Validating Gaussian Process Emulators

Leo Bastos

University of Sheffield

Managing Uncertainty in Complex Models

Joint work: Jeremy Oakley and Tony O'Hagan

Outline

1. Computer model
   - Definition
2. Emulation
   - Gaussian Process Emulator
   - Toy Example
3. Diagnostics and Validation
   - Numerical diagnostics
   - Graphical diagnostics
   - Examples
4. Conclusions

What is a computer model?

- **Computer model** is a mathematical representation $\eta(\cdot)$ of a complex physical system implemented in a computer.
- We need Computer models when real experiments are very expensive or even implossible to be “done” (e.g. Nuclear experiments)
- Computer models have an important role in almost all fields of science and technology
  - System Biology models (Rotavirus outbreaks)
  - Cosmological models (Galaxy formation)
  - Climate models* (Global warming)

What is a computer model?

- **Computer model** is a mathematical representation $\eta(\cdot)$ of a complex physical system implemented in a computer.
- We need Computer models when real experiments are very expensive or even implossible to be “done” (e.g. Nuclear experiments)
- Computer models have an important role in almost all fields of science and technology
  - System Biology models (Rotavirus outbreaks)
  - Cosmological models (Galaxy formation)
  - Climate models* (Global warming)

What is a computer model?

- **Computer model** is a mathematical representation $\eta(\cdot)$ of a complex physical system implemented in a computer.
- We need Computer models when real experiments are very expensive or even implossible to be "done" (e.g. Nuclear experiments)
- Computer models have an important role in almost all fields of science and technology
  - System Biology models (Rotavirus outbreaks)
  - Cosmological models (Galaxy formation)
  - Climate models* (Global warming)

The University Of Sheffield.

What is a computer model?

- **Computer model** is a mathematical representation $\eta(\cdot)$ of a complex physical system implemented in a computer.
- We need Computer models when real experiments are very expensive or even implossible to be “done” (e.g. Nuclear experiments)
- Computer models have an important role in almost all fields of science and technology
  - System Biology models (Rotavirus outbreaks)
  - Cosmological models (Galaxy formation)
  - Climate models* (Global warming)

What is a computer model?

- **Computer model** is a mathematical representation $\eta(\cdot)$ of a complex physical system implemented in a computer.
- We need Computer models when real experiments are very expensive or even implossible to be "done" (e.g. Nuclear experiments)
- Computer models have an important role in almost all fields of science and technology
  - System Biology models (Rotavirus outbreaks)
  - Cosmological models (Galaxy formation)
  - Climate models* (Global warming)

The University Of Sheffield.

What is a computer model?

- **Computer model** is a mathematical representation $\eta(\cdot)$ of a complex physical system implemented in a computer.
- We need Computer models when real experiments are very expensive or even implossible to be "done" (e.g. Nuclear experiments)
- Computer models have an important role in almost all fields of science and technology
  - System Biology models (Rotavirus outbreaks)
  - Cosmological models (Galaxy formation)
  - Climate models* (Global warming)

The University Of Sheffield.

C-GOLDSTEIN Model

- C-GOLDSTEIN is a simplified* climate model
  - Sea surface temperature
  - ocean salinity and ocean temp at different depths in the ocean
  - area of sea ice
  - thickness of sea ice
  - atmospheric CO2 concentrations
  - ...
- Large number of outputs (Both time series and field data)
- Several inputs (e.g. model resolution, initial conditions)
- Each run takes about an hour on the Linux Boxes at NOC

C-GOLDSTEIN Model

- C-GOLDSTEIN is a simplified* climate model
  - Sea surface temperature
  - ocean salinity and ocean temp at different depths in the ocean
  - area of sea ice
  - thickness of sea ice
  - atmospheric CO_2 concentrations
  - ...
- Large number of outputs (Both time series and field data)
- Several inputs (e.g. model resolution, initial conditions)
- Each run takes about an hour on the Linux Boxes at NOC

The University Of Sheffield.

C-GOLDSTEIN Model

- C-GOLDSTEIN is a simplified* climate model
  - Sea surface temperature
  - ocean salinity and ocean temp at different depths in the ocean
  - area of sea ice
  - thickness of sea ice
  - atmospheric CO2 concentrations
  - ...
- Large number of outputs (Both time series and field data)
- Several inputs (e.g. model resolution, initial conditions)
- Each run takes about an hour on the Linux Boxes at NOC

C-GOLDSTEIN Model

- C-GOLDSTEIN is a simplified* climate model
  - Sea surface temperature
  - ocean salinity and ocean temp at different depths in the ocean
  - area of sea ice
  - thickness of sea ice
  - atmospheric CO2 concentrations
  - ...
- Large number of outputs (Both time series and field data)
- Several inputs (e.g. model resolution, initial conditions)
- Each run takes about an hour on the Linux Boxes at NOC

C-GOLDSTEIN Model

- C-GOLDSTEIN is a simplified* climate model
  - Sea surface temperature
  - ocean salinity and ocean temp at different depths in the ocean
  - area of sea ice
  - thickness of sea ice
  - atmospheric CO2 concentrations
  - ...
- Large number of outputs (Both time series and field data)
- Several inputs (e.g. model resolution, initial conditions)
- Each run takes about an hour on the Linux Boxes at NOC

The University Of Sheffield.

C-GOLDSTEIN Model

- C-GOLDSTEIN is a simplified* climate model
  - Sea surface temperature
  - ocean salinity and ocean temp at different depths in the ocean
  - area of sea ice
  - thickness of sea ice
  - atmospheric CO_2 concentrations
  - ...
- Large number of outputs (Both time series and field data)
- Several inputs (e.g. model resolution, initial conditions)
- Each run takes about an hour on the Linux Boxes at NOC

C-GOLDSTEIN Model

- C-GOLDSTEIN is a simplified* climate model
  - Sea surface temperature
  - ocean salinity and ocean temp at different depths in the ocean
  - area of sea ice
  - thickness of sea ice
  - atmospheric CO2 concentrations
  - ...
- Large number of outputs (Both time series and field data)
- Several inputs (e.g. model resolution, initial conditions)
- Each run takes about an hour on the Linux Boxes at NOC

The University Of Sheffield.

C-GOLDSTEIN Model

- C-GOLDSTEIN is a simplified* climate model
    - Sea surface temperature
    - ocean salinity and ocean temp at different depths in the ocean
    - area of sea ice
    - thickness of sea ice
    - atmospheric CO2 concentrations
    - ...
- Large number of outputs (Both time series and field data)
- Several inputs (e.g. model resolution, initial conditions)
- Each run takes about an hour on the Linux Boxes at NOC

The University Of Sheffield.

C-GOLDSTEIN Model

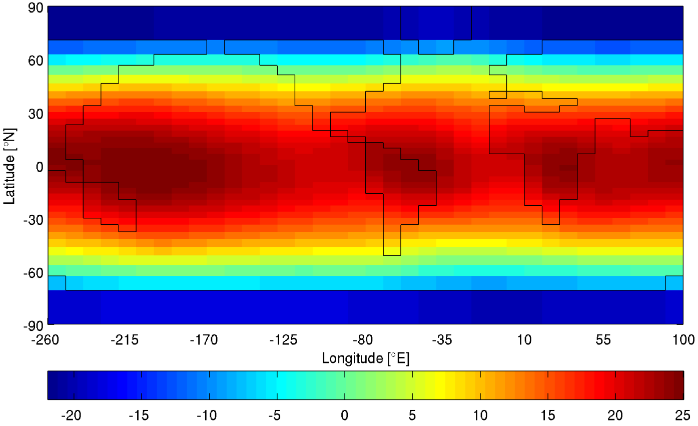
- C-GOLDSTEIN is a simplified* climate model
  - Sea surface temperature
  - ocean salinity and ocean temp at different depths in the ocean
  - area of sea ice
  - thickness of sea ice
  - atmospheric CO2 concentrations
  - ...
- Large number of outputs (Both time series and field data)
- Several inputs (e.g. model resolution, initial conditions)
- Each run takes about an hour on the Linux Boxes at NOC

The University Of Sheffield.

C-GOLDSTEIN Model

- C-GOLDSTEIN is a simplified* climate model
  - Sea surface temperature
  - ocean salinity and ocean temp at different depths in the ocean
  - area of sea ice
  - thickness of sea ice
  - atmospheric CO2 concentrations
  - ...
- Large number of outputs (Both time series and field data)
- Several inputs (e.g. model resolution, initial conditions)
- Each run takes about an hour on the Linux Boxes at NOC

The University Of Sheffield.


90
60
30
0
-30
-60
-90
Latitude [°N]
-260
-215
-170
-125
-80
-35
10
55
100
Longitude [°E]
-20
-15
-10
-5
0
5
10
15
20
25

Computer models can be very expensive

IBM supercomputers used for climate and weather forecasts

- One single run of the computer model can take a lot of time
- Most of analyses need several runs

The University Of Sheffield.

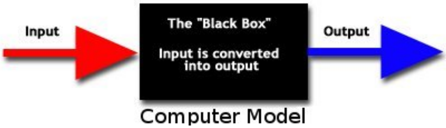
Computer models can be very expensive

IBM supercomputers used for climate and weather forecasts

- One single run of the computer model can take a lot of time
- Most of analyses need several runs

The University Of Sheffield.

- $\eta(\cdot)$ is considered an unknown function

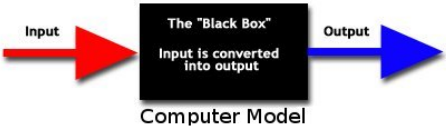


- **Emulator** is a predictive function for the computer model outputs
- Assumptions for the computer model:
  - Deterministic single-output model $\eta(\cdot): \mathcal{X} \subset \mathbb{R}^p \to \mathbb{R}$
  - Relatively "Smooth" function
- **Statistical Emulator** is an stochastic representation of our judgements about the computer model $\eta(\cdot)$.

The University Of Sheffield.

Emulating a computer model

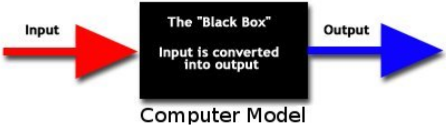
- $\eta(\cdot)$ is considered an unknown function



- **Emulator** is a predictive function for the computer model outputs
- Assumptions for the computer model:
  - Deterministic single-output model $\eta(\cdot)$ $\quad \eta : \mathcal{X} \in \Re^p \rightarrow \Re$
  - Relatively "Smooth" function
- **Statistical Emulator** is an stochastic representation of our judgements about the computer model $\eta(\cdot)$.

Emulating a computer model

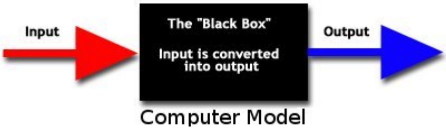
- $\eta(\cdot)$ is considered an unknown function



- **Emulator** is a predictive function for the computer model outputs
- Assumptions for the computer model:
  - Deterministic single-output model $\eta(\cdot)$ $\quad \eta : \mathcal{X} \in \Re^p \rightarrow \Re$
  - Relatively "Smooth" function
- **Statistical Emulator** is an stochastic representation of our judgements about the computer model $\eta(\cdot)$.

Emulating a computer model

- $\eta(\cdot)$ is considered an unknown function

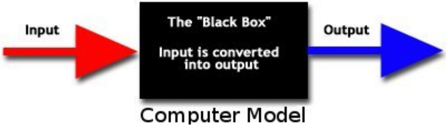


- **Emulator** is a predictive function for the computer model outputs
- Assumptions for the computer model:
  - Deterministic single-output model $\eta(\cdot)$ $\quad \eta : \mathcal{X} \in \Re^p \rightarrow \Re$
  - Relatively "Smooth" function
- **Statistical Emulator** is an stochastic representation of our judgements about the computer model $\eta(\cdot)$.

The University Of Sheffield.

Emulating a computer model

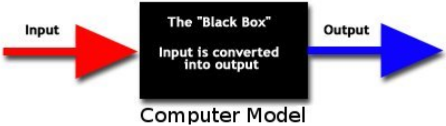
- $\eta(\cdot)$ is considered an unknown function



- **Emulator** is a predictive function for the computer model outputs
- Assumptions for the computer model:
  - Deterministic single-output model $\eta(\cdot)$ $\eta : \mathcal{X} \in \Re^p \rightarrow \Re$
  - Relatively "Smooth" function
- **Statistical Emulator** is an stochastic representation of our judgements about the computer model $\eta(\cdot)$.

Emulating a computer model

- $\eta(\cdot)$ is considered an unknown function



- **Emulator** is a predictive function for the computer model outputs
- Assumptions for the computer model:
  - Deterministic single-output model $\eta(\cdot)$ $\quad \eta : \mathcal{X} \in \Re^p \rightarrow \Re$
  - Relatively "Smooth" function
- **Statistical Emulator** is an stochastic representation of our judgements about the computer model $\eta(\cdot)$.

The University Of Sheffield.

Gaussian Process Emulator

- **Gaussian process emulator**:

$$\eta(\cdot)|\beta, \sigma^2, \psi \sim GP\left(m_0(\cdot), V_0(\cdot,\cdot)\right),$$

  where

$$\begin{aligned} m_0(\mathbf{x}) &= h(\mathbf{x})^T\beta \\ V_0(\mathbf{x}, \mathbf{x}') &= \sigma^2 C(\mathbf{x}, \mathbf{x}'; \psi) \end{aligned}$$

- Prior distribution for (β, σ^2, ψ)
- Conditioning on some training data

$$y_k = \eta(\mathbf{x_k}), \quad k = 1, \ldots, n$$

Gaussian Process Emulator

- **Gaussian process emulator**:

$$\eta(\cdot)|\beta, \sigma^2, \psi \sim GP\left(m_0(\cdot), V_0(\cdot, \cdot)\right),$$

where

$$\begin{aligned} m_0(\mathbf{x}) &= h(\mathbf{x})^T \beta \\ V_0(\mathbf{x}, \mathbf{x}') &= \sigma^2 C(\mathbf{x}, \mathbf{x}'; \psi) \end{aligned}$$

- Prior distribution for (β, σ^2, ψ)
- Conditioning on some training data

$$y_k = \eta(\mathbf{x_k}), \quad k = 1, \ldots, n$$

Gaussian Process Emulator

- **Gaussian process emulator**:

$$\eta(\cdot)|\beta, \sigma^2, \psi \sim GP\left(m_0(\cdot), V_0(\cdot, \cdot)\right),$$

where

$$\begin{aligned} m_0(\mathbf{x}) &= h(\mathbf{x})^T \beta \\ V_0(\mathbf{x}, \mathbf{x}') &= \sigma^2 C(\mathbf{x}, \mathbf{x}'; \psi) \end{aligned}$$

- Prior distribution for (β, σ^2, ψ)
- Conditioning on some training data

$$y_k = \eta(\mathbf{x_k}), \quad k = 1, \ldots, n$$

The University Of Sheffield.

- **Predictive Gaussian Process Emulator**

$$\eta(\cdot)|\mathbf{y}, \mathbf{X}, \psi \sim \text{Student-Process}\left(n-q, m_1(\cdot), V_1(\cdot,\cdot)\right),$$

where

$$\begin{aligned} m_1(x) &= h(x)^T\widehat{\beta} + t(x)^T\mathbf{A}^{-1}(\mathbf{y} - H\widehat{\beta}), \\ V_1(x,x') &= \hat{\sigma}^2\Big[C(x,x';\psi) - t(x)^T\mathbf{A}^{-1}t(x') + \Big(h(x) - t(x)^T\mathbf{A}^{-1}H\Big) \\ &\times (H^T\mathbf{A}^{-1}H)^{-1}\Big(h(x') - t(x')^T\mathbf{A}^{-1}H\Big)^T\Big]. \end{aligned}$$

Toy Example

- $\eta(\cdot)$ is a two-dimensional known function
- GP emulator:

$$\eta(\cdot)|\beta, \sigma^2, \psi \sim GP\left(h(\cdot)^T\beta, \sigma^2 C(\mathbf{x}, \mathbf{x}'; \psi)\right),$$

  - $h(\mathbf{x}) = (1, \mathbf{x})^T$
  - $C(\mathbf{x}, \mathbf{x}') = \exp\left[-\sum_k \left(\frac{x_k - x'_k}{\psi_k}\right)^2\right]$
- $p(\beta, \sigma^2, \psi) \propto \sigma^{-2}$

Toy Example

- $\eta(\cdot)$ is a two-dimensional known function
- GP emulator:

$$\eta(\cdot)|\beta, \sigma^2, \psi \sim GP\left(h(\cdot)^T\beta, \sigma^2 C(\mathbf{x}, \mathbf{x}'; \psi)\right),$$

  - $h(\mathbf{x}) = (1, \mathbf{x})^T$
  - $C(\mathbf{x}, \mathbf{x}') = \exp\left[-\sum_k \left(\frac{\mathbf{x}_k - \mathbf{x}'_k}{\psi_k}\right)^2\right]$
- $p(\beta, \sigma^2, \psi) \propto \sigma^{-2}$

The University Of Sheffield.

Toy Example

- $\eta(\cdot)$ is a two-dimensional known function
- GP emulator:

$$\eta(\cdot)|\beta, \sigma^2, \psi \sim GP\left(h(\cdot)^T\beta, \sigma^2 C(\mathbf{x}, \mathbf{x}'; \psi)\right),$$

  - $h(\mathbf{x}) = (1, \mathbf{x})^T$
  - $C(\mathbf{x}, \mathbf{x}') = \exp\left[-\sum_k \left(\frac{\mathbf{x}_k - \mathbf{x}'_k}{\psi_k}\right)^2\right]$
- $p(\beta, \sigma^2, \psi) \propto \sigma^{-2}$

Toy Example

- $\eta(\cdot)$ is a two-dimensional known function
- GP emulator:

$$\eta(\cdot)|\beta, \sigma^2, \psi \sim GP\left(h(\cdot)^T\beta, \sigma^2 C(\mathbf{x}, \mathbf{x}'; \psi)\right),$$

  - $h(\mathbf{x}) = (1, \mathbf{x})^T$
  - $C(\mathbf{x}, \mathbf{x}') = \exp\left[-\sum_k \left(\frac{\mathbf{x}_k - \mathbf{x}'_k}{\psi_k}\right)^2\right]$

- $p(\beta, \sigma^2, \psi) \propto \sigma^{-2}$

The University Of Sheffield.

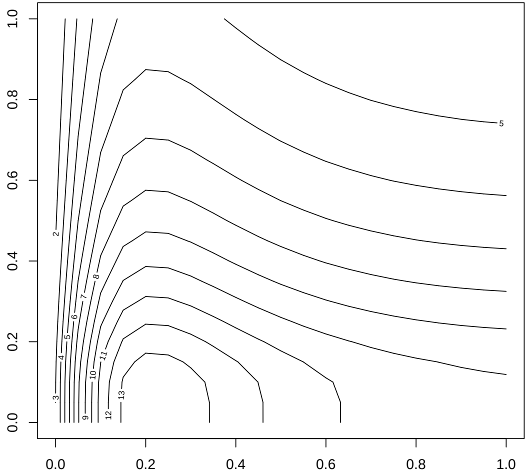
Toy Example

- $\eta(\cdot)$ is a two-dimensional known function
- GP emulator:

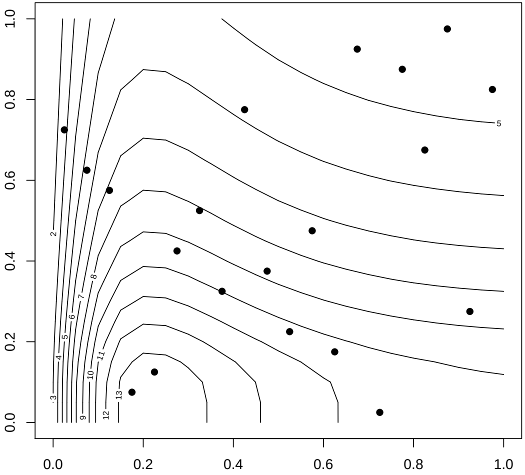
$$\eta(\cdot)|\beta, \sigma^2, \psi \sim GP\left(h(\cdot)^T\beta, \sigma^2 C(\mathbf{x}, \mathbf{x}'; \psi)\right),$$

  - $h(\mathbf{x}) = (1, \mathbf{x})^T$
  - $C(\mathbf{x}, \mathbf{x}') = \exp\left[-\sum_k \left(\frac{\mathbf{x}_k - \mathbf{x}'_k}{\psi_k}\right)^2\right]$
- $p(\beta, \sigma^2, \psi) \propto \sigma^{-2}$

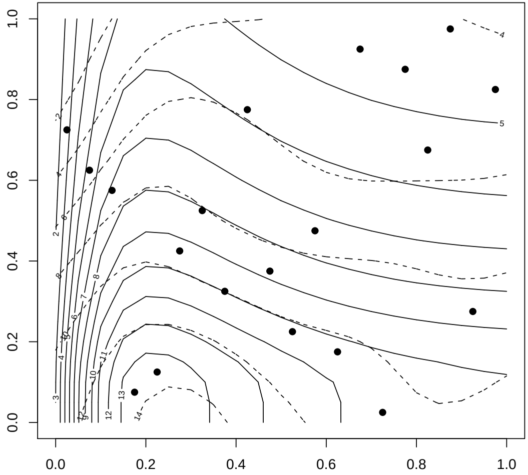
Toy example



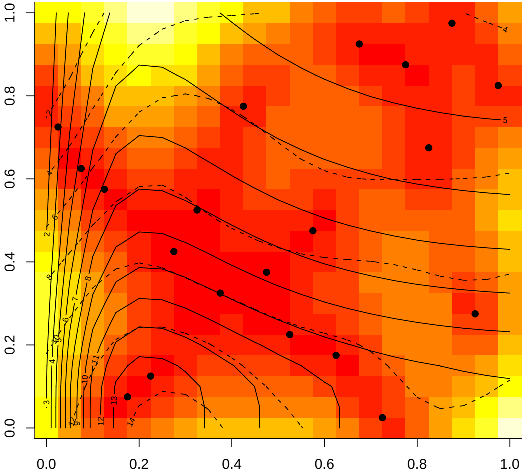
Toy example



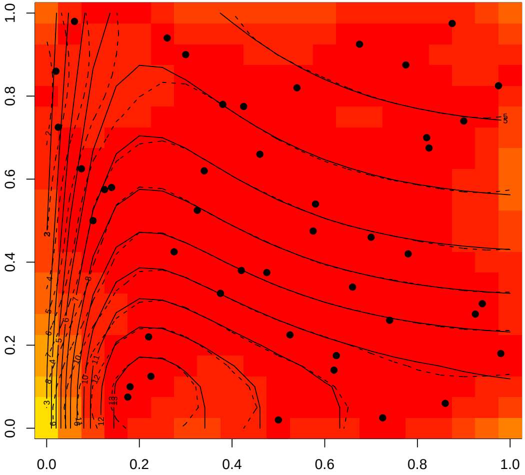
Toy example



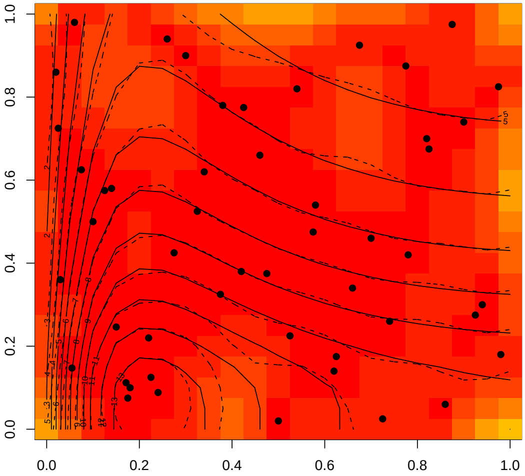
Toy example



Toy example



Toy example



MUCM Topics

- Design for Computer models
- Emulation (Multiple output emulation, Dynamic emulation)
- UA/SA - Uncertainty and Sensitivity Analyses
- Calibration (Bayes Linear and Full Bayesian approaches)
- Diagnostics and Validation

The University Of Sheffield.

MUCM Topics

- Design for Computer models
- Emulation (Multiple output emulation, Dynamic emulation)
- UA/SA - Uncertainty and Sensitivity Analyses
- Calibration (Bayes Linear and Full Bayesian approaches)
- Diagnostics and Validation

MUCM Topics

- Design for Computer models
- Emulation (Multiple output emulation, Dynamic emulation)
- UA/SA - Uncertainty and Sensitivity Analyses
- Calibration (Bayes Linear and Full Bayesian approaches)
- Diagnostics and Validation

MUCM Topics

- Design for Computer models
- Emulation (Multiple output emulation, Dynamic emulation)
- UA/SA - Uncertainty and Sensitivity Analyses
- Calibration (Bayes Linear and Full Bayesian approaches)
- Diagnostics and Validation

MUCM Topics

- Design for Computer models
- Emulation (Multiple output emulation, Dynamic emulation)
- UA/SA - Uncertainty and Sensitivity Analyses
- Calibration (Bayes Linear and Full Bayesian approaches)
- Diagnostics and Validation

Diagnostics and Validation

- Every emulator should be validated
- Non-valid emulators can induce wrong conclusions
- There is little research into validating emulators
- **Validation** generally means: "*the emulator predictions are close enough to the simulator outputs*".
- We want to take account all the uncertainty associated with the emulator.
- "Do the choices that I have made, based on my knowledge of this simulator, appear to be consistent with the observations?"
- Choices for the Gaussian process emulator:
  - Normality
  - Stationarity
  - Correlation parameters

The University Of Sheffield.

Diagnostics and Validation

- Every emulator should be validated
- Non-valid emulators can induce wrong conclusions
- There is little research into validating emulators
- **Validation** generally means: "*the emulator predictions are close enough to the simulator outputs*".
- We want to take account all the uncertainty associated with the emulator.
- "Do the choices that I have made, based on my knowledge of this simulator, appear to be consistent with the observations?"
- Choices for the Gaussian process emulator:
  1. Normality
  2. Stationarity
  3. Correlation parameters

Diagnostics and Validation

- Every emulator should be validated
- Non-valid emulators can induce wrong conclusions
- There is little research into validating emulators
- **Validation** generally means: "*the emulator predictions are close enough to the simulator outputs*".
- We want to take account all the uncertainty associated with the emulator.
- "Do the choices that I have made, based on my knowledge of this simulator, appear to be consistent with the observations?"
- Choices for the Gaussian process emulator:
  - Normality
  - Stationarity
  - Correlation parameters

Diagnostics and Validation

- Every emulator should be validated
- Non-valid emulators can induce wrong conclusions
- There is little research into validating emulators
- **Validation** generally means: "*the emulator predictions are close enough to the simulator outputs*".
- We want to take account all the uncertainty associated with the emulator.
- "Do the choices that I have made, based on my knowledge of this simulator, appear to be consistent with the observations?"
- Choices for the Gaussian process emulator:
  - Normality
  - Stationarity
  - Correlation parameters

Diagnostics and Validation

- Every emulator should be validated
- Non-valid emulators can induce wrong conclusions
- There is little research into validating emulators
- **Validation** generally means: "*the emulator predictions are close enough to the simulator outputs*".
- We want to take account all the uncertainty associated with the emulator.
- "Do the choices that I have made, based on my knowledge of this simulator, appear to be consistent with the observations?"
- Choices for the Gaussian process emulator:
  - Normality
  - Stationarity
  - Correlation parameters

The University Of Sheffield.

Diagnostics and Validation

- Every emulator should be validated
- Non-valid emulators can induce wrong conclusions
- There is little research into validating emulators
- **Validation** generally means: "*the emulator predictions are close enough to the simulator outputs*".
- We want to take account all the uncertainty associated with the emulator.
- "Do the choices that I have made, based on my knowledge of this simulator, appear to be consistent with the observations?"
- Choices for the Gaussian process emulator:
  - Normality
  - Stationarity
  - Correlation parameters

Diagnostics and Validation

- Every emulator should be validated
- Non-valid emulators can induce wrong conclusions
- There is little research into validating emulators
- **Validation** generally means: "*the emulator predictions are close enough to the simulator outputs*".
- We want to take account all the uncertainty associated with the emulator.
- "Do the choices that I have made, based on my knowledge of this simulator, appear to be consistent with the observations?"
- Choices for the Gaussian process emulator:
  - Normality
  - Stationarity
  - Correlation parameters

Diagnostics and Validation

- Every emulator should be validated
- Non-valid emulators can induce wrong conclusions
- There is little research into validating emulators
- **Validation** generally means: "*the emulator predictions are close enough to the simulator outputs*".
- We want to take account all the uncertainty associated with the emulator.
- "Do the choices that I have made, based on my knowledge of this simulator, appear to be consistent with the observations?"
- Choices for the Gaussian process emulator:
  - Normality
  - Stationarity
  - Correlation parameters

Diagnostics and Validation

- Every emulator should be validated
- Non-valid emulators can induce wrong conclusions
- There is little research into validating emulators
- **Validation** generally means: "*the emulator predictions are close enough to the simulator outputs*".
- We want to take account all the uncertainty associated with the emulator.
- "Do the choices that I have made, based on my knowledge of this simulator, appear to be consistent with the observations?"
- Choices for the Gaussian process emulator:
  - Normality
  - Stationarity
  - Correlation parameters

Diagnostics and Validation

- Every emulator should be validated
- Non-valid emulators can induce wrong conclusions
- There is little research into validating emulators
- **Validation** generally means: "*the emulator predictions are close enough to the simulator outputs*".
- We want to take account all the uncertainty associated with the emulator.
- "Do the choices that I have made, based on my knowledge of this simulator, appear to be consistent with the observations?"
- Choices for the Gaussian process emulator:
  - Normality
  - Stationarity
  - Correlation parameters

Validating a GP Emulator

- Our diagnostics should be based on a set of new runs of the simulator
  - Why? Because predictions at observed input points are perfect.
  - **Validation data** $(\mathbf{y}^*, \mathbf{X}^*) \quad : \quad y_k^* = \eta(\mathbf{x_k}^*), \quad k = 1, \ldots, m$
- Simulator and the predictive emulator outputs are compared
  - **Numerical diagnostics**
  - **Graphical diagnostics**

Validating a GP Emulator

- Our diagnostics should be based on a set of new runs of the simulator
  - Why? Because predictions at observed input points are perfect.
  - **Validation data** $(\mathbf{y}^*, \mathbf{X}^*) \quad : \quad y_k^* = \eta(\mathbf{x_k}^*), \quad k = 1, \ldots, m$
- Simulator and the predictive emulator outputs are compared
  - **Numerical diagnostics**
  - **Graphical diagnostics**

The University Of Sheffield.

Validating a GP Emulator

- Our diagnostics should be based on a set of new runs of the simulator
  - Why? Because predictions at observed input points are perfect.
  - **Validation data** $(\mathbf{y}^*, \mathbf{X}^*) \quad : \quad y_k^* = \eta(\mathbf{x_k}^*), \quad k = 1, \ldots, m$

- Simulator and the predictive emulator outputs are compared
  - **Numerical diagnostics**
  - **Graphical diagnostics**

Validating a GP Emulator

- Our diagnostics should be based on a set of new runs of the simulator
  - Why? Because predictions at observed input points are perfect.
  - **Validation data** $(\mathbf{y}^*, \mathbf{X}^*) \quad : \quad y_k^* = \eta(\mathbf{x_k}^*), \quad k = 1, \ldots, m$
- Simulator and the predictive emulator outputs are compared
  - Numerical diagnostics
  - Graphical diagnostics

The University Of Sheffield.

Validating a GP Emulator

- Our diagnostics should be based on a set of new runs of the simulator
  - Why? Because predictions at observed input points are perfect.
  - **Validation data** $(\mathbf{y}^*, \mathbf{X}^*) \quad : \quad y_k^* = \eta(\mathbf{x_k}^*), \quad k = 1, \ldots, m$
- Simulator and the predictive emulator outputs are compared
  - **Numerical diagnostics**
  - Graphical diagnostics

The University Of Sheffield.

- Our diagnostics should be based on a set of new runs of the simulator
  - Why? Because predictions at observed input points are perfect.
  - **Validation data** $(\mathbf{y}^*, \mathbf{X}^*) \quad : \quad y_k^* = \eta(\mathbf{x_k}^*), \quad k = 1, \ldots, m$
- Simulator and the predictive emulator outputs are compared
  - **Numerical diagnostics**
  - **Graphical diagnostics**

Numerical diagnostics

Individual predictive errors

$$D_i^I(\mathbf{y}^*) = \frac{(y_i^* - m_1(\mathbf{x}_i^*))}{\sqrt{V_1(\mathbf{x}_i^*, \mathbf{x}_i^*)}}$$

However, the $D^I(\mathbf{y}^*)$s are correlated:

$$D^I(\eta(\mathbf{X}^*)) \sim \text{Student-t}_m(n-q, \mathbf{0}, C_1(\mathbf{X}^*))$$

Mahalanobis distance

$$D_{MD}(\mathbf{y}^*) = (\mathbf{y}^* - m_1(\mathbf{X}^*))^T V_1(\mathbf{X}^*)^{-1} (\mathbf{y}^* - m_1(\mathbf{X}^*))$$

$$\frac{(n-q)}{m(n-q-2)} D_{MD}(\eta(\mathbf{X}^*)) \sim F_{m,n-q}$$

Numerical diagnostics

Individual predictive errors

$$D_i^I(\mathbf{y}^*) = \frac{(y_i^* - m_1(\mathbf{x}_i^*))}{\sqrt{V_1(\mathbf{x}_i^*, \mathbf{x}_i^*)}}$$

However, the $D^I(\mathbf{y}^*)$s are correlated:

$$D^I(\eta(\mathbf{X}^*)) \sim \text{Student-t}_m(n-q, \mathbf{0}, C_1(\mathbf{X}^*))$$

Mahalanobis distance

$$D_{MD}(\mathbf{y}^*) = (\mathbf{y}^* - m_1(\mathbf{X}^*))^T V_1(\mathbf{X}^*)^{-1} (\mathbf{y}^* - m_1(\mathbf{X}^*))$$

$$\frac{(n-q)}{m(n-q-2)} D_{MD}(\eta(\mathbf{X}^*)) \sim F_{m,n-q}$$

The University Of Sheffield.

Individual predictive errors

$$D_i^I(\mathbf{y}^*) = \frac{(y_i^* - m_1(\mathbf{x}_i^*))}{\sqrt{V_1(\mathbf{x}_i^*, \mathbf{x}_i^*)}}$$

However, the $D^I(\mathbf{y}^*)$s are correlated:

$$D^I(\eta(\mathbf{X}^*)) \sim \text{Student-t}_m(n-q, \mathbf{0}, C_1(\mathbf{X}^*))$$

Mahalanobis distance

$$D_{MD}(\mathbf{y}^*) = (\mathbf{y}^* - m_1(\mathbf{X}^*))^T V_1(\mathbf{X}^*)^{-1} (\mathbf{y}^* - m_1(\mathbf{X}^*))$$

$$\frac{(n-q)}{m(n-q-2)} D_{MD}(\eta(\mathbf{X}^*)) \sim F_{m,n-q}$$

Numerical diagnostics

Individual predictive errors

$$D_i^I(\mathbf{y}^*) = \frac{(y_i^* - m_1(\mathbf{x}_i^*))}{\sqrt{V_1(\mathbf{x}_i^*, \mathbf{x}_i^*)}}$$

However, the $D^I(\mathbf{y}^*)$s are correlated:

$$D^I(\eta(\mathbf{X}^*)) \sim \text{Student-t}_m(n-q, \mathbf{0}, C_1(\mathbf{X}^*))$$

Mahalanobis distance

$$D_{MD}(\mathbf{y}^*) = (\mathbf{y}^* - m_1(\mathbf{X}^*))^T V_1(\mathbf{X}^*)^{-1} (\mathbf{y}^* - m_1(\mathbf{X}^*))$$

$$\frac{(n-q)}{m(n-q-2)} D_{MD}(\eta(\mathbf{X}^*)) \sim F_{m,n-q}$$

Numerical diagnostics

Pivoted Cholesky errors

$$D^{PC}(\mathbf{y}^*) \;=\; (\mathbf{G}^{-1})^T(\mathbf{y}^* - m_1(\mathbf{X}^*))$$

where $V_1(\mathbf{X}^*) = \mathbf{G}^\mathbf{T}\mathbf{G}$, and $\mathbf{G} = \mathbf{P}\mathbf{R}^T$.

Properties:

- $D^{PC}(\mathbf{y}^*)^T D^{PC}(\mathbf{y}^*) = D_{MD}(\mathbf{y}^*)$
- $Var(D^{PC}(\mathbf{y}(\mathbf{X}^*))) = I$
- Invariant to the data order
- Pivoting order given by $\mathbf{P}$ has an intuitive explanation
- Each $D^{PC}(\mathbf{y}^*)$ associated with a validation element

Pivoted Cholesky errors

$$D^{PC}(\mathbf{y}^*) = (\mathbf{G}^{-1})^T(\mathbf{y}^* - m_1(\mathbf{X}^*))$$

where $V_1(\mathbf{X}^*) = \mathbf{G}^\mathbf{T}\mathbf{G}$, and $\mathbf{G} = \mathbf{PR}^T$.

Properties:

- $D^{PC}(\mathbf{y}^*)^T D^{PC}(\mathbf{y}^*) = D_{MD}(\mathbf{y}^*)$
- $Var(D^{PC}(\eta(\mathbf{X})) = \mathcal{I}$
- Invariant to the data order
- Pivoting order given by **P** has an intuitive explanation
- Each $D^{PC}(\mathbf{y}^*)$ associated with a validation element

Numerical diagnostics

Pivoted Cholesky errors

$$D^{PC}(\mathbf{y}^*) = (\mathbf{G}^{-1})^T(\mathbf{y}^* - m_1(\mathbf{X}^*))$$

where $V_1(\mathbf{X}^*) = \mathbf{G}^\mathbf{T}\mathbf{G}$, and $\mathbf{G} = \mathbf{P}\mathbf{R}^T$.

Properties:

- $D^{PC}(\mathbf{y}^*)^T D^{PC}(\mathbf{y}^*) = D_{MD}(\mathbf{y}^*)$
- $Var(D^{PC}(\eta(\mathbf{X})) = \mathcal{I}$
- Invariant to the data order
- Pivoting order given by $\mathbf{P}$ has an intuitive explanation
- Each $D^{PC}(\mathbf{y}^*)$ associated with a validation element

Pivoted Cholesky errors

$$D^{PC}(\mathbf{y}^*) = (\mathbf{G}^{-1})^T(\mathbf{y}^* - m_1(\mathbf{X}^*))$$

where $V_1(\mathbf{X}^*) = \mathbf{G}^\mathbf{T}\mathbf{G}$, and $\mathbf{G} = \mathbf{P}\mathbf{R}^T$.

Properties:

- $D^{PC}(\mathbf{y}^*)^T D^{PC}(\mathbf{y}^*) = D_{MD}(\mathbf{y}^*)$
- $Var(D^{PC}(\eta(\mathbf{X})) = \mathcal{I}$
- Invariant to the data order
- Pivoting order given by **P** has an intuitive explanation
- Each $D^{PC}(\mathbf{y}^*)$ associated with a validation element

Pivoted Cholesky errors

$$D^{PC}(\mathbf{y}^*) \;=\; (\mathbf{G}^{-1})^T(\mathbf{y}^* - m_1(\mathbf{X}^*))$$

where $V_1(\mathbf{X}^*) = \mathbf{G^TG}$, and $\mathbf{G} = \mathbf{PR}^T$.

Properties:

- $D^{PC}(\mathbf{y}^*)^T D^{PC}(\mathbf{y}^*) = D_{MD}(\mathbf{y}^*)$
- $Var(D^{PC}(\eta(\mathbf{X})) = \mathcal{I}$
- Invariant to the data order
- Pivoting order given by **P** has an intuitive explanation
- Each $D^{PC}(\mathbf{y}^*)$ associated with a validation element

Numerical diagnostics

Pivoted Cholesky errors

$$D^{PC}(\mathbf{y}^*) \;=\; (\mathbf{G}^{-1})^T(\mathbf{y}^* - m_1(\mathbf{X}^*))$$

where $V_1(\mathbf{X}^*) = \mathbf{G^T G}$, and $\mathbf{G} = \mathbf{PR}^T$.

Properties:

- $D^{PC}(\mathbf{y}^*)^T D^{PC}(\mathbf{y}^*) = D_{MD}(\mathbf{y}^*)$
- $Var(D^{PC}(\eta(\mathbf{X})) = \mathcal{I}$
- Invariant to the data order
- Pivoting order given by **P** has an intuitive explanation
- Each $D^{PC}(\mathbf{y}^*)$ associated with a validation element

Numerical diagnostics

Pivoted Cholesky errors

$$D^{PC}(\mathbf{y}^*) \quad = \quad (\mathbf{G}^{-1})^T(\mathbf{y}^* - m_1(\mathbf{X}^*))$$

where $V_1(\mathbf{X}^*) = \mathbf{G}^\mathbf{T}\mathbf{G}$, and $\mathbf{G} = \mathbf{PR}^T$.

Properties:

- $D^{PC}(\mathbf{y}^*)^T D^{PC}(\mathbf{y}^*) = D_{MD}(\mathbf{y}^*)$
- $Var(D^{PC}(\eta(\mathbf{X})) = \mathcal{I}$
- Invariant to the data order
- Pivoting order given by **P** has an intuitive explanation
- Each $D^{PC}(\mathbf{y}^*)$ associated with a validation element

Graphical diagnostics

- Some possible Graphical diagnostics:
  - **Individual errors against emulator's predictions**
    Problems on mean function, non-stationarity
  - **Errors againts the pivoting order**
    Poor estimation of the variance, correlation parameters
  - **QQ-plots of the uncorrelated standardized errors**
    Non-normality, Local fitting problems or non-stationarity
  - **Individual or (pivoted) Cholesky errors against inputs**
    Non-stationarity, pattern not included in the mean function

Graphical diagnostics

- Some possible Graphical diagnostics:
  - **Individual errors against emulator's predictions**
    Problems on mean function, non-stationarity
  - **Errors againts the pivoting order**
    Poor estimation of the variance, correlation parameters
  - **QQ-plots of the uncorrelated standardized errors**
    Non-normality, Local fitting problems or non-stationarity
  - **Individual or (pivoted) Cholesky errors against inputs**
    Non-stationarity, pattern not included in the mean function

The University Of Sheffield.

Graphical diagnostics

- Some possible Graphical diagnostics:
  - **Individual errors against emulator's predictions**
    Problems on mean function, non-stationarity
  - **Errors againts the pivoting order**
    Poor estimation of the variance, correlation parameters
  - **QQ-plots of the uncorrelated standardized errors**
    Non-normality, Local fitting problems or non-stationarity
  - **Individual or (pivoted) Cholesky errors against inputs**
    Non-stationarity, pattern not included in the mean function

The University Of Sheffield.

Graphical diagnostics

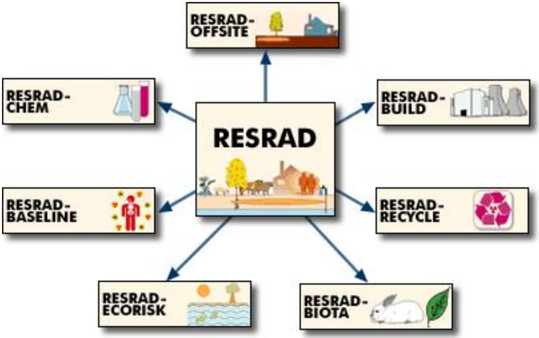
- Some possible Graphical diagnostics:
  - **Individual errors against emulator's predictions**
    Problems on mean function, non-stationarity
  - **Errors againts the pivoting order**
    Poor estimation of the variance, correlation parameters
  - **QQ-plots of the uncorrelated standardized errors**
    Non-normality, Local fitting problems or non-stationarity
  - **Individual or (pivoted) Cholesky errors against inputs**
    Non-stationarity, pattern not included in the mean function

The University Of Sheffield.

Graphical diagnostics

- Some possible Graphical diagnostics:
  - **Individual errors against emulator's predictions**
    Problems on mean function, non-stationarity
  - **Errors againts the pivoting order**
    Poor estimation of the variance, correlation parameters
  - **QQ-plots of the uncorrelated standardized errors**
    Non-normality, Local fitting problems or non-stationarity
  - **Individual or (pivoted) Cholesky errors against inputs**
    Non-stationarity, pattern not included in the mean function

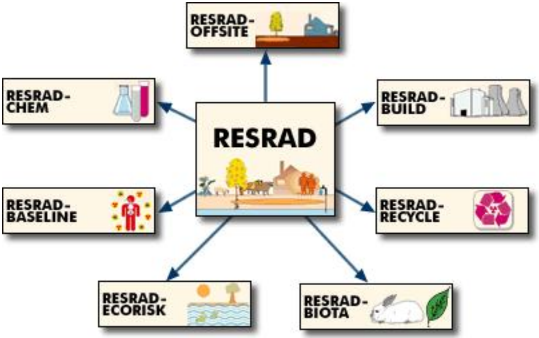
Example: Nuclear Waste Repository



Source: *http://web.ead.anl.gov/resrad/*

- RESRAD is a computer model designed to estimate radiation doses and risks from RESidual RADioactive materials.
- Output: 10,000 year time series of the release of contamination in drinking water (in millirems)

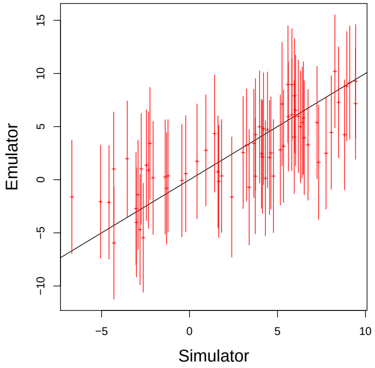
Example: Nuclear Waste Repository



Source: *http://web.ead.anl.gov/resrad/*

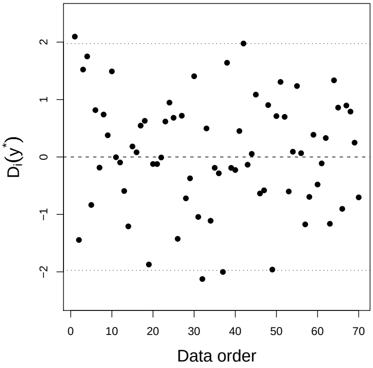
- RESRAD is a computer model designed to estimate radiation doses and risks from RESidual RADioactive materials.
- Output: 10,000 year time series of the release of contamination in drinking water (in millirems)

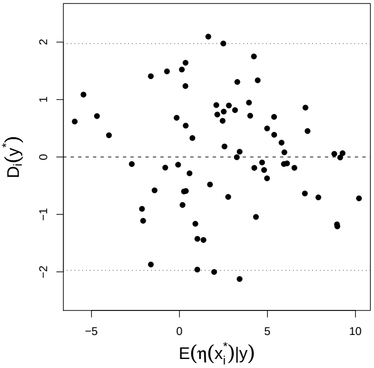
Example: Nuclear Waste Repository



- Output - Log of maximal dose of radiation in drinking water
- 27 inputs
- Training data: $n = 190^{*}(900)$
- Validation data: $m = 69^{*}(300)$

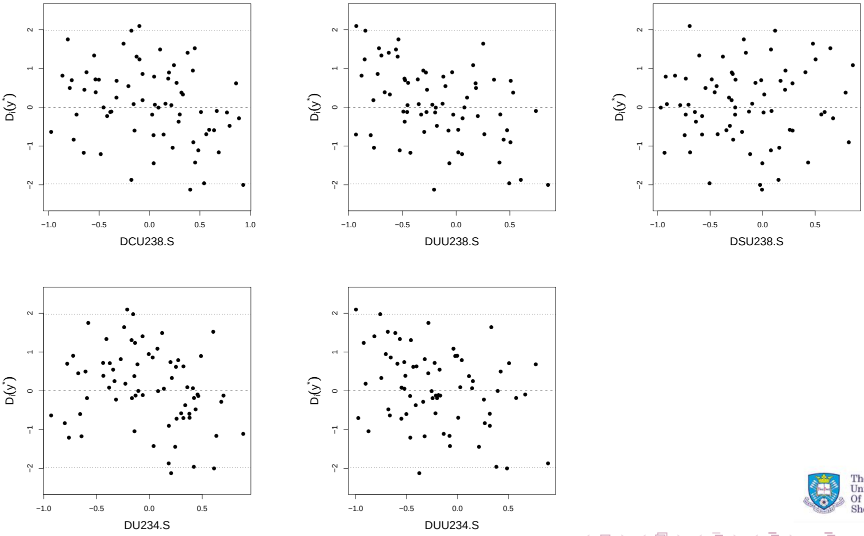
Graphical Diagnostics: Individual errors



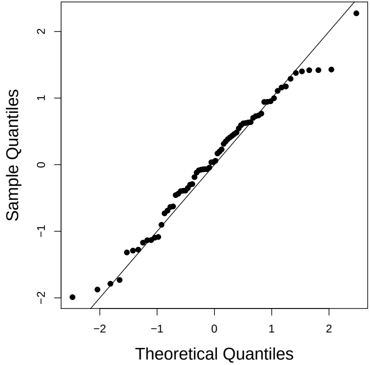


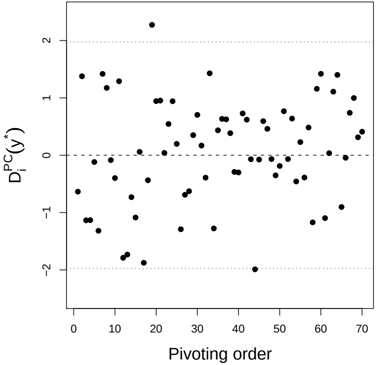
The University Of Sheffield.

Graphical Diagnostics: Individual errors



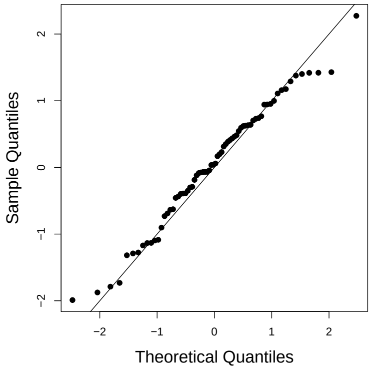
Graphical Diagnostics: Correlated errors

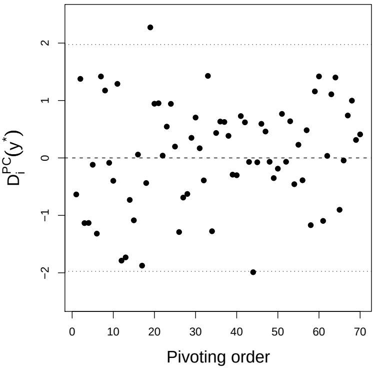




$D_{MD}(\mathbf{y}^*) = 58.96$ and the 95% CI is (47.13; 104.70)

Graphical Diagnostics: Correlated errors





$D_{MD}(\mathbf{y}^*) = 58.96$ and the 95% CI is (47.13; 104.70)

Example: Nilson-Kuusk model

- Nilson-Kuusk model is a plant canopy reflectance model.
- For interpretation of remote sensoring data
- For determination of agronomical and phytometric parameters
  - The Nilson-Kuusk model is a single output model with 5 inputs
  - The training data contains 150 points
  - The validation data contains 100 points

Example: Nilson-Kuusk model

- Nilson-Kuusk model is a plant canopy reflectance model.
- For interpretation of remote sensoring data
- For determination of agronomical and phytometric parameters
  - The Nilson-Kuusk model is a single output model with 5 inputs
  - The training data contains 150 points
  - The validation data contains 100 points

The University Of Sheffield.

Example: Nilson-Kuusk model

- Nilson-Kuusk model is a plant canopy reflectance model.
- For interpretation of remote sensoring data
- For determination of agronomical and phytometric parameters
  - The Nilson-Kuusk model is a single output model with 5 inputs
  - The training data contains 150 points
  - The validation data contains 100 points

The University Of Sheffield.

Example: Nilson-Kuusk model

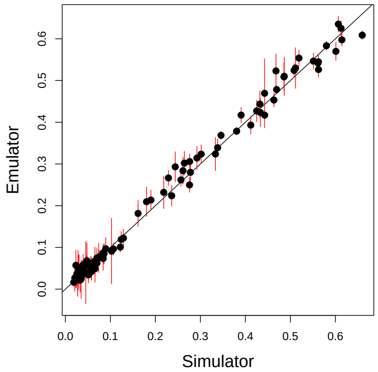
- Nilson-Kuusk model is a plant canopy reflectance model.
- For interpretation of remote sensoring data
- For determination of agronomical and phytometric parameters
    - The Nilson-Kuusk model is a single output model with 5 inputs
    - The training data contains 150 points
    - The validation data contains 100 points

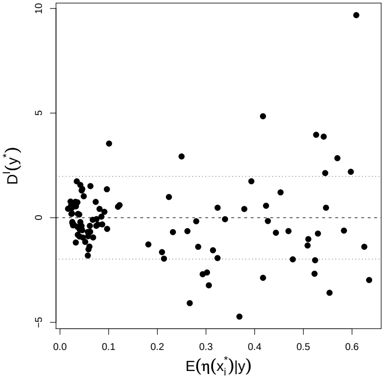
Example: Nilson-Kuusk model

- Nilson-Kuusk model is a plant canopy reflectance model.
- For interpretation of remote sensoring data
- For determination of agronomical and phytometric parameters
    - The Nilson-Kuusk model is a single output model with 5 inputs
    - The training data contains 150 points
    - The validation data contains 100 points

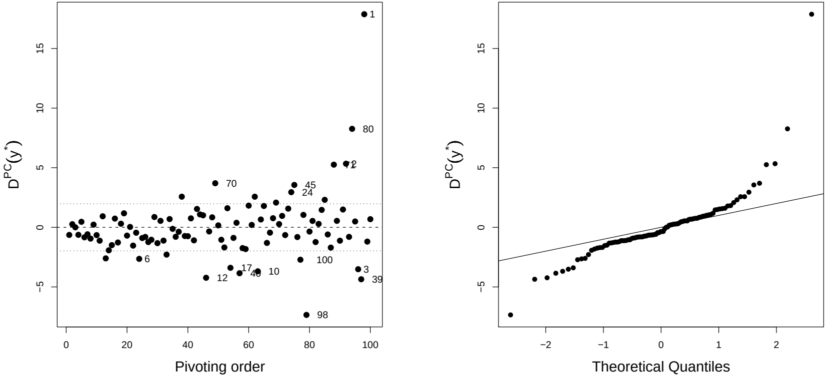
Example: Nilson-Kuusk model

- Nilson-Kuusk model is a plant canopy reflectance model.
- For interpretation of remote sensoring data
- For determination of agronomical and phytometric parameters
    - The Nilson-Kuusk model is a single output model with 5 inputs
    - The training data contains 150 points
    - The validation data contains 100 points


Emulator
Simulator
0.0
0.1
0.2
0.3
0.4
0.5
0.6


$D^I(y^*)$
$E(\eta(x_i^*)|y)$
-5
0
5
10
0.0
0.1
0.2
0.3
0.4
0.5
0.6

Graphical Diagnostics - Uncorrelated Errors

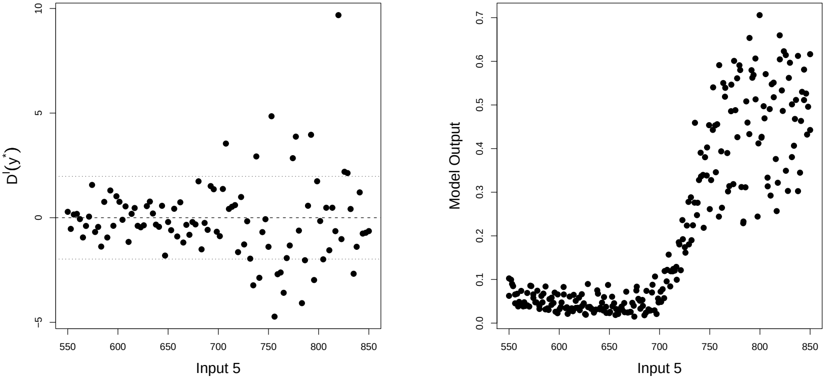


$D_{MD}(\mathbf{y}^*) = 750.237$ and the 95% CI is (69.0, 142.6)

Indicating a conflict between emulator and simulator.

The University Of Sheffield.

Graphical Diagnostics - Input 5



Actions for the Kuusk emulator

- The mean function $h(\cdot) = (1, \mathbf{x}, x_5^2, x_5^3, x_5^4)$
- Log transformation on outputs
- "new" dataset for validation

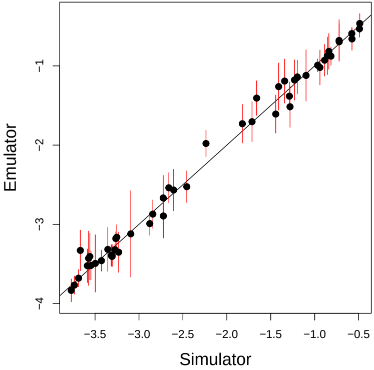
Actions for the Kuusk emulator

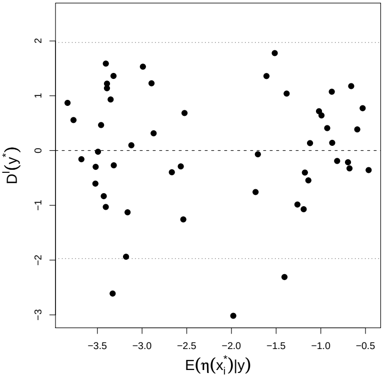
- The mean function $h(\cdot) = (1, \mathbf{x}, x_5^2, x_5^3, x_5^4)$
- Log transformation on outputs
- "new" dataset for validation

Actions for the Kuusk emulator

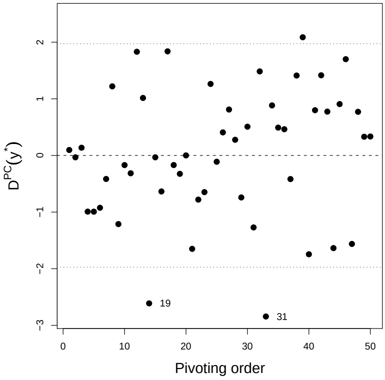
- The mean function $h(\cdot) = (1, \mathbf{x}, x_5^2, x_5^3, x_5^4)$
- Log transformation on outputs
- “new” dataset for validation

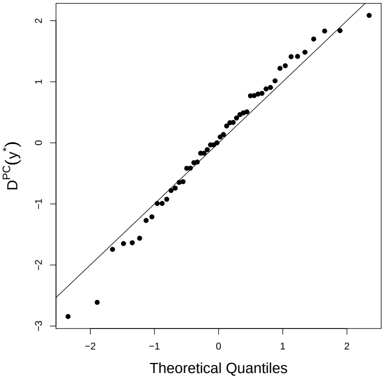
Individual errors





Uncorrelated Errors





$D_{MD}(\mathbf{y}^*) = 63.873$ and the 95% CI is (32.582, 79.508)

Conclusions

- Emulation is important when the computer model is expensive.
- Validating the emulator is necessary before using it for analyses using tyhe emulator as a surrogate of the computer model.
- Our diagnostics are useful tools inside the validation process.

Thank you!

Conclusions

- Emulation is important when the computer model is expensive.
- Validating the emulator is necessary before using it for analyses using tyhe emulator as a surrogate of the computer model.
- Our diagnostics are useful tools inside the validation process.

Thank you!

Conclusions

- Emulation is important when the computer model is expensive.
- Validating the emulator is necessary before using it for analyses using tyhe emulator as a surrogate of the computer model.
- Our diagnostics are useful tools inside the validation process.

Thank you!

The University Of Sheffield.

Conclusions

- Emulation is important when the computer model is expensive.
- Validating the emulator is necessary before using it for analyses using tyhe emulator as a surrogate of the computer model.
- Our diagnostics are useful tools inside the validation process.

Thank you!