb{R}^p}{\rm argmin} \left\{ \left\| \boldsymbol{y}_{\rm obs} - \mathbf{H}\boldsymbol{\beta} \right\|^2 \right\}$$

$\hookrightarrow$ least-squares, quadratic minimization problem.

- Solution:

$$\widehat{\boldsymbol{\beta}} = \mathbf{H}^+ \boldsymbol{y}_{\rm obs}$$

where $\mathbf{H}^+$ is the pseudo-inverse of $\mathbf{H}$ ($\mathbf{H}^+ = \lim_{\delta\to 0}(\mathbf{H}^T\mathbf{H} + \delta \mathbf{I})^{-1}\mathbf{H}^T$).
$\rightarrow$ numerically it is often necessary to regularize :

$$\widehat{\boldsymbol{\beta}} = \mathbf{H}_\delta^+ \boldsymbol{y}_{\rm obs}, \qquad \mathbf{H}_\delta^+ = (\mathbf{H}^T\mathbf{H} + \delta \mathbf{I})^{-1}\mathbf{H}^T$$

- Surrogate model:

$$\hat{Y}(\boldsymbol{x}) = \sum_{j=1}^p \hat{\beta}_j \phi_j(\boldsymbol{x}) = \hat{\boldsymbol{\beta}}^T \boldsymbol{\Phi}(\boldsymbol{x}), \qquad \boldsymbol{\Phi}(\boldsymbol{x}) = (\phi_j(\boldsymbol{x}))_{j=1}^p$$

Bayesian viewpoint on linear regression

- We assume that we have a priori information on β, in the form of a prior distribution $\mathcal{N}(\beta_{\text{prior}}, \mathbf{Q}_{\text{prior}})$.
- The likelihood (distribution of $\boldsymbol{y}_{\text{obs}}$ given β) is $\mathcal{N}(\mathbf{H}\beta, \sigma^2_{\text{meas}}\mathbf{I})$.

$\hookrightarrow$ Bayes theorem.

- The posterior distribution of β given $\boldsymbol{y}_{\text{obs}}$ is $\mathcal{N}(\beta_{\text{post}}, \mathbf{Q}_{\text{post}})$ with

$$\beta_{\text{post}} = \beta_{\text{prior}} + (\sigma^2_{\text{meas}}\mathbf{Q}^{-1}_{\text{prior}} + \mathbf{H}^T\mathbf{H})^{-1}\mathbf{H}^T(\boldsymbol{y}_{\text{obs}} - \mathbf{H}\beta_{\text{prior}})$$

$$\mathbf{Q}_{\text{post}} = \sigma^2_{\text{meas}}(\sigma^2_{\text{meas}}\mathbf{Q}^{-1}_{\text{prior}} + \mathbf{H}^T\mathbf{H})^{-1}$$

- If $\mathbf{Q}_{\text{prior}}$ is large, then we recover the least-squares results:

$$\beta_{\text{post}} = \mathbf{H}^+\boldsymbol{y}_{\text{obs}} = \widehat{\beta}$$

$$\mathbf{Q}_{\text{post}} = \sigma^2_{\text{meas}}(\mathbf{H}^T\mathbf{H})^+$$

- Warning about the propagation of uncertainty for the prediction:

$$Y(\boldsymbol{x}) \mid \boldsymbol{y}_{\text{obs}} \sim \mathcal{N}(\beta^T_{\text{post}}\boldsymbol{\Phi}(\boldsymbol{x}), \boldsymbol{\Phi}(\boldsymbol{x})^T\mathbf{Q}_{\text{post}}\boldsymbol{\Phi}(\boldsymbol{x}))$$

- Residue test based on $\widehat{\varepsilon} = (Y_{\text{obs}}(\boldsymbol{x}^{(i)}) - \beta^T_{\text{post}}\Phi(\boldsymbol{x}^{(i)}))_{i=1}^n$ allows to answer the question: is the model compatible with data ?

Example 1

The true function is ($x \in [0,1]$):

$$f(x) = x$$

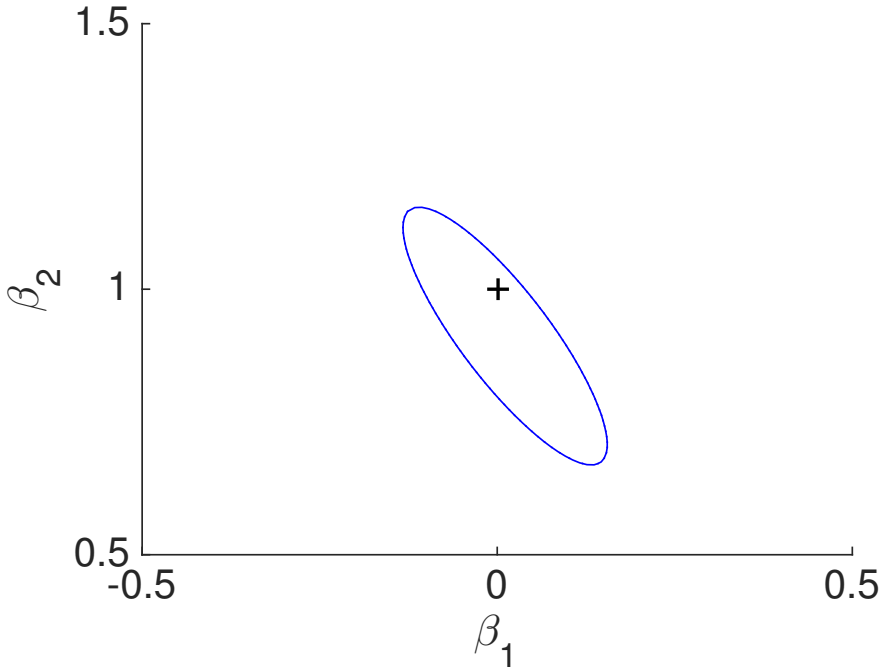
The observed function is:

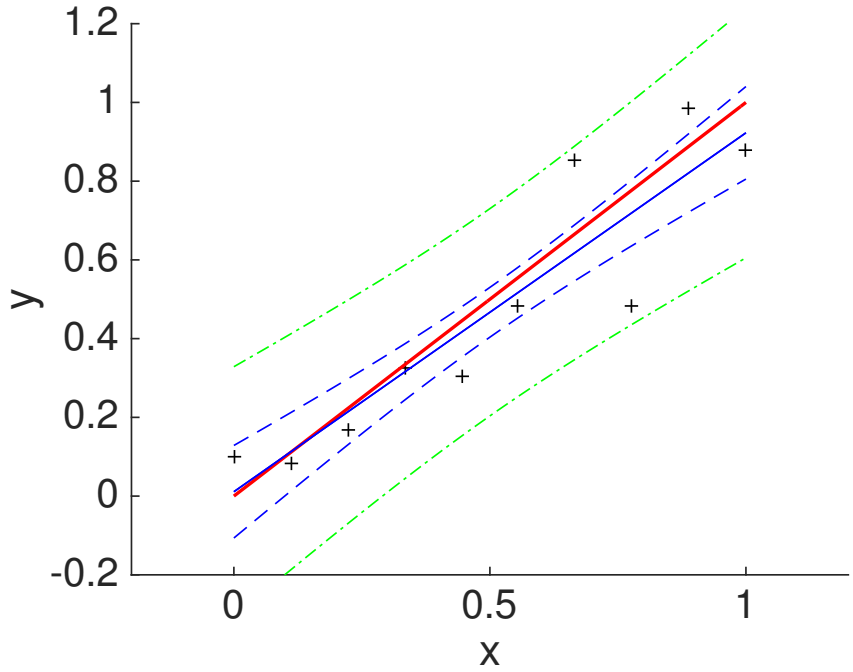
$$Y_{\text{obs}}(x) = f(x) + \epsilon_{\text{meas}}(x)$$

where $\epsilon_{\text{meas}}(x) \sim \mathcal{N}(0, \sigma^2_{\text{meas}})$.
The metamodel with $p = 2$ parameters $\boldsymbol{\beta} = (\beta_1, \beta_2)^T$ is:

$$Y_{\text{meta}}(\boldsymbol{x}) = \beta_1 + \beta_2 x = \boldsymbol{\beta}^T \boldsymbol{\Phi}(x), \qquad \phi_1(x) = 1, \quad \phi_2(x) = x$$

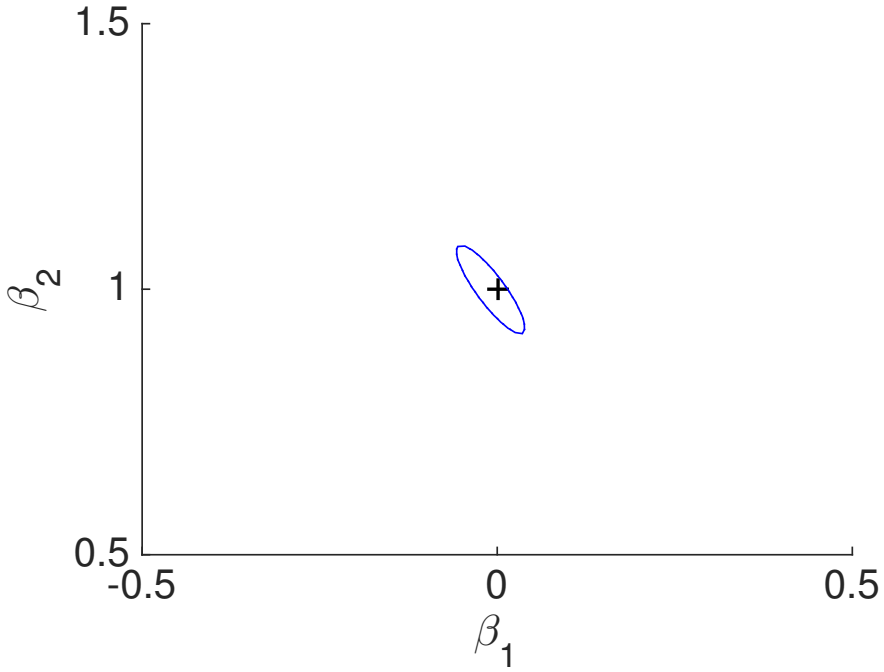
Example 1

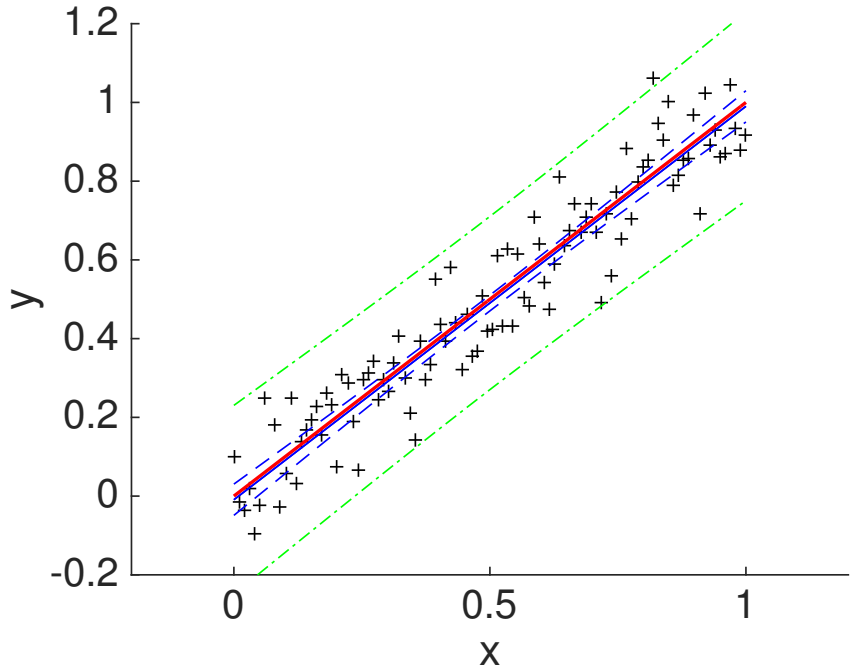




Here $n = 10$ and $\sigma_{\text{meas}} = 0.1$.
We find $\widehat{\beta} = (-0.11, 0.91)^T$.The true value is $\beta_{\text{true}} = (0, 1)^T$.
The tube of confidence for $f(x)$ is blue.
The tube of confidence for $Y_{\text{obs}}(x)$ is green.
Residue test: $\frac{1}{(n-p)\sigma^2_{\text{meas}}} \|\widehat{\varepsilon}\|^2 = 2.06 \rightarrow$ *accept (i.e., does not reject).*

Example 1





Here $n = 100$ and $\sigma_{\rm meas} = 0.1$.
We find $\widehat{\beta} = (-0.01, 1.00)^T$. The true value is $\beta_{\rm true} = (0, 1)^T$.
The tube of confidence for $f(x)$ is blue.
The tube of confidence for $Y_{\rm obs}(x)$ is green.
Residue test: $\frac{1}{(n-p)\sigma^2_{\rm meas}} \|\widehat{\varepsilon}\|^2 = 0.96 \rightarrow$ *accept (i.e., does not reject).*

Example 2

The true function is:

$$f(x) = x^2$$

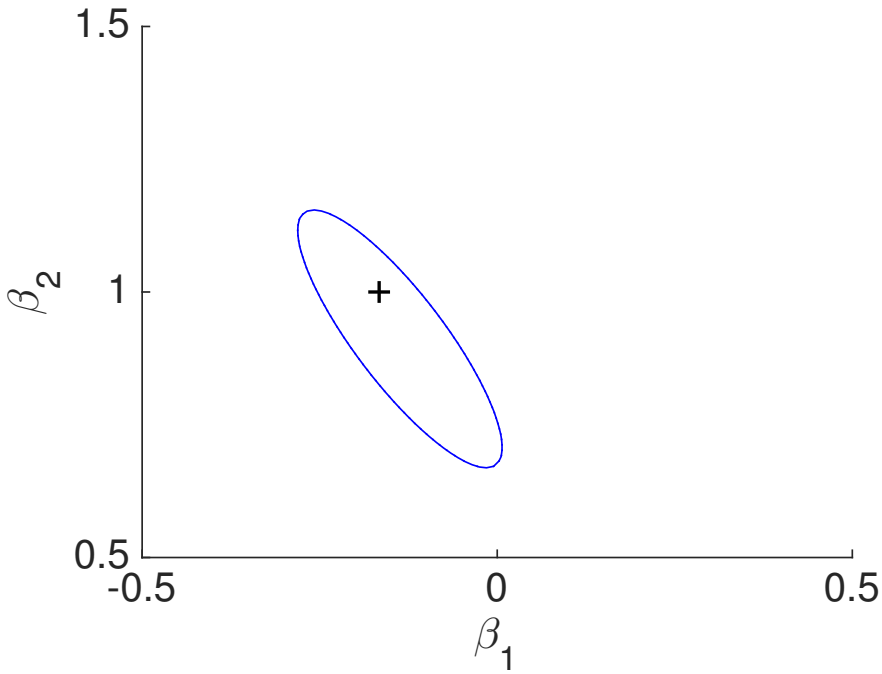
The observed function is:

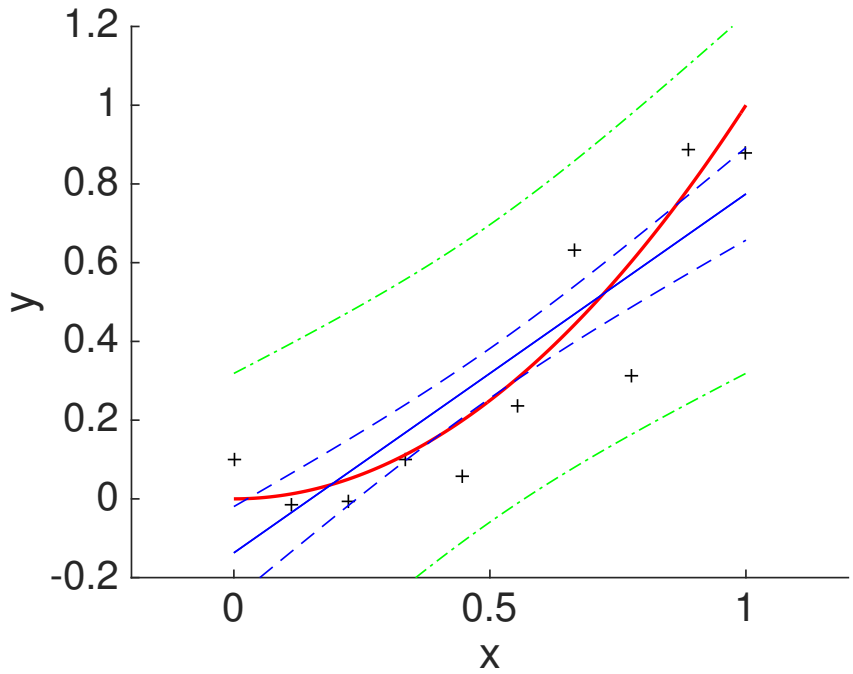
$$Y_{\text{obs}}(x) = f(x) + \epsilon_{\text{meas}}(x)$$

where $\epsilon_{\text{meas}}(x) \sim \mathcal{N}(0, \sigma^2_{\text{meas}})$.
The metamodel with $p = 2$ parameters $\boldsymbol{\beta} = (\beta_1, \beta_2)^T$ is:

$$Y_{\text{meta}}(\boldsymbol{x}) = \beta_1 + \beta_2 x = \boldsymbol{\beta}^T \boldsymbol{\Phi}(x), \qquad \phi_1(x) = 1, \quad \phi_2(x) = x$$

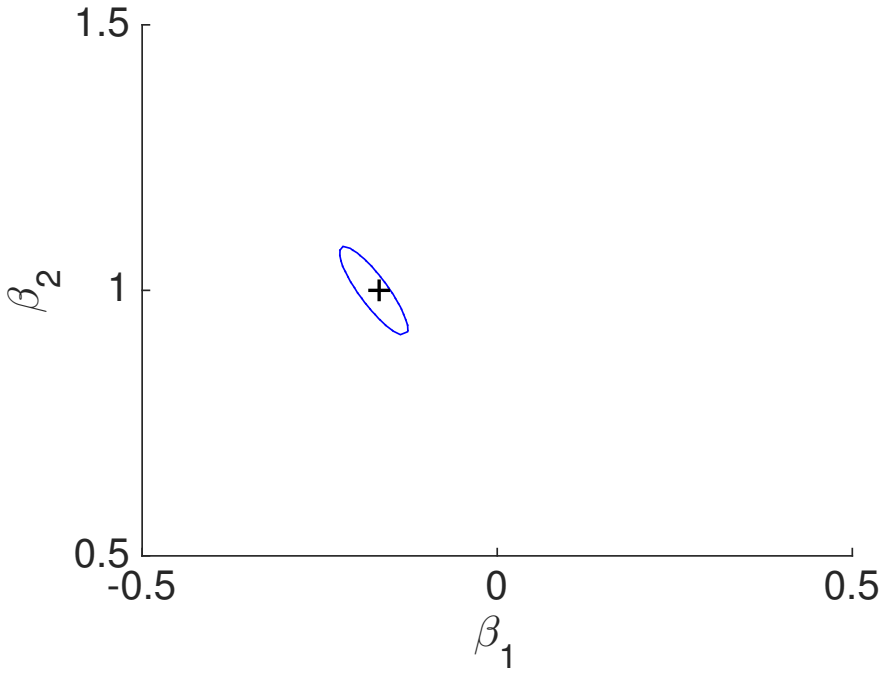
Example 2

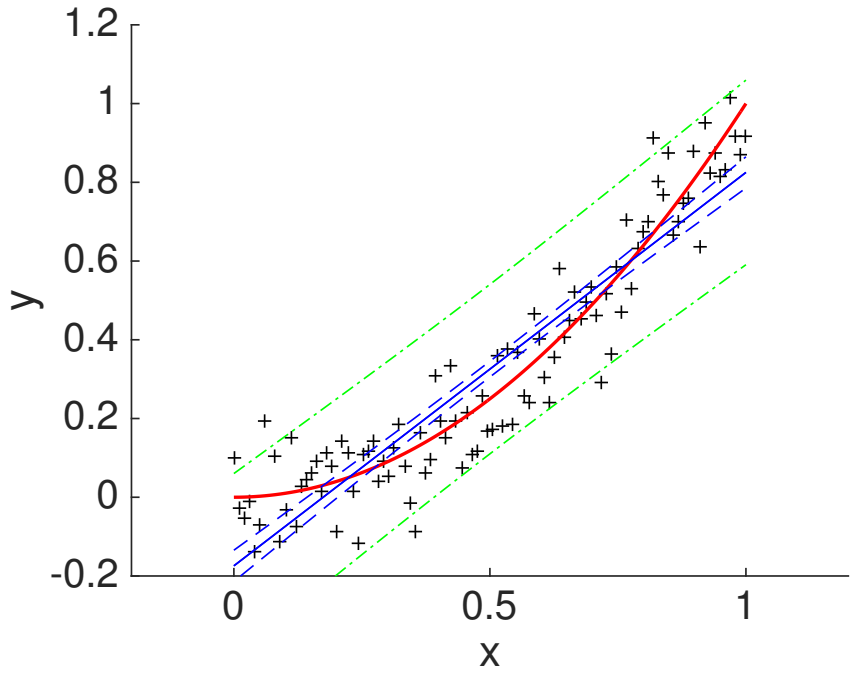




Here $n = 10$ and $\sigma_{\text{meas}} = 0.1$.
We find $\widehat{\beta} = (-0.14, 0.91)^T$.
$\rightarrow$ A small covariance for β means that we have identified the best model among the class of models, not that the model is good.
Residue test: $\frac{1}{(n-p)\sigma^2_{\text{meas}}} \|\widehat{\varepsilon}\|^2 = 3.48 \rightarrow$ *reject.*

Example 2





Here $n = 100$ and $\sigma_{\text{meas}} = 0.1$.
We find $\widehat{\beta} = (-0.17, 0.91)^T$.
$\rightarrow$ A small covariance for β means that we have identified the best model among the class of models, not that the model is good.
Residue test: $\frac{1}{(n-p)\sigma^2_{\text{meas}}} \|\widehat{\varepsilon}\|^2 = 1.50 \rightarrow$ *reject.*

What should we do? Change the model !

Gaussian process regression

Gaussian process regression

• The metamodel obtained by linear regression cannot reproduce perfectly the true function (even with the best calibration of $\boldsymbol{\beta}$).
$\hookrightarrow$ We introduce a nonparametric model error $Z_{\mathrm{mod}}(\boldsymbol{x})$:

$$Y_{\mathrm{meta}}(\boldsymbol{x}) = \boldsymbol{\beta}^T \boldsymbol{\Phi}(\boldsymbol{x}) + Z_{\mathrm{mod}}(\boldsymbol{x})$$

• Bayesian approach: we have an a priori distribution for $\boldsymbol{\beta}$ and for the model error that we assume to be *a stationary Gaussian process with mean zero* and with covariance of the form

$$\mathbb{E}\left[Z_{\mathrm{mod}}(\boldsymbol{x}) Z_{\mathrm{mod}}(\boldsymbol{x}')\right] = C_{\mathrm{mod}}(\boldsymbol{x} - \boldsymbol{x}')$$

$\rightarrow$ Parametric model for C_{mod} (we speak about hyperparameters).

Covariance structure of the model error

a) RBF (Gaussian) model:

$$C_{\rm mod}(\boldsymbol{x}-\boldsymbol{x}') = \sigma^2 \exp\Big(-\frac{|\boldsymbol{x}-\boldsymbol{x}'|^2}{\ell_c^2}\Big)$$

◦ The hyperparameter σ^2 (variance) controls the amplitude of the model error.
◦ The hyperparameter ℓ_c (correlation radius) controls the range of the model error: Two points separated by less than ℓ_c have correlated errors.
◦ The errors are smooth.

Generalization of the Gaussian model to anisotropic models:

$$C_{\rm mod}(\boldsymbol{x}-\boldsymbol{x}') = \sigma^2 \exp\Big(-\sum_{j=1}^{d}\frac{(x_j-x_j')^2}{\ell_{c,j}^2}\Big)$$

where $\ell_{c,j}$ is the correlation radius of the model error in the jth dimension.

b) Matérn model:

$$C_{\rm mod}(\boldsymbol{x}-\boldsymbol{x}') = \frac{\sigma^2}{\Gamma(\nu)2^{\nu-1}}\Big(\frac{2\sqrt{\nu}|\boldsymbol{x}-\boldsymbol{x}'|}{\ell_c}\Big)^{\nu} K_\nu\Big(\frac{2\sqrt{\nu}|\boldsymbol{x}-\boldsymbol{x}'|}{\ell_c}\Big),$$

has three hyperparameters (σ^2, ℓ_c, ν). The hyperparameter $\nu \in [1/2, \infty)$ controls the regularity of the realizations.

◦ When $\nu \to \infty$, we get the Gaussian model (smooth, $\mathcal{C}^\infty$-realizations):

$$C_{\rm mod}(\boldsymbol{x}-\boldsymbol{x}') = \sigma^2 \exp\Big(-\frac{|\boldsymbol{x}-\boldsymbol{x}'|^2}{\ell_c^2}\Big)$$

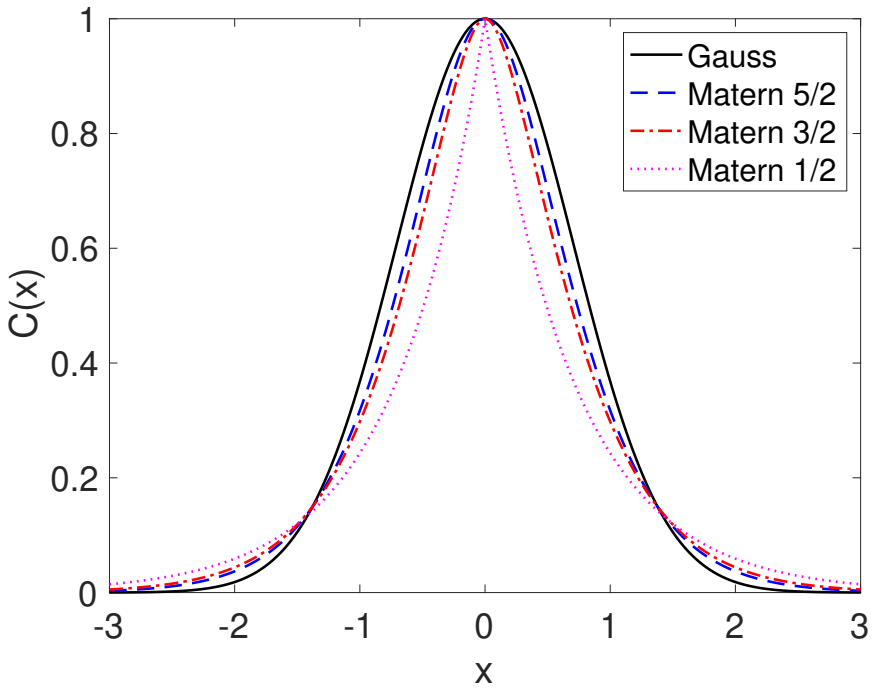
◦ When $\nu = 1/2$ we get the exponential model (OU process, rough).

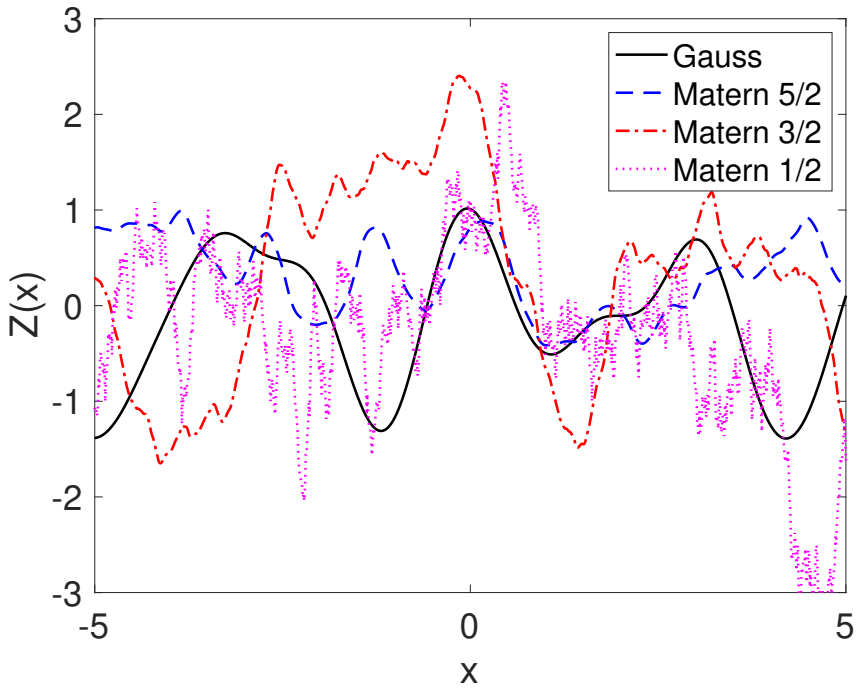
$$C_{\rm mod}(\boldsymbol{x}-\boldsymbol{x}') = \sigma^2 \exp\Big(-\frac{\sqrt{2}|\boldsymbol{x}-\boldsymbol{x}'|}{\ell_c}\Big)$$

◦ When $\nu = 5/2$ we get (intermediate, $\mathcal{C}^2$ realizations):

$$C_{\rm mod}(\boldsymbol{x}-\boldsymbol{x}') = \sigma^2\Big(1+\frac{\sqrt{2}|\boldsymbol{x}-\boldsymbol{x}'|}{\ell_c}+\frac{10|\boldsymbol{x}-\boldsymbol{x}'|^2}{3\ell_c}\Big)\exp\Big(-\frac{\sqrt{10}|\boldsymbol{x}-\boldsymbol{x}'|}{\ell_c}\Big)$$

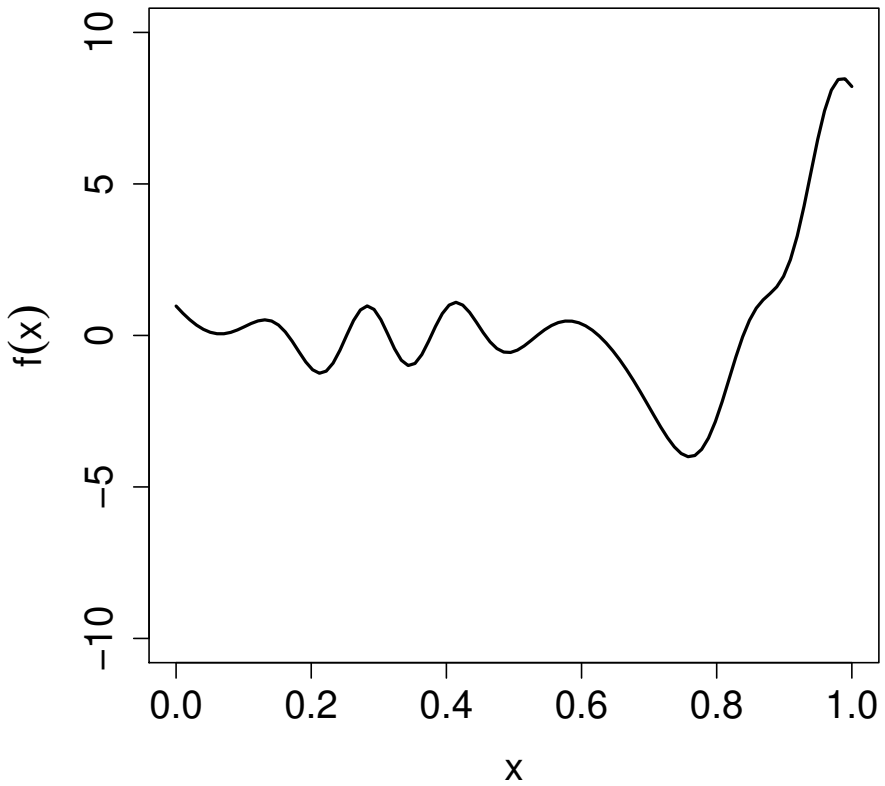
• It is possible to estimate the hyperparameters of $C_{\rm mod}$ from the observations (by maximum likelihood or by cross validation).





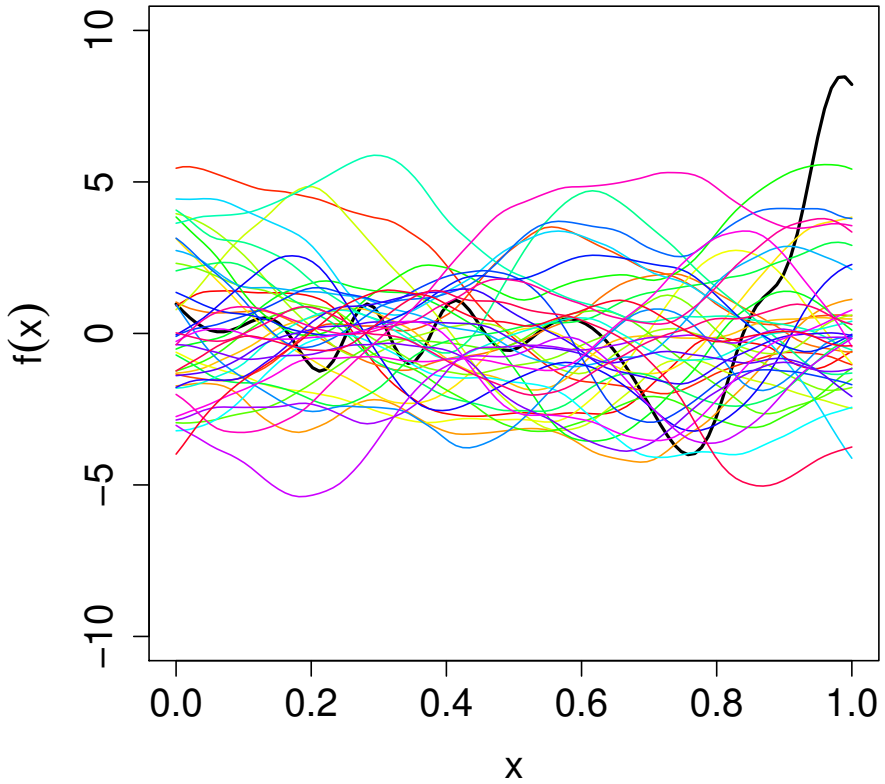
Left : covariance functions. Right: realizations.

Gaussian process regression pipeline



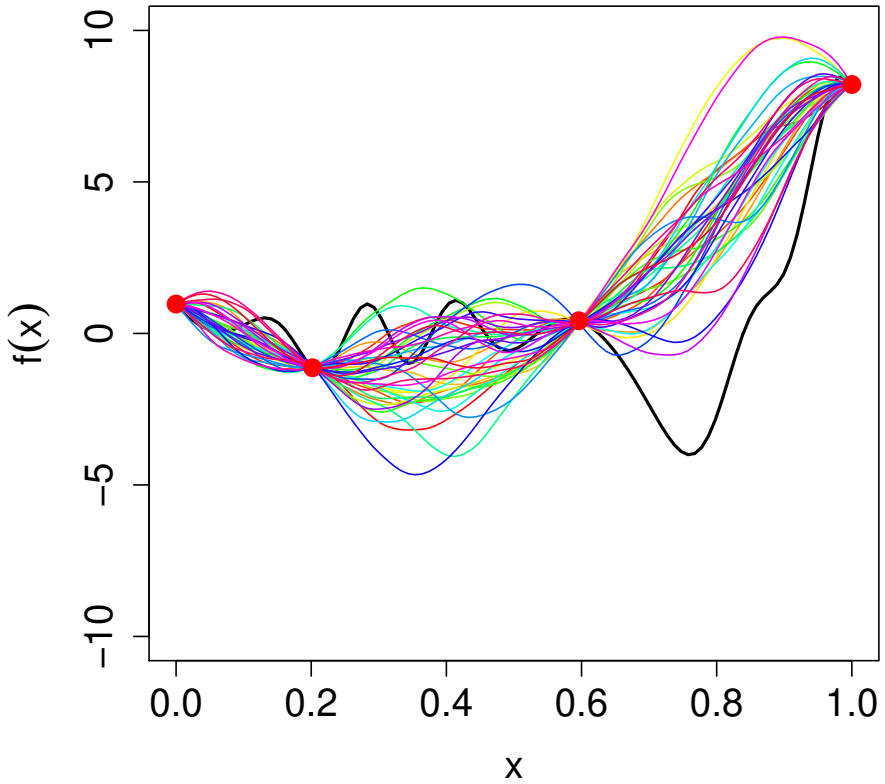
- Goal: build a metamodel for $f(x)$.

Gaussian process regression pipeline



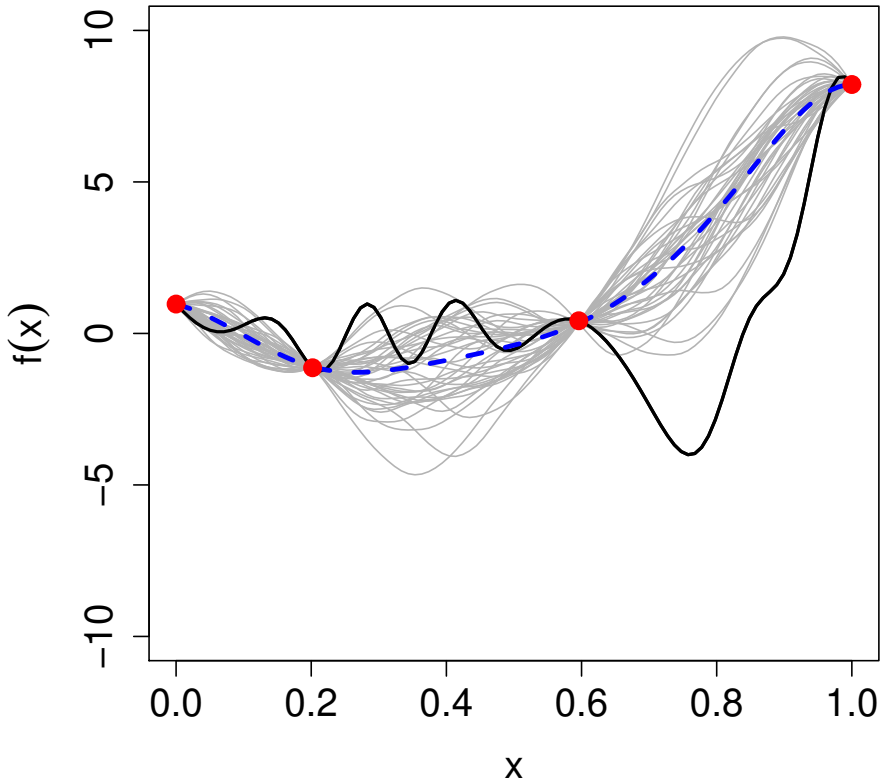
- Bayesian viewpoint
- Prior: a Gaussian process $Z(x)$ with mean $m(x)$ and covariance $k(x, x')$.
- For example $m(x) = 0$ and $k(x, x') = \sigma^2 \exp(-\frac{(x-x')^2}{\ell_c^2})$

Gaussian process regression pipeline



- We observe Z at points $x_1, \ldots, x_n \in \mathbb{R}$.
- We want to predict $Z(x)$ given the observed values $Z(x_1) = f(x_1), \ldots, Z(x_n) = f(x_n)$.
- Bayes theorem gives the posterior distribution of $Z(x)$. It is Gaussian with posterior mean $\hat{f}(x)$ and variance $\hat{\sigma}^2(x)$.

Gaussian process regression pipeline



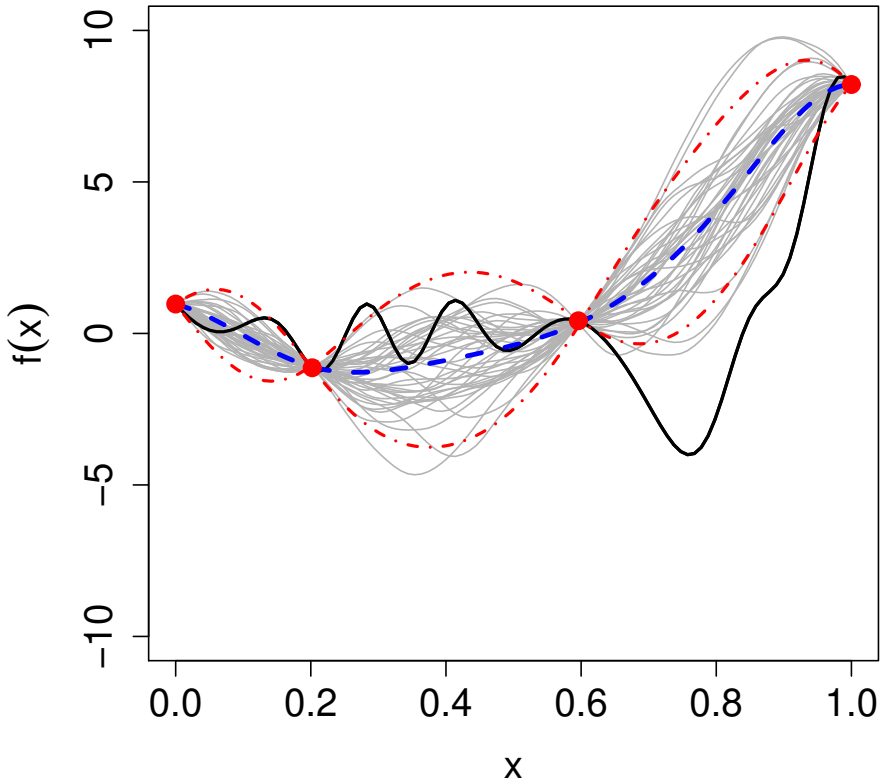
- The posterior mean is the Best Linear Unbiased Predictor (BLUP) :

$$\hat{f}(x) = m(x) + \boldsymbol{k}(x)^T \mathbf{K}^{-1}(\boldsymbol{f} - \boldsymbol{m})$$

where
$\boldsymbol{k}(x) = [k(x, x_i)]_{i=1}^n$
$\mathbf{K} = [k(x_i, x_j)]_{i,j=1}^n$
$\boldsymbol{f} = [f(x_i)]_{i=1}^n$
$\boldsymbol{m} = [m(x_i)]_{i=1}^n$

- Dashed blue line : BLUP

Gaussian process regression pipeline

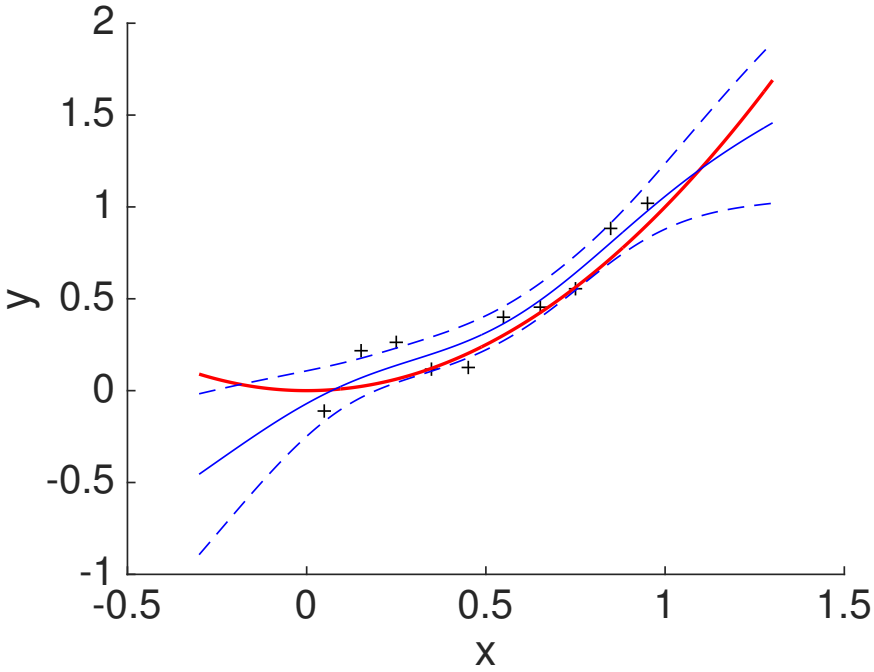


- The posterior mean minimizes the Mean Square Error (MSE) which is the posterior variance:

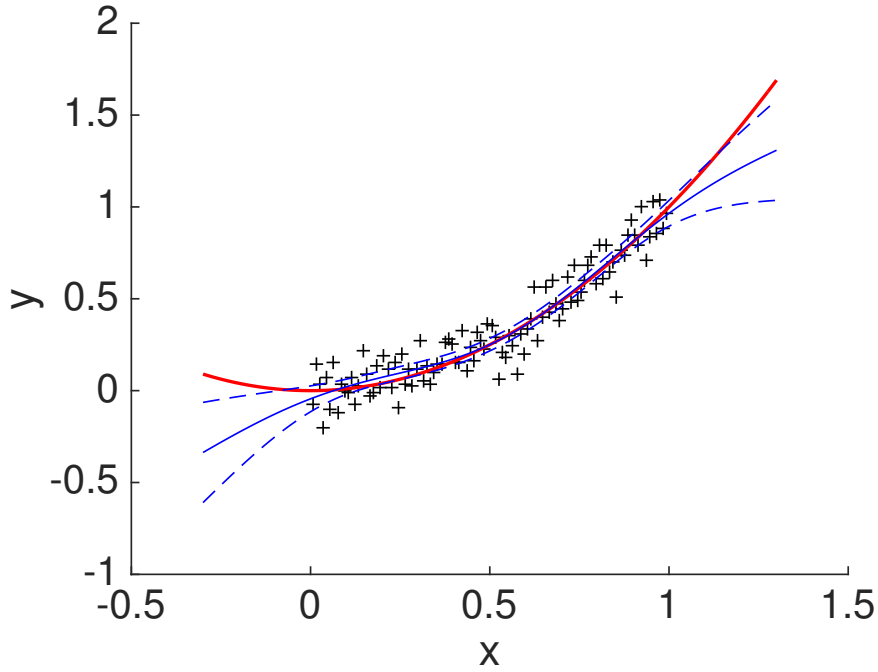
$$\hat{\sigma}^2(x) = k(x,x) - \boldsymbol{k}(x)^T \mathbf{K}^{-1} \boldsymbol{k}(x)$$

- Dashed blue line: BLUP (Best Linear Unbiased Predictor)
- Dash-dotted red lines: credibility interval $\hat{f}(x) \pm 2\hat{\sigma}(x)$.

Example 2



$n = 10$



$n = 100$

The true and observed functions are:

$$f(x) = x^2, \qquad Y_{\text{obs}}(x) = f(x) + \epsilon_{\text{meas}}(x)$$

The metamodel is:

$$Y_{\text{mod}}(x) = \beta_1 + \beta_2 x + Z_{\text{mod}}(x)$$

with $\mathbb{E}\left[Z_{\text{mod}}(x) Z_{\text{mod}}(x')\right] = C_{\text{mod}}(x - x')$ Gaussian.

Polynomial chaos expansion

Orthogonal polynomials

- Let w be a probability density on $\mathbb{R}$.
- A family of polynomials $(\phi_n)_{n\in\mathbb{N}}$ is called orthogonal for w if

1) ϕ_n has degree n for all n,

2) there exist positive real numbers $(\gamma_n)_{n\in\mathbb{N}}$ such that for all $n, m \in \mathbb{N}$:

$$\int_{\mathbb{R}} \phi_n(x)\phi_m(x)w(x)dx = \gamma_n \mathbf{1}_0(m-n).$$

If $\gamma_n = 1$, then the polynomials are orthonormal.

- Orthogonal polynomials are used in Gaussian quadrature, in linear regression (here) and in many other fields.

- Classical examples:

- $w(x) = \frac{1}{2}\mathbf{1}_{[-1,1]}(x) \rightarrow$ Legendre polynomials,

- $w(x) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{x^2}{2}\right) \rightarrow$ Hermite polynomials.

- If w has an exponential moment, then an orthormal polynomial family is an orthonormal basis of $L^2(w)$.

Polynomial chaos expansion

- Let X be a real-valued random variable with density w and $f : \mathbb{R} \to \mathbb{R}$.

If $Y = f(X)$ is square integrable, then:

$$Y = \sum_{n=0}^{\infty} y_n \phi_n(X),$$

where

$$y_n = \int_{\mathbb{R}} f(x)\phi_n(x)w(x)dx = \mathbb{E}\big[Y\phi_n(X)\big].$$

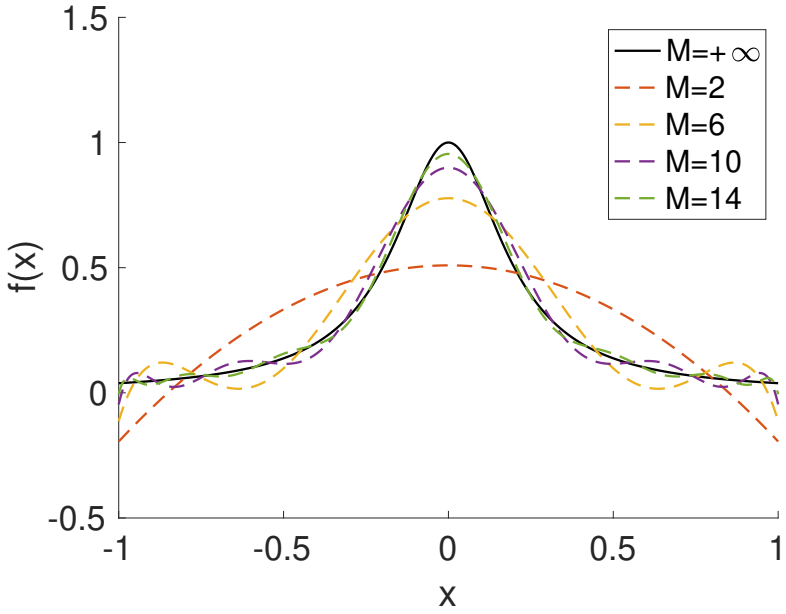
- In order to build a metamodel, we truncate the expansion:

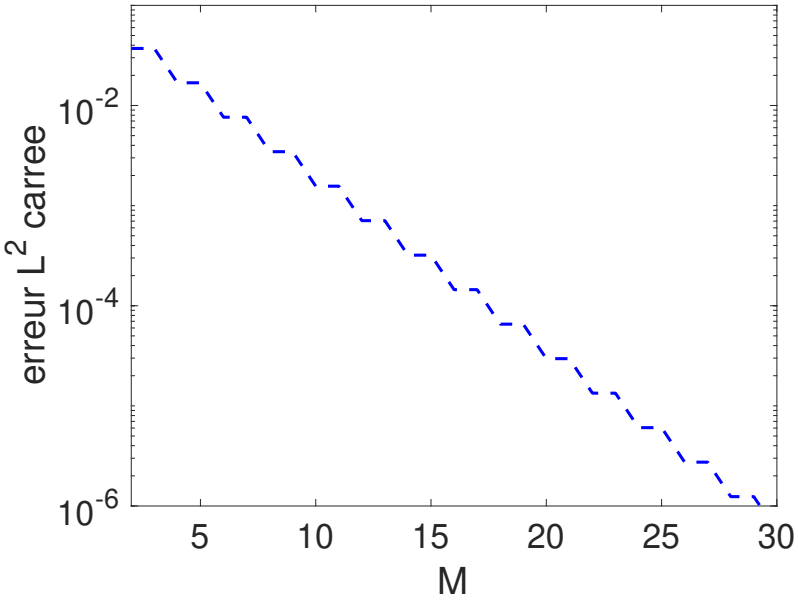
$$Y^{(M)} = \sum_{n=0}^{M} y_n \phi_n(X).$$

$Y^{(M)}$ is the best approximation (in L^2 sense) of Y by a polynomial in X with degree at most M:

$$\mathbb{E}\big[(Y - Y^{(M)})^2\big] = \inf_{p \in \mathcal{F}_M} \mathbb{E}\big[(Y - p(X))^2\big]$$

We will see that it also possesses additional valuable properties.





Polynomial Chaos Metamodeling of

$$x \in [-1,1] \mapsto f(x) = \frac{1}{1+25x^2}$$

Mth polynomial:

$$P_M(x) = \sum_{j=0}^{M} y_j \phi_j(x)$$

with ϕ_j =normalized Legendre polynomial and $y_j = \frac{1}{2}\int_{-1}^{1} \phi_j(x) f(x) dx$.

Generalized polynomial chaos

- We assume:
  - The random vector $\boldsymbol{X} = (X_j)_{j=1}^d$ with X_i *independent* with density $(w_j)_{j=1}^d$.
  - The output is $Y = f(\boldsymbol{X})$ (square integrable).
  - For any $j = 1, \ldots, d$, we know an orthonormal polynomial family $\{\phi_{\alpha_j}^{(j)}, \alpha_j \in \mathbb{N}\}$ in $L^2(\mathbb{R}, w_j)$.
- Then
  - The tensorized polynomials $\phi_{\boldsymbol{\alpha}}(\boldsymbol{x}) = \phi_{\alpha_1}^{(1)}(x_1) \cdots \phi_{\alpha_d}^{(d)}(x_d)$ are orthonormal,
$$\mathbb{E}\left[\phi_{\boldsymbol{\alpha}}(\boldsymbol{X})\phi_{\boldsymbol{\alpha}'}(\boldsymbol{X})\right] = \delta_{\boldsymbol{\alpha}\boldsymbol{\alpha}'}$$
  - The random variable $Y = f(\boldsymbol{X})$ can be expanded as:
$$Y = \sum_{\boldsymbol{\alpha} \in \mathbb{N}^d} y_{\boldsymbol{\alpha}} \phi_{\boldsymbol{\alpha}}(\boldsymbol{X}), \qquad y_{\boldsymbol{\alpha}} = \mathbb{E}[f(\boldsymbol{X})\phi_{\boldsymbol{\alpha}}(\boldsymbol{X})].$$

Polynomial chaos metamodel

• Fix a truncation method (along the degree) for the polynomial chaos expansion, relabel the polynomials $(\Phi_j)_{j\in\mathbb{N}} = (\phi_{\alpha_1,\ldots,\alpha_d})_{\alpha_1,\ldots,\alpha_d\in\mathbb{N}}$ so that the truncated polynomial has the form:

$$Y^{(M)} = \sum_{j=0}^{M} y_j \Phi_j(\boldsymbol{X}),$$

with

$$y_j = \mathbb{E}\left[\Phi_j(\boldsymbol{X}) f(\boldsymbol{X})\right], \qquad j = 0, \ldots, M.$$

Estimation of the coefficients by Monte Carlo

- The jth coefficient y_j of the polynomial chaos expansion of $f(\boldsymbol{X})$ is:

$$y_j = \mathbb{E}\left[\Phi_j(\boldsymbol{X}) f(\boldsymbol{X})\right].$$

- If $(\boldsymbol{X}^{(i)}, f(\boldsymbol{X}^{(i)}))_{i=1}^n$ is an i.i.d. sample, then y_j can be estimated by

$$\hat{y}_{\mathrm{MC},j} = \frac{1}{n}\sum_{i=1}^{n} \Phi_j(\boldsymbol{X}^{(i)}) f(\boldsymbol{X}^{(i)})$$

This is a convergent estimator of y_j by the law of large numbers.

Estimation of the coefficients by regression

- We want to approximate $Y = f(\boldsymbol{X})$ by a metamodel $\boldsymbol{\beta}^T \boldsymbol{\Phi}(\boldsymbol{X})$ with

$$\boldsymbol{\Phi}(\boldsymbol{x}) = (\Phi_j(\boldsymbol{x}))_{j=0}^M, \qquad \boldsymbol{\beta} = (\beta_j)_{j=0}^M.$$

The idea is to minimize the residual error:

$$\boldsymbol{y}_{\text{reg}} = \underset{\boldsymbol{\beta} \in \mathbb{R}^{M+1}}{\text{argmin}} \ \mathbb{E}\big[(\boldsymbol{\beta}^T \boldsymbol{\Phi}(\boldsymbol{X}) - f(\boldsymbol{X}))^2\big].$$

By orthogonality of the polynomials, we have

$$\mathbb{E}\big[(\boldsymbol{\beta}^T \boldsymbol{\Phi}(\boldsymbol{X}) - f(\boldsymbol{X}))^2\big] = \sum_{j=0}^{M} (\beta_j - y_j)^2 + \sum_{j=M+1}^{\infty} y_j^2,$$

and $y_{\text{reg},j}$ is the coefficient y_j of the polynomial chaos expansion of $f(\boldsymbol{X})$, for $j \leq M$.
$\hookrightarrow$ If $(\boldsymbol{X}^{(i)}, f(\boldsymbol{X}^{(i)}))_{i=1}^n$ is an i.i.d. sample, the coefficients can be estimated by

$$\hat{\boldsymbol{y}}_{\text{reg}} = \underset{\boldsymbol{\beta} \in \mathbb{R}^{M+1}}{\text{argmin}} \ \sum_{i=1}^{n} (\boldsymbol{\beta}^T \boldsymbol{\Phi}(\boldsymbol{X}^{(i)}) - f(\boldsymbol{X}^{(i)}))^2.$$

- We must solve:

$$\hat{\boldsymbol{y}}_{\rm reg} = \underset{\boldsymbol{\beta}\in\mathbb{R}^{M+1}}{\rm argmin} \ \|\mathbf{H}\boldsymbol{\beta} - \boldsymbol{F}\|^2,$$

with $H_{ij} = \Phi_j(\boldsymbol{X}^{(i)})$ and $F_i = f(\boldsymbol{X}^{(i)})$.

- Solution:

$$\hat{\boldsymbol{y}}_{\rm reg} = \mathbf{H}^{+}\boldsymbol{F}$$

- *Remember*: Monte Carlo method:

$$\hat{\boldsymbol{y}}_{\rm MC} = \frac{1}{n}\mathbf{H}^T\boldsymbol{F}$$

$\hookrightarrow$ the regression method consists in applying the pseudo-inverse $\lim_{\delta\to 0}(\mathbf{H}^T\mathbf{H} + \delta\mathbf{I})^{-1}\mathbf{H}^T$ to $\boldsymbol{F}$, rather than its approximation $\frac{1}{n}\mathbf{H}^T$.

- Remark 1: the difference with MC is small but nonnegligible !

$$(\mathbf{H}^T\mathbf{H})_{jj'} = \sum_{i=1}^{n}\phi_j(\boldsymbol{X}^{(i)})\phi_{j'}(\boldsymbol{X}^{(i)}) \simeq n\mathbb{E}[\phi_j(\boldsymbol{X})\phi_{j'}(\boldsymbol{X})] = n\delta_{jj'}$$

- Remark 2: if we compare with linear regression with an arbitrary basis (for instance, monomials), then we can see that $\mathbf{H}$ is **well-conditioned** !

- How to select M ?
- The generalization error results from a trade-off bias-variance:

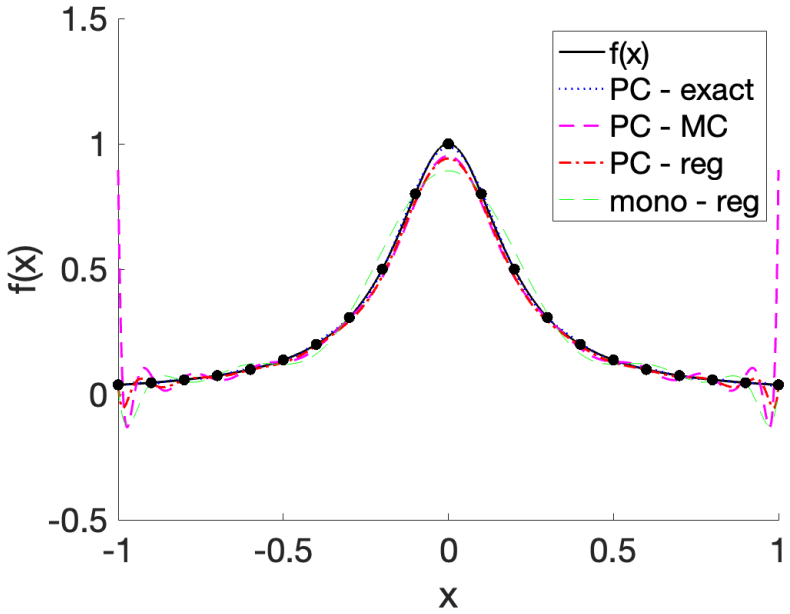
$$\mathbb{E}\left[\left(f(\boldsymbol{X}) - \hat{\boldsymbol{y}}_{\text{reg}}^T \boldsymbol{\Phi}(\boldsymbol{X})\right)^2 \mid \hat{\boldsymbol{y}}_{\text{reg}}\right] = \underbrace{\sum_{j=0}^{M} (\hat{y}_{\text{reg},j} - y_j)^2}_{\textit{estimation error}} + \underbrace{\sum_{j=M+1}^{\infty} y_j^2}_{\textit{bias}}$$

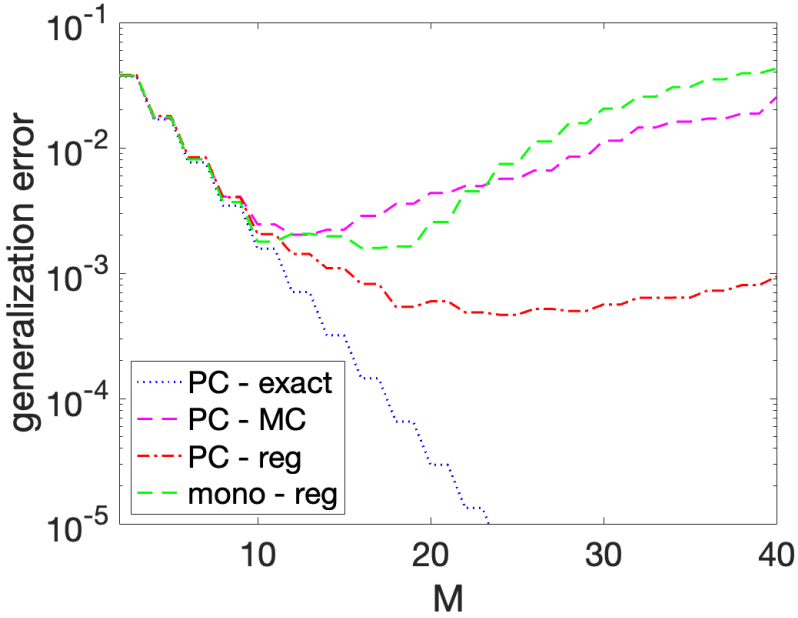
with $\hat{\boldsymbol{y}}_{\text{reg}}$ determined from a sample of size n.

- For fixed M, when n increases:
- the biais does not change (error between $f(\boldsymbol{X})$ and $\sum_{j=0}^{M} y_j \Phi_j(\boldsymbol{X})$),
- the estimation error decays (goes to 0 as $1/n$).

- For fixed n, when M increases:
- the biais decays (all the faster as f is smoother),
- the estimation error increases (there are more and more coefficients y_j to estimate).

- Selection of M by cross validation (e.g., minimization of the LOO error).





$M = 20,\ n = 21$

Polynomial Chaos Metamodeling of

$$x \in [-1, 1] \mapsto f(x) = \frac{1}{1 + 25x^2}$$

PC - exact: truncated PC, with $y_j = \mathbb{E}[\phi_j(X)f(X)]$.
PC - MC: truncated PC obtained with n points, with $\hat{y}_{j,\mathrm{MC}}$.
PC - reg: truncated PC obtained with n points, with $\hat{y}_{j,\mathrm{reg}}$.
mono - reg: ordinary polynomial regression (with monomials) obtained with n points.

Neural network

Neural network surrogate model

- Given a sample dataset $(\boldsymbol{x}^{(i)}, y^{(i)})_{i=1}^n$ with $y^{(i)} = f(\boldsymbol{x}^{(i)})$.
  1. Choose the network architecture:
     - Example: Fully connected feed-forward network
     - Layers: $\boldsymbol{x} \to h_1 \to h_2 \to \cdots \to \hat{y}$
     - Activation: ReLU, tanh, ...
  2. Train the network parameters θ by minimizing the loss:

$$\underset{\theta}{\operatorname{argmin}} \sum_{i=1}^{n} \left(y^{(i)} - NN_\theta(\boldsymbol{x}^{(i)})\right)^2$$

     Supports gradient-based optimization via automatic differentiation.
     Requires regularization: dropout, weight decay, ...
  3. Validate the model on a test set to prevent overfitting.

Surrogate models in one slide

1. Expectation:
$$\mathbb{E}[Y] = \operatorname*{argmin}_{a\in\mathbb{R}} \mathbb{E}\big[(Y-a)^2\big]$$
2. Linear regression:
$$Y_{\text{reg}} = \alpha_{\text{reg}} + \beta_{\text{reg}}^T \boldsymbol{X}, \text{ where } (\alpha_{\text{reg}}, \beta_{\text{reg}}) = \operatorname*{argmin}_{\alpha\in\mathbb{R},\beta\in\mathbb{R}^d} \mathbb{E}\big[(Y-\alpha-\beta^T\boldsymbol{X})^2\big]$$
3. Polynomial chaos:
$$Y^{(M)} = P_M(\boldsymbol{X}), \text{ where } P_M = \operatorname*{argmin}_{p\in\mathcal{F}_M} \mathbb{E}\big[(Y-p(\boldsymbol{X}))^2\big]$$
4. Neural network:
$$Y_{\text{NN}} = \text{NN}_{\theta^*}(\boldsymbol{X}), \text{ where } \theta^* = \operatorname*{argmin}_{\theta\in\mathbb{R}^q} \mathbb{E}\big[(Y-\text{NN}_\theta(\boldsymbol{X}))^2\big]$$
5. Conditional expectation:
$$\mathbb{E}[Y|\boldsymbol{X}] = \Psi(\boldsymbol{X}), \text{ where } \Psi = \operatorname*{argmin}_{\psi:\mathbb{R}^d\to\mathbb{R}} \mathbb{E}\big[(Y-\psi(\boldsymbol{X}))^2\big]$$

Estimation obtained by replacing $\mathbb{E}[F(\boldsymbol{X}, Y)]$ by $\sum_{i=1}^n F(\boldsymbol{x}^{(i)}, y^{(i)})$.

Summary: Linear Regression

- Advantages:
  - Simple, fast, analytical solution.
  - Easy to interpret (coefficients directly show influence).
  - Works well when the true relation is "linear" or nearly so.
  - Requires little data.
- Drawbacks:
  - Limited expressiveness (cannot model complicated nonlinearities).
  - Sensitive to multicollinearity.
  - Assumes additive Gaussian noise.
- UQ aspects:
  - Can provide analytical confidence intervals on coefficients and predictions under Gaussian noise assumptions.
  - However, does not capture model-form error.
  - Overly optimistic if true relation is "nonlinear".

Summary: Polynomial Chaos Expansion

- Advantages:
  - Built for UQ: designed to represent random inputs and outputs explicitly.
  - Analytical decomposition of uncertainty propagation.
  - Fast evaluation once built (polynomial surrogate).
- Drawbacks:
  - Grows exponentially with number of random variables (curse of dimensionality).
  - Depends on the input distribution.
  - Requires smooth response (poor for highly nonlinear or discontinuous systems).
- UQ aspects:
  - Gives analytic expressions for moments (mean, variance, etc) of the output.
  - Coefficients directly linked to Sobol sensitivity indices.
  - Does not capture model-form error (no explicit predictive variance on the model error).

Summary: Gaussian Process Regression

- Advantages:
  - Excellent for small-to-medium datasets.
  - Provides built-in probabilistic uncertainty estimate at each prediction point.
  - Flexible and non-parametric.
  - Theoretically grounded in Bayesian inference.
- Drawbacks:
  - Computationally expensive (training cost is $O(n^3)$).
  - Poor scalability in high dimensions ($> 10 - 20$ variables).
  - Hyperparameter tuning can be delicate: optimization should be carried out carefully.
- UQ aspects:
  - Strong UQ capability: each prediction gives a mean and variance (posterior predictive distribution).
  - Can over- or under-estimate uncertainty if kernel/hyperparameters are mis-specified.
  - Excellent for adaptive sampling / active learning.

Summary: Neural Network

- Advantages:
  - Very flexible (can approximate any continuous function).
  - Scales well with high-dimensional data.
  - Many architectures (MLP, CNN, etc.) for different data types.
- Drawbacks:
  - Training can be computationally expensive and data-hungry.
  - Poor interpretability.
  - No inherent UQ: must be added externally.
  - Sensitive to architecture, hyperparameters, and data scaling.
- UQ aspects:
  - Standard NN gives point estimates only (no uncertainty).
  - UQ can be added via Monte Carlo Dropout, Bayesian Neural Networks, or deep ensembles, which approximate predictive uncertainty.
  - Hard to calibrate and interpret uncertainty.

Sensitivity analysis

Sensitivity analysis

- Goal: Quantify how uncertainty in $Y = f(\boldsymbol{X})$ is apportioned to different sources of uncertainty in $\boldsymbol{X}$.
- Global sensitivity analysis uses entire input-output relationship.
- Two main objectives for the definition of global sensitivity indices:
  - it should explain the complex relationship between the inputs and the output and be as quantitative as possible,
  - it should be estimated with as few calls as possible.

References

- A. Saltelli et al., *Global Sensitivity Analysis: The Primer*, Wiley, 2008.
- S. Da Veiga, F. Gamboa, B. Iooss, and C. Prieur, *Basics and Trends in Sensitivity Analysis*, SIAM, 2021.

Variance-based approaches to sensitivity analysis

First-order Sobol' index

- Let $\boldsymbol{X} = (X_j)_{j=1}^d$, $Y = f(\boldsymbol{X})$ square integrable, $j \in \{1, \dots, d\}$.
  - $f_j(X_j) = \mathbb{E}[Y|X_j]$ is the best approximation of Y by a function of X_j:

$$f_j = \underset{g \in L^2_{P_j}}{\operatorname{argmin}} \ \mathbb{E}\big[(Y - g(X_j))^2\big]$$

  - Remember the total variance formula:

$$\mathrm{Var}(Y) = \underbrace{\mathrm{Var}(\mathbb{E}[Y|X_j])}_{\text{explained}} + \underbrace{\mathbb{E}[\mathrm{Var}(Y|X_j)]}_{\text{residual}}.$$

  - If $\mathrm{Var}(\mathbb{E}[Y|X_j]) \simeq \mathrm{Var}(Y)$, then $\mathrm{Var}(Y|X_j)$ is much smaller than $\mathrm{Var}(Y)$, which means that fixing X_j strongly reduces the fluctuations of $Y \Longrightarrow Y$ essentially depends on X_j.

$\hookrightarrow$ First-order Sobol' index

$$S_j = \frac{\mathrm{Var}(\mathbb{E}[Y|X_j])}{\mathrm{Var}(Y)} \quad \in [0, 1]$$

If $S_j \simeq 1$, then X_j is an *explanatory* factor.

Total Sobol' index

- First-order Sobol' index:

$$S_j = \frac{\mathrm{Var}(\mathbb{E}[Y|X_j])}{\mathrm{Var}(Y)}$$

$S_j \simeq 0$ does not imply that X_j is irrelevant !
[Example: $Y = X_1 X_2$, X_1, X_2 i.i.d. $\mathcal{N}(0,1) \to S_1 = S_2 = 0$.]

- Let $\boldsymbol{X} = (X_j)_{j=1}^d$, $Y = f(\boldsymbol{X})$, $j \in \{1, \ldots, d\}$, $\boldsymbol{X}_{-j} = (X_i)_{i \neq j}$.
  - If $\mathrm{Var}(Y|\boldsymbol{X}_{-j})$ is much smaller than $\mathrm{Var}(Y)$, then fixing $\boldsymbol{X}_{-j}$ strongly reduces the fluctuations of $Y \Longrightarrow Y$ does not depend on X_j.

$\hookrightarrow$ Total Sobol' index

$$T_j = \frac{\mathbb{E}[\mathrm{Var}(Y|\boldsymbol{X}_{-j})]}{\mathrm{Var}(Y)} \quad \in [0,1]$$

If $T_j \simeq 0$, then X_j is an *irrelevant* factor.
[Example: $Y = X_1 X_2$, X_1, X_2 i.i.d. $\mathcal{N}(0,1) \to S_1 = S_2 = 0$, $T_1 = T_2 = 1$.]

Hoeffding-Sobol' decomposition and analysis of variance

- For *independent* inputs $\boldsymbol{X} = (X_1, \ldots, X_d)$ and $f \in L^2_P$,

$$f(\boldsymbol{X}) = \sum_{u \subseteq \{1,\ldots,d\}} f_u(\boldsymbol{X}_u) = f_0 + \sum_i f_i(X_i) + \sum_{i<j} f_{ij}(X_i, X_j) + \cdots,$$

where $f_0 = \mathbb{E}[f(\boldsymbol{X})]$,

$$\mathbb{E}[f_u(\boldsymbol{X}_u)] = 0 \text{ if } u \neq \emptyset, \quad \mathbb{E}[f_u(\boldsymbol{X}_u) f_v(\boldsymbol{X}_v)] = 0 \text{ if } u \neq v.$$

[Uniqueness ensured by the condition $\mathbb{E}[f_u(\boldsymbol{X}_u)|\boldsymbol{X}_{u\setminus\{j\}}] = 0 \ \forall j \in u$].

- Then

$$\mathrm{Var}(Y) = \sum_{u \subseteq \{1,\ldots,d\}} \mathrm{Var}(f_u(\boldsymbol{X}_u)).$$

Sobol' indices:

$$S_u = \frac{\mathrm{Var}(f_u(\boldsymbol{X}_u))}{\mathrm{Var}(Y)}$$

such that $S_u \in [0,1]$, $\sum_u S_u = 1$.

Sobol' sensitivity indices

- **First-order index:**

$$S_j = \frac{\mathrm{Var}(\mathbb{E}[Y|X_j])}{\mathrm{Var}(Y)} = \frac{\mathrm{Var}(f_j(X_j))}{\mathrm{Var}(Y)}.$$

- **Total index:**

$$T_j = \frac{\mathbb{E}[\mathrm{Var}(Y|\boldsymbol{X}_{-j})]}{\mathrm{Var}(Y)} = \sum_{v, j \in v} \frac{\mathrm{Var}(f_v(\boldsymbol{X}_v))}{\mathrm{Var}(Y)} = \sum_{v, j \in v} S_v.$$

• Estimation seems tricky (variance of conditional expectation seems to require nested Monte Carlo), but it is actually possible by pick-freeze principle: If

$$Y = f(\boldsymbol{X}), \quad Y_j = f(X_j, \boldsymbol{X}'_{-j}), \quad Y_{-j} = f(X'_j, \boldsymbol{X}_{-j})$$

where $\boldsymbol{X}, \boldsymbol{X}'$ are i.i.d., then we have

$$S_j = \frac{\mathrm{Cov}(Y, Y_j)}{\mathrm{Var}(Y)}, \qquad T_j = \frac{1}{2\mathrm{Var}(Y)} \mathbb{E}[(Y - Y_{-j})^2].$$

Pick-freeze estimators

- To estimate S_j or T_j, use specific experimental design:

$$Y^{(i)} = f(\boldsymbol{X}^{(i)}), \quad Y_j^{(i)} = f(X_j^{(i)}, \boldsymbol{X}_{-j}^{\prime(i)}), \quad Y_{-j}^{(i)} = f(X_j^{\prime(i)}, \boldsymbol{X}_{-j}^{(i)}),$$

where $\boldsymbol{X}^{(i)}, \boldsymbol{X}^{\prime(i)}$, $i = 1, \ldots, n$, are i.i.d..
First-order index:

$$\widehat{S}_j^{PF} = \frac{\frac{1}{n}\sum_{i=1}^n Y^{(i)} Y_j^{(i)} - (\bar{Y})^2}{\frac{1}{n}\sum_{i=1}^n (Y^{(i)})^2 - (\bar{Y})^2}, \qquad \bar{Y} = \frac{1}{n}\sum_{i=1}^n Y^{(i)}$$

Total index:

$$\widehat{T}_j^{PF} = \frac{\frac{1}{2n}\sum_{i=1}^n (Y^{(i)} - Y_{-j}^{(i)})^2}{\frac{1}{n}\sum_{i=1}^n (Y^{(i)})^2 - (\bar{Y})^2}.$$

- Properties [Janon 2013]:
  - Consistent, asymptotically normal (convergence rate $O(n^{-1/2})$).
  - Asymptotically efficient (attains the Cramér-Rao lower bound).
  - Finite-sample concentration inequalities are also available.
  - Requires $(d+1)n$ model evaluations for all first-order indices.

Rank-based estimator

- Estimator of the first-order indices for real-valued inputs X_j [Gamboa 2022]:
  - Sample $(\boldsymbol{X}^{(i)}, Y^{(i)})_{i=1}^n$ with $Y^{(i)} = f(\boldsymbol{X}^{(i)})$ and $\boldsymbol{X}^{(i)}$ i.i.d..
  - Sort pairs $(X_j^{(i)}, Y^{(i)})_{i=1}^n$ by X_j and denote ordered $Y_{[i]}$.
  - Define

$$\widehat{S}_j^{\text{rank}} = \frac{\frac{1}{n}\sum_{i=1}^{n-1} Y_{[i]} Y_{[i+1]} - \bar{Y}^2}{\frac{1}{n}\sum_i Y_{[i]}^2 - \bar{Y}^2}, \qquad \bar{Y} = \frac{1}{n}\sum_{i=1}^n Y_{[i]} = \frac{1}{n}\sum_{i=1}^n Y^{(i)}$$

- Properties:
  - Uses n samples for all d first-order indices.
  - Consistent, convergence rate $O(n^{-1/2})$, asymptotically normal.
  - Interpretation of rank-based estimator:
    - If $Y \perp X_j$: consecutive $(Y_{[i]}, Y_{[i+1]})$ are independent $\rightarrow$ numerator ≈ 0.
    - If $Y = f(X_j)$: $Y_{[i+1]} \approx Y_{[i]} \rightarrow$ ratio ≈ 1.

Limitations and extensions of variance-based indices

- Depend only on second-order moments $\rightarrow$ insensitive to tail behavior, skewness ,...
- Require input independence for orthogonal ANOVA decomposition.
- Vector or functional outputs: extend to Hilbert space norms [Gamboa et al. 2014].
- Stochastic simulators: introduction of an extra latent variable, second-level sensitivity analysis [Fort et al. 2021].

Kernel-based approaches to sensitivity analysis

Sensitivity index based on independence criterion

- Principle: For $u \subset \{1, \dots, d\}$, let $\boldsymbol{X}_u = (X_j)_{j \in u}$. We have

$$Y \perp \boldsymbol{X}_u \implies \boldsymbol{X}_u \text{ is not influential on } Y$$

$\hookrightarrow$ Idea: Define an index S_u based on a distance (divergence) between $P_{\boldsymbol{X}_u, Y}$ and $P_{\boldsymbol{X}_u} \otimes P_Y$:

$$S_u \approx D\big(P_{\boldsymbol{X}_u, Y}, P_{\boldsymbol{X}_u} \otimes P_Y\big)$$

- Which distance ?

RKHS-based distance

- General distance between two probability measures P and Q on $\mathbb{R}^d$:

$$D_{\mathcal{H}}(P,Q) = \sup_{h\in\mathcal{H},\|h\|_{\mathcal{H}}\leq 1} \left|\mathbb{E}_{X\sim P}[h(X)] - \mathbb{E}_{Y\sim Q}[h(Y)]\right|$$

  - $\mathcal{H}$ = bounded functions $\Longrightarrow$ total variation distance
  - $\mathcal{H}$ = Lipschitz functions $\Longrightarrow$ Wasserstein-1 distance
  - $\mathcal{H}$ = Reproducing Kernel Hilbert Space (RKHS) $\Longrightarrow$ Maximum Mean Discrepancy (MMD)
- RKHS characterized by a symmetric positive definite kernel $\kappa(x,x')$:

$$\mathcal{H} = \overline{\mathcal{H}_0}, \qquad \mathcal{H}_0 = \left\{ h(\cdot) = \sum_{i=1}^{n} \alpha_i \kappa(x_i,\cdot),\ \alpha_i \in \mathbb{R}, x_i \in \mathbb{R}^d \right\}$$

with $\left\langle \sum_{i=1}^{n} \alpha_i \kappa(x_i,\cdot), \sum_{j=1}^{n'} \alpha'_j \kappa(x'_j,\cdot) \right\rangle = \sum_{i,j} \alpha_i \alpha'_j \kappa(x_i,x'_j)$.
Property: $h(x) = \langle h, \kappa(x,\cdot)\rangle$.
Example: if κ is a ν-Matérn kernel, then $\mathcal{H} = W^{\nu+d/2,2}(\mathbb{R}^d)$.

Maximum Mean Discrepancy (MMD)

- Let P and Q be probability measures on $\mathbb{R}^d$.
  - Given an RKHS $\mathcal{H}$ with reproducing kernel κ:

$$\mathrm{MMD}(P, Q) = \sup_{\|h\|_{\mathcal{H}} \leq 1} |\mathbb{E}_P[h(X)] - \mathbb{E}_Q[h(Y)]| .$$

  - Equivalently, using the kernel trick,

$$\mathrm{MMD}^2(P, Q) = \mathbb{E}_{P\otimes P}[\kappa(X, X')] + \mathbb{E}_{Q\otimes Q}[\kappa(Y, Y')] - 2\mathbb{E}_{P\otimes Q}[\kappa(X, Y)].$$

    $\hookrightarrow$ Monte Carlo-type estimators.
  - If κ is characteristic, then $\mathrm{MMD}(P, Q) = 0 \iff P = Q$.
    $\hookrightarrow$ X and Y are independent $\iff \mathrm{MMD}(P_{X,Y}, P_X \otimes P_Y) = 0$.
    Examples of characteristic kernel: Matérn or Gaussian kernel.

Empirical MMD estimator

- Given independent samples $\{x^{(i)}\}_{i=1}^n \sim P$, $\{y^{(i)}\}_{i=1}^n \sim Q$:

$$\widehat{\mathrm{MMD}}^2(P,Q) = \frac{1}{n^2}\sum_{i,j=1}^n \kappa(x^{(i)},x^{(j)}) + \frac{1}{n^2}\sum_{i,j=1}^n \kappa(y^{(i)},y^{(j)}) - \frac{2}{n^2}\sum_{i,j=1}^n \kappa(x^{(i)},y^{(j)})$$

  - Converge rate $O(n^{-1/2})$.
  - Choice of kernel controls sensitivity to different moments or scales.
  - If the kernel is characteristic, MMD detects all distributional differences.

Hilbert-Schmidt Independence Criterion (HSIC)

- Consider joint distribution $P_{X,Y}$ and product $P_X \otimes P_Y$. Define

$$\mathrm{HSIC}(X, Y) = \mathrm{MMD}^2(P_{X,Y},\, P_X \otimes P_Y).$$

  in terms of a given kernel κ, typically

$$\kappa((x, y), (x', y')) = \kappa_{\mathcal{X}}(x, x')\kappa_{\mathcal{Y}}(y, y')$$

- $\mathrm{HSIC}(X, Y) = 0 \iff X \perp Y$ (if κ is characteristic).
- Equivalent (kernel trick) formulation:

$$\begin{aligned}\mathrm{HSIC}(X, Y) =& \mathbb{E}[\kappa_{\mathcal{X}}(X, X')\kappa_{\mathcal{Y}}(Y, Y')] + \mathbb{E}[\kappa_{\mathcal{X}}(X, X')]\mathbb{E}[\kappa_{\mathcal{Y}}(Y, Y')] \\ & - 2\mathbb{E}[\kappa_{\mathcal{X}}(X, X')\kappa_{\mathcal{Y}}(Y, Y'')],\end{aligned}$$

  with (X, Y), (X', Y'), (X'', Y'') independent with distribution $P_{X,Y}$.

Empirical HSIC estimator

- Given samples $\{(x^{(i)}, y^{(i)})\}_{i=1}^n$ with distribution $P_{X,Y}$, let

$$K_{ij} = \kappa_{\mathcal{X}}(x^{(i)}, x^{(j)}), \quad L_{ij} = \kappa_{\mathcal{Y}}(y^{(i)}, y^{(j)}), \quad \mathbf{H} = \mathbf{I} - \frac{1}{n}\mathbf{1}\mathbf{1}^\top,$$

$$\tilde{K}_{ij} = (1 - \delta_{ij})K_{ij}, \quad \tilde{L}_{ij} = (1 - \delta_{ij})L_{ij}$$

  - Monte Carlo (U-statistics): unbiased but may be negative

$$\widehat{\mathrm{HSIC}}_{U,n} = \frac{1}{n(n-3)}\left(\mathrm{Tr}(\tilde{\mathbf{K}}\tilde{\mathbf{L}}) - \frac{(\mathbf{1}^T\tilde{\mathbf{K}}\mathbf{1})(\mathbf{1}^T\tilde{\mathbf{L}}\mathbf{1})}{(n-1)(n-2)} - \frac{2}{n-2}\mathbf{1}^T\tilde{\mathbf{K}}\tilde{\mathbf{L}}\mathbf{1}\right).$$

  - Monte Carlo (V-statistics): biased but nonnegative

$$\widehat{\mathrm{HSIC}}_{V,n} = \frac{1}{n^2}\mathrm{Tr}(\mathbf{KHLH}).$$

  - Practical choice of the kernels $\kappa_{\mathcal{X}}$ and $\kappa_{\mathcal{Y}}$: Gaussian with correlation length equal to the empirical standard deviation.

HSIC-based sensitivity indices

$$Y = f(\boldsymbol{X}), \qquad \boldsymbol{X} = (X_j)_{j=1}^d$$

- Define for a subset of variables $\boldsymbol{X}_u = (X_j)_{j\in u}$:

$$S_u^{HSIC} = \frac{\mathrm{HSIC}(\boldsymbol{X}_u, Y)}{\sqrt{\mathrm{HSIC}(\boldsymbol{X}_u, \boldsymbol{X}_u)\,\mathrm{HSIC}(Y, Y)}}.$$

  - Normalized dependence measure: $\mathrm{HSIC}(Y, \boldsymbol{X}_u) \in [0, 1]$.
  - $S_u^{HSIC} = 0$ if and only if $Y \perp \boldsymbol{X}_u$.
  - Unlike variance-based indices, HSIC-based indices S_u^{HSIC} are not constrained to sum to one.
  - Monte Carlo estimation of HSIC sensitivity indices:
    - Sample $(\boldsymbol{X}^{(i)})_{i=1}^n$ i.i.d. and compute $Y^{(i)} = f(\boldsymbol{X}^{(i)})$ for all i.
    - Build the sample $(\boldsymbol{X}_u^{(i)}, Y^{(i)})_{i=1}^n$.
    - Estimate $\widehat{\mathrm{HSIC}}_n(\boldsymbol{X}_u, \boldsymbol{X}_u)$, $\widehat{\mathrm{HSIC}}_n(Y, Y)$, $\widehat{\mathrm{HSIC}}_n(\boldsymbol{X}_u, Y)$.

Summary: sensitivity analysis

- For real-valued inputs:

	S_j	T_j	S_j^{HSIC}
screening	no	yes	yes
ranking	yes	yes	no
given sample	yes	no	yes
small sample	yes	no	yes
input dependence	no	no	yes

  - Sobol' indices $\rightarrow$ variance-based, interpretable, quantitative (sum to one), rather costly to estimate.
  - HSIC indices $\rightarrow$ general dependence, not easy to interpret (does not sum to one $\rightarrow$ HSIC-ANOVA), rather cheap to estimate, sensitive to the choice of the (correlation length of the) kernel.

[Exercise: $Y = \sin(2\pi X_1) + 2\sin(8\pi X_2)$, X_1, X_2 i.i.d. $\mathcal{U}(0,1)$
Compute S_1, S_2, T_1, T_2 and discuss S_1^{HSIC}, S_2^{HSIC} w.r.t. correlation length.]