NPL
National Physical Laboratory

Uncertainties in modelling

Louise Wright
25 January 2018

Data Science
National Physical Laboratory, UK

Acknowledgments

NPL National Physical Laboratory

- Much of the work was funded through the European Metrology Research Programme (EMRP) Project NEW04 Uncertainty. The EMRP is jointly funded by the EMRP participating countries within EURAMET and the European Union.

- Largely based on a freely available guide developed as part of the project credited previously.
- Includes longer explanation of the material you're about to see and real world examples.
- http://www.mathmet.org/publications/guides/index.php#expensive

EURAMET
European Association of National Metrology Institutes

Models in this talk

- Focus is on:
  - Physics-driven models
  - Computationally expensive models (e.g. finite element, finite volume, etc.)
  - Black box software
  - Propagation of uncertainties through a model
- Not considering model validation though there's an overlap
- Not considering approximation errors: assumed minimal compared to other uncertainty sources.

Sources of uncertainty in models

NPL National Physical Laboratory

- Need to either minimise or characterise all sources
- Physics
  - Multiphysics (e.g. losses) & nature of coupling
  - Steady state vs frequency domain vs transient
- Boundary conditions
  - Location of “infinity”
  - Point source approximations
- Geometry
  - Measurement uncertainty & spec vs true shape
  - Quality of contact between regions

Sources of uncertainty in models

- Material properties
  - Measurement uncertainty
    - Use of literature values
  - Temperature/field strength dependence
  - Batch to batch variability
  - Microstructure vs effective properties
  - Spatial uniformity, stability over time

Assumptions

- You already have a tested & validated model.
  - Uncertainties associated with model & software assumed to be minimised.
- You already have a joint distribution associated with your input quantities.
  - See GUM for guidance if not.
- You know which output quantities require evaluation of uncertainties.
- You have an upper limit on the number of model evaluations that can be made.

What's the problem?

- Model inputs have associated uncertainties, so model results need associated uncertainties.
- Many models (e.g. FE, CFD, etc.) take a long time to run.
- Software often "black box" so can't use analytical propagation of uncertainties.
- Computationally expensive so can't carry out enough model evaluations for Monte Carlo sampling to be viable.
  - A possible definition of "computationally expensive": MC would take too long.

Notation

- Model problem: random quantities, upper case.

$$\mathbf{Y} = \mathbf{F}(\mathbf{X}) \qquad \mathbf{Y} = \{Y_1, Y_2, \dots, Y_M\}^{\mathrm{T}}$$

$$\mathbf{X} = \{X_1, X_2, \dots, X_N\}^{\mathrm{T}}$$

- Model evaluations: values of random quantities, lower case.

$$\mathbf{y} = \mathbf{F}(\mathbf{x}) \qquad \mathbf{y} = \{y_1, y_2, \dots, y_M\}^{\mathrm{T}}$$

$$\mathbf{x} = \{x_1, x_2, \dots, x_N\}^{\mathrm{T}}$$

- Samples: superscript to indicate sample number.

$$\mathbf{x}^{(k)} = \left\{x_1^{(k)}, x_2^{(k)}, \dots, x_N^{(k)}\right\}^{\mathrm{T}} \qquad \mathbf{y}^{(k)} = \left\{y_1^{(k)}, y_2^{(k)}, \dots, y_M^{(k)}\right\}^{\mathrm{T}} \qquad k = 1, 2, \dots, K$$

- *N* is always number of input quantities, *K* is always number of model evaluations.

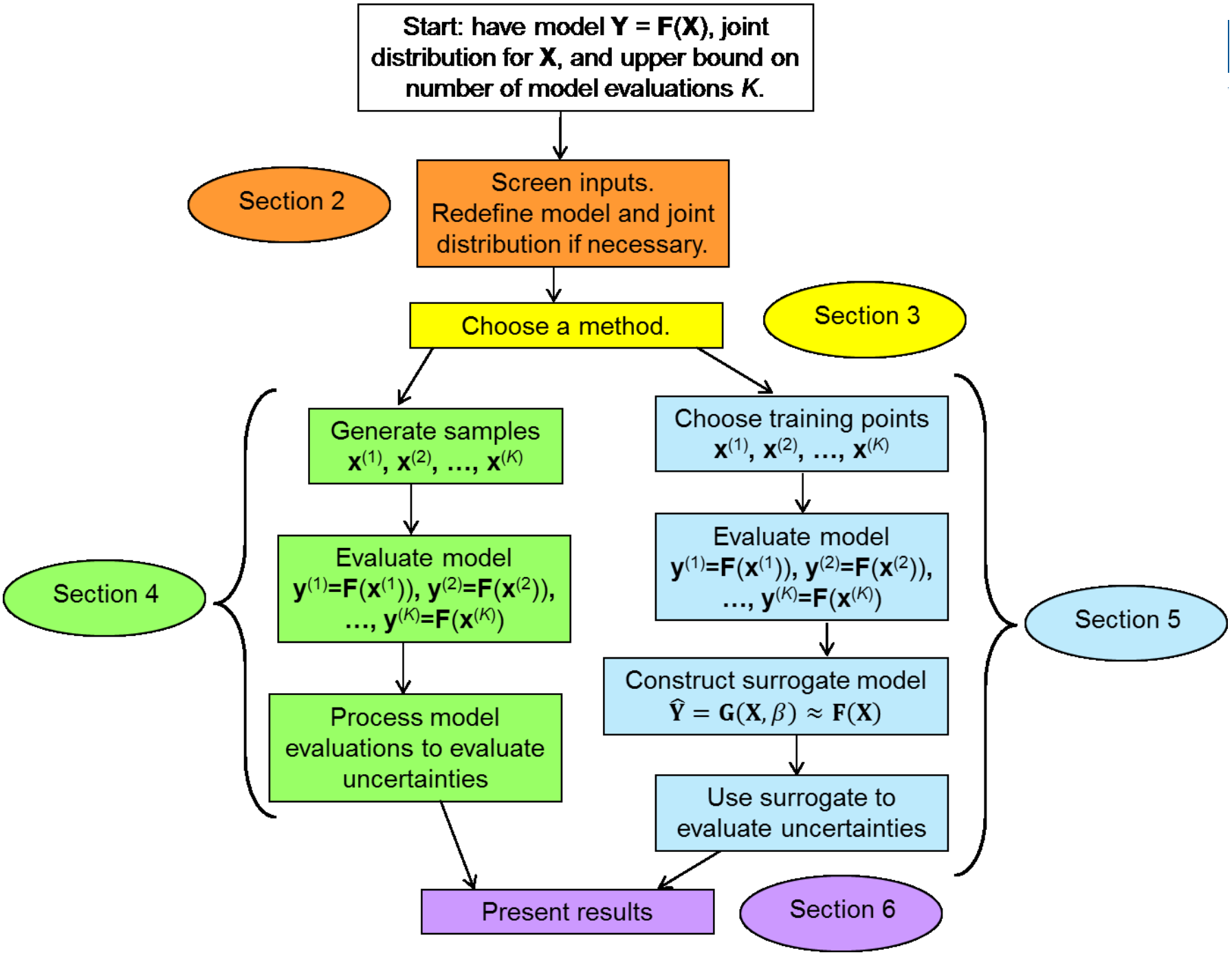
Vocabulary

- A model uses values of input quantities $\boldsymbol{x}$ to generate values of output quantities $\boldsymbol{y}$.
- A sampling method generates sample points $\boldsymbol{x}^{(k)}$.
- A surrogate model uses training points $\boldsymbol{x}^{(k)}$ and the associated function values $\boldsymbol{y}^{(k)}$, and possibly parameter values, to generate a new model that approximates the true function $\boldsymbol{F}$.

Toy problem

- Three input quantities
- X_1 uniformly distributed on [0, 1]
- X_2 triangular on [0,1] with a mode of 0.25
- X_3 Gaussian, mean 0.5, standard deviation 0.01.

- Single output quantity.
- $Y = X_1X_2 + X_1X_3 + X_2X_3 + \sin(2\pi X_1)$ but treat as a black box.
- Mean of Y is 0.67, standard deviation 0.57.
- Can afford 20 evaluations of the model.


Start: have model Y = F(X), joint distribution for X, and upper bound on number of model evaluations K.
Section 2
Screen inputs. Redefine model and joint distribution if necessary.
Choose a method.
Section 3
Section 4
Generate samples x(1), x(2), …, x(K)
Evaluate model y(1)=F(x(1)), y(2)=F(x(2)), …, y(K)=F(x(K))
Process model evaluations to evaluate uncertainties
Choose training points x(1), x(2), …, x(K)
Evaluate model y(1)=F(x(1)), y(2)=F(x(2)), …, y(K)=F(x(K))
Construct surrogate model Ŷ = G(X, β) ≈ F(X)
Use surrogate to evaluate uncertainties
Section 5
Present results
Section 6

Input screening & sensitivity

Screen inputs.
Redefine model and joint distribution if necessary.

What?

- Define a set of values of the input quantities.
- Evaluate the model using those input quantity values.
- Process the results to get information about the sensitivity of the output quantities to the input quantities.
  - Identify any insignficant input quantities.
- Redefine the model & distribution if necessary.

- Extra computational cost.
  - Potential benefits in the long run.

Why?

- Can identify input quantities that are not important.
  - Some uncertainty evaluation methods work better for fewer input quantities.
  - Reduced model may run more quickly.
  - Sampling over a reduced input space may be more likely to be space filling.
- Understanding sensitivity can provide insight into underpinning physics.

How?

- Design of experiments (DoE)
  - Full factorial design
  - Fractional factorial design
- “One at a time” designs
  - Morris
- Sobol’ indices

- Focus on DoE here: information on other methods in the BPG.

Design of Experiments

- Choose input quantity values to get as much information as possible about sensitivity.
- Full factorial designs select a fixed number of values of each input and evaluate every possible combination of those values.
  - Needs (number of values)^N model evaluations.
- Fractional factorial designs reduce the number of model evaluations
  - Lose the ability to distinguish between some interactions.

Full factorial design: two level design (effectively linear)

- Define a "high" and "low" value for each input quantity: x_i^+, x_i^-, $i = 1, 2, \ldots, N$.
- Construct every possible set of input quantities containing one of the two values for each input quantity.
- Evaluate the model 2^N times.
- To obtain the main effect for a given input x_i, evaluate

$$\frac{1}{2^{N-1}}\left(\sum_{k\in I^+}\mathbf{y}^{(k)} - \sum_{k\in I^-}\mathbf{y}^{(k)}\right) \qquad I^+ = \left\{k: x_i^+ \in \mathbf{x}^{(k)}\right\} \qquad I^- = \left\{k: x_i^- \in \mathbf{x}^{(k)}\right\}$$

- Similar expressions for interactions.

Significance

- Cannot decide whether to neglect an input quantity without a separate measure of importance.
  - Least signifcant input may still be signifcant.
- Use the standard error: $S/\sqrt{(2^N)}$, where S is the standard deviation of the 2^N output quantity values.
  - If an effect if less that the standard error, consider it to be insignificant
- NOTE: an input quantity with an insignificant main effect may be significant via interactions.
  - See BPG for an example.

Toy example: input screening

X_1	X_2	X_3	Y
0	0	0.48	0
0	0	0.52	0
0	1	0.48	0.48
0	1	0.52	0.52
1	0	0.48	0.48
1	0	0.52	0.52
1	1	0.48	1.96
1	1	0.52	2.04

$$\frac{S}{\sqrt{2^N}} \approx 0.28$$

X_3 effect = [(0 + 0.52 + 0.52 + 2.04) - (0 + 0.48 + 0.48 + 1.96)]/4

= 0.04

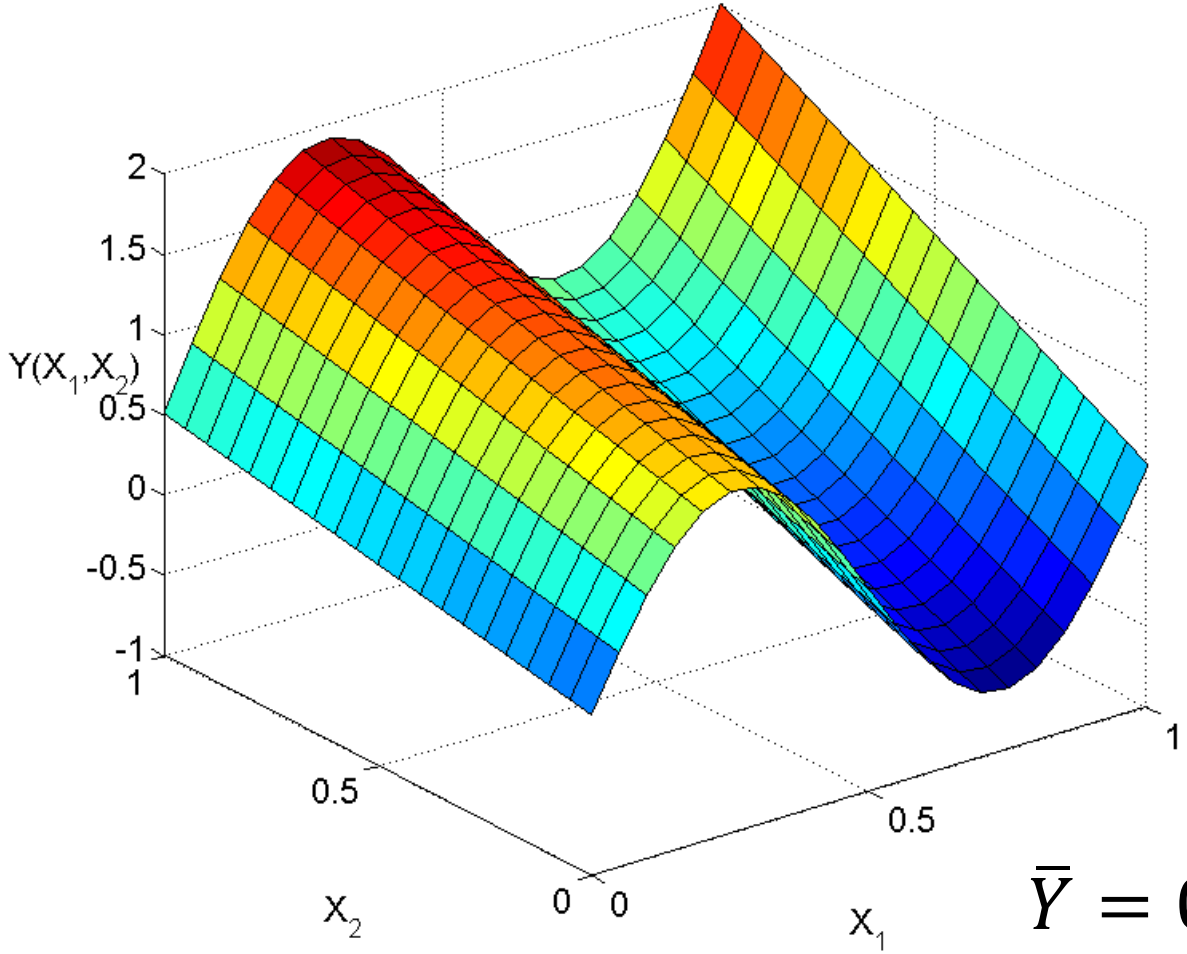
< standard error.

Toy problem: all effects

Variable or interaction	Size of effect $S/\sqrt{2^N} \approx 0.28$	Significant?
X_1	1	✅
X_2	1	✅
X_3	0.04	❌
$X_1 X_2$	0.5	✅
$X_1 X_3$	0.02	❌
$X_2 X_3$	0.02	❌
$X_1 X_2 X_3$	0	❌

X_3 is not significant as a main effect or an interaction, so we can neglect its uncertainty and fix it at its mean value.

New reduced model



$$Y(X_1,X_2) = X_1X_2 + 0.5\,X_1 + 0.5\,X_2 + \sin(2\,\pi\,X_1)$$

$$\bar{Y} = 0.66, \quad s = 0.57$$

Method choice

Choose a method.

Choosing a method

- Can’t produce definitive advice that applies to all problems.
- Set of points to consider: depends on what is most important for your application.
- Advice on comparing methods in the BPG.

- Methods are not necessarily mutually exclusive: may be able to use sampled points as training points for a surrogate model.

Points to consider - 1

- Number of input & output quantities.
  - More inputs → more parameters → more model evaluations, possible instability.
  - Higher dimensional space → more model evaluations if quadrature required.
- Method complexity and software availability.
  - Some methods require understanding of high level mathematics to implement.
  - Parameter estimation can require nonlinear optimisation.
  - May not matter if software is available.

Points to consider - 2

- Prior knowledge of model and input quantities.
  - Can improve efficiency if it is known which parts of the input space are important.
- Nature of joint distribution of inputs.
  - Some methods struggle with input quantity covariance.
  - Some methods will only work for some distributions.
- Historical model evaluations & need for sample size flexibility.
  - Stratified methods are not easy to adapt.

Method choice: toy problem

	RS	SS	LHS	PC	NN	QRS	GP
Small number of inputs	-	-	-	-	-	-	-
Complex methods are OK	-	-	-	-	-	-	-
No knowledge of function	-	✗	-	-	-	-	-
Independent inputs	-	-	-	-	-	-	-
Triangular distribution	-	-	-	✗	-	-	-
No historical evaluations	-	-	-	-	-	-	-
Final:	✓	✗	✓	✗	✓	✓	✓

Comparison of methods

- Extra computational cost: use a simpler model if possible.
- Compare accuracy
  - Requires reference results.
    - Large scale random sampling.
    - Literature values.
- Compare repeatability
  - Repeat runs for the same sample sizes.
  - Check sensitivity to training point & hyperparameter choices.
    - “Leave one out” method.

Smart sampling methods: what

Generate samples
$\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \ldots, \mathbf{x}^{(K)}$

↓

Evaluate model
$\mathbf{y}^{(1)}=\mathbf{F}(\mathbf{x}^{(1)}), \mathbf{y}^{(2)}=\mathbf{F}(\mathbf{x}^{(2)}), \ldots, \mathbf{y}^{(K)}=\mathbf{F}(\mathbf{x}^{(K)})$

↓

Process model evaluations to evaluate uncertainties

Why?

- Smart sampling methods can outperform MC sampling for small sample sizes.
  - Typically reduced sample to sample variability.
- Fairly simple to set up.
  - No parameter determination.
- Usually few restrictions on model & distributions.
- Can usually handle any number of input quantities.

How?

- Random/Monte Carlo sampling
  - Not recommended for small sample sizes.
- Importance sampling
  - Variance reduction method: not discussed here.
- Stratified sampling
- Latin hypercube sampling

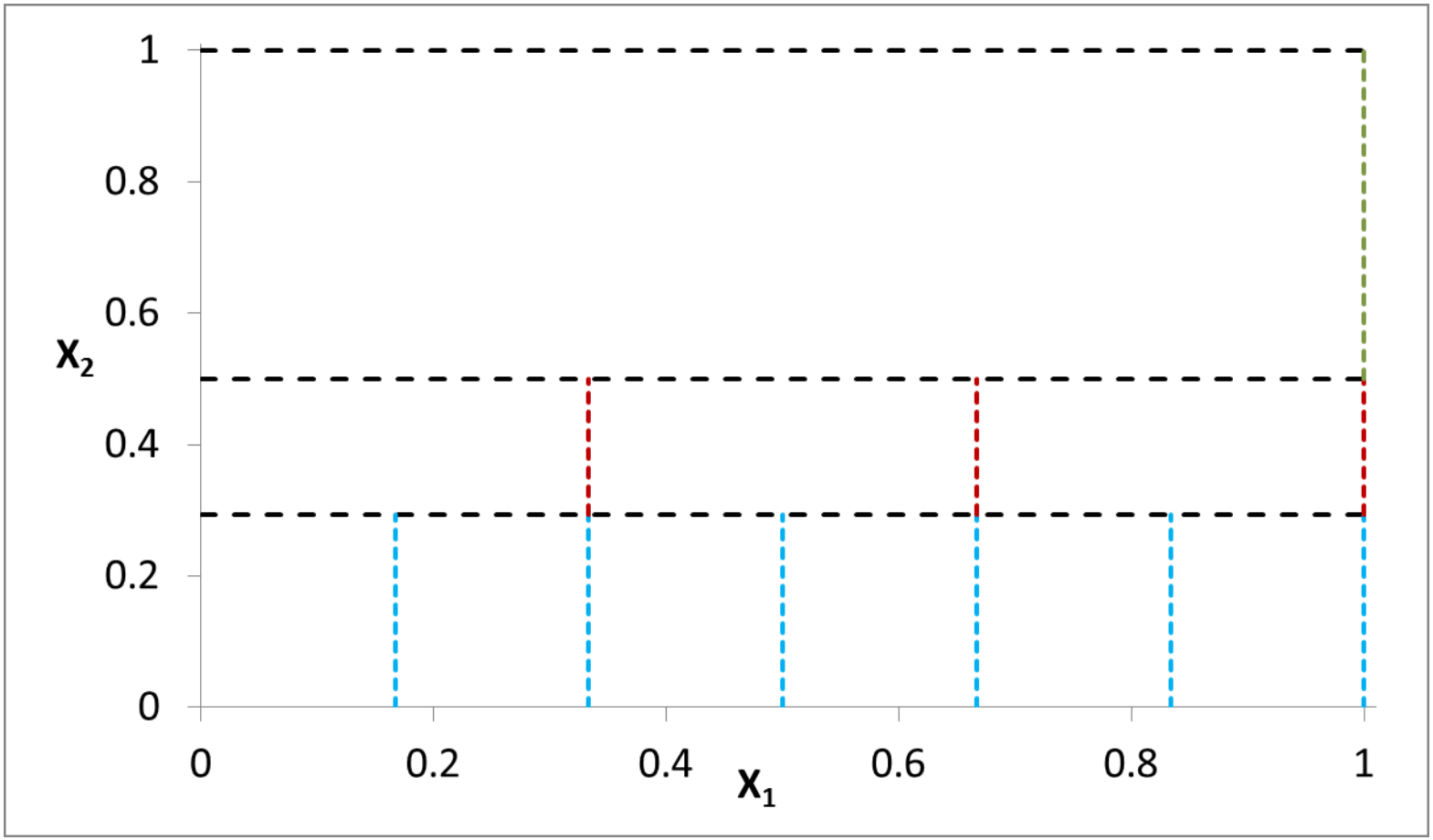
- Polynomial chaos

Stratified sampling

- Use any knowledge of model and input quantities to increase sample density in key regions.
- Subdivide the input space into non-overlapping regions of known probability p_j, $j = 1, 2, \ldots, J$.
- Sample K_j times within the jth region $\mathbf{x}(k,j)$, $j = 1, 2, \ldots, J$, $k = 1, 2, \ldots, K_j$.
- Process to get statistics.

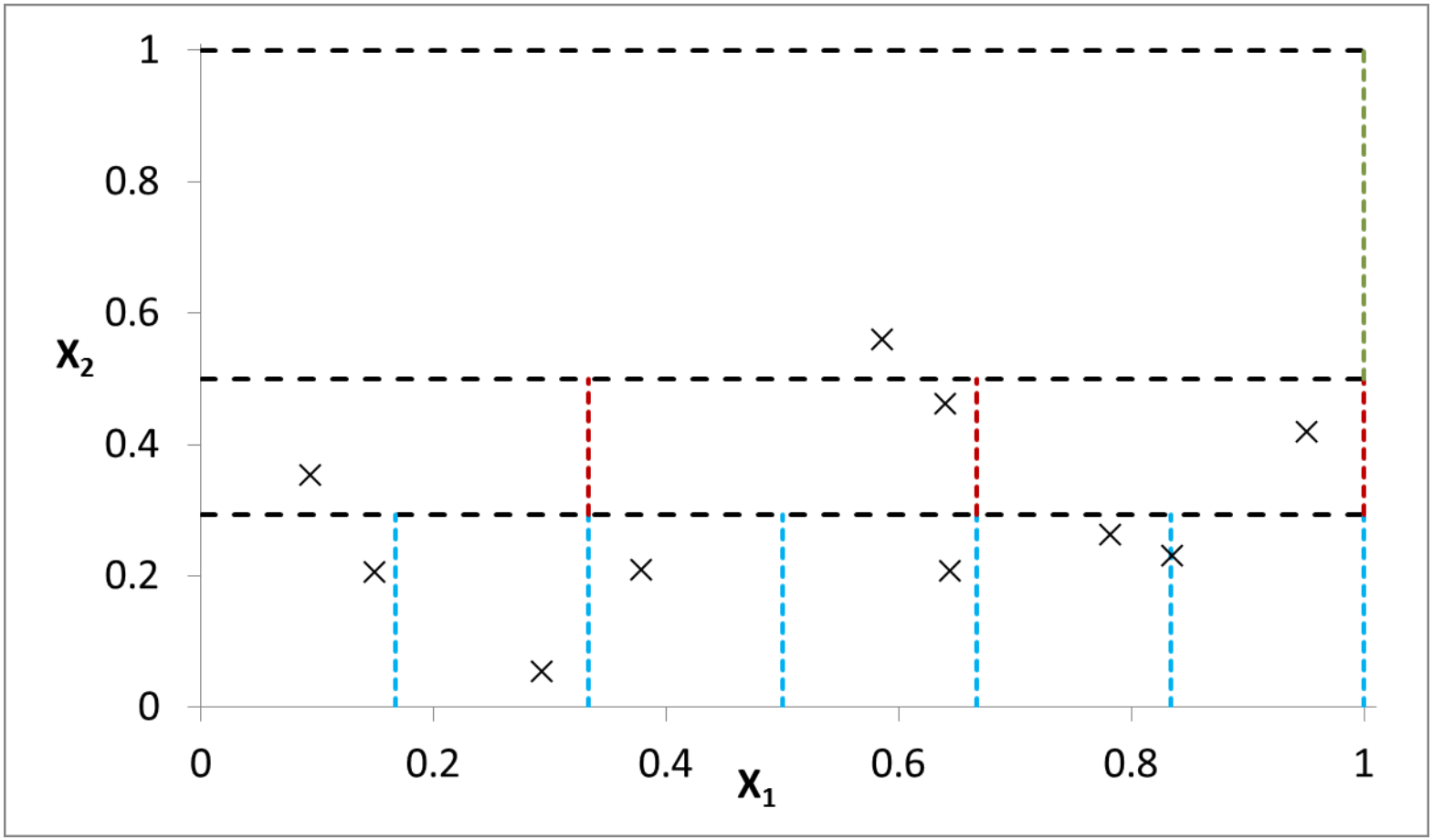
$$\overline{\mathbf{Y}} = \sum_{j=1}^{J} \frac{p_j}{K_j} \sum_{k=1}^{K_j} \mathbf{F}\left(\mathbf{x}^{(k,j)}\right) \qquad V = \sum_{j=1}^{J} \frac{p_j}{K_j} \sum_{k=1}^{K_j} \left(\mathbf{F}\left(\mathbf{x}^{(k,j)}\right) - \overline{\mathbf{Y}}\right)\left(\mathbf{F}\left(\mathbf{x}^{(k,j)}\right) - \overline{\mathbf{Y}}\right)^{\mathrm{T}}$$

Stratified sampling: how



X_1 uniformly distributed on [0, 1]
X_2 triangular on [0,1] with a mode of 0.25
Ten regions of equal probability defined.

Stratified sampling: how



$$\bar{Y}_{SS} = 0.47, \quad s_{SS} = 0.56$$

Stratified sampling: pros & cons

- Plus points:
  - uses available knowledge,
  - straightforward for independent inputs,
  - quite general.
- Minus points:
  - more difficult for correlated inputs,
  - best choice of subdivision not obvious in some cases.

Latin hypercube sampling

- Extension of stratified sampling
  - Assumes no knowledge of function.
- Ensure the full range of each input quantity is sampled.
  - Optimally space filling methods are available.

Latin hypercube sampling: how

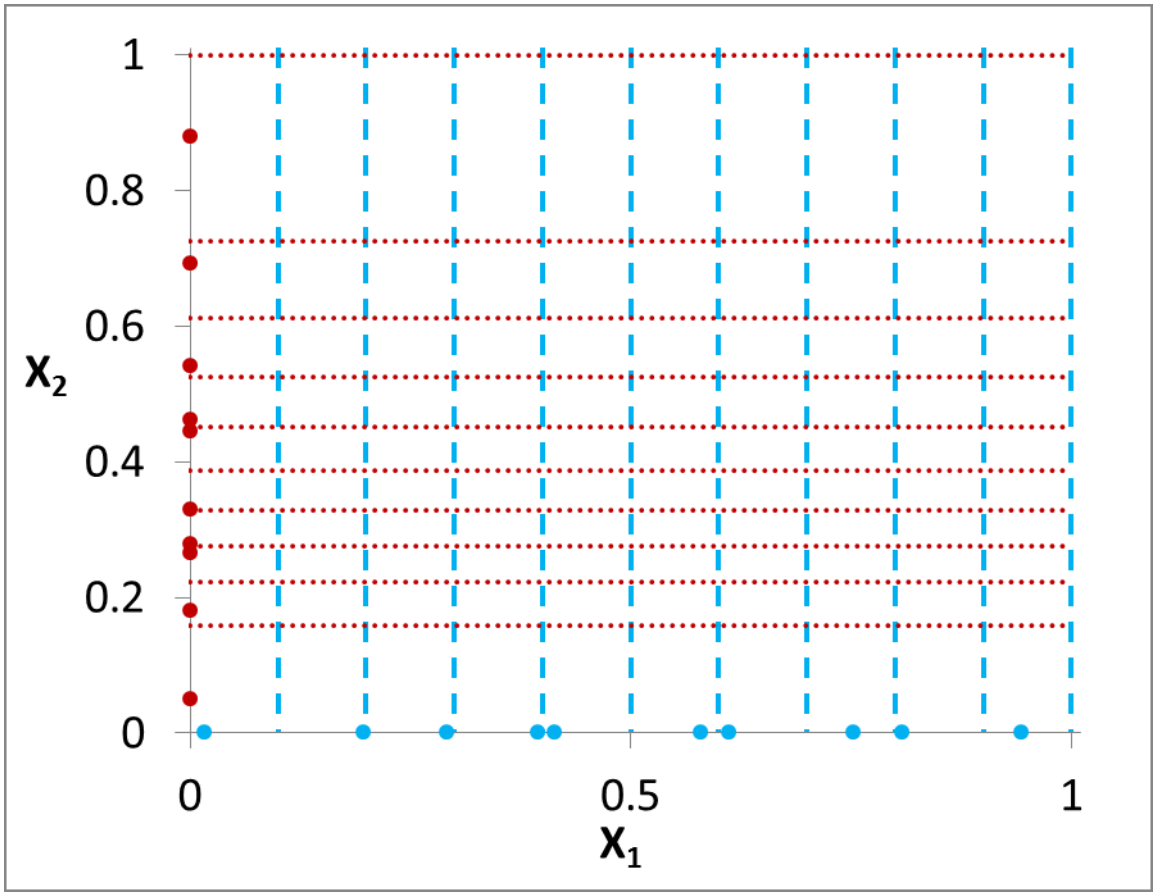
- Divide each input quantity into K regions of equal probability and sample once within each region:

  $x_i^{(k)}$, $i = 1, 2, \ldots, N$, $k = 1, 2, \ldots, K$ is in the kth region.
- Create N different permutations of the integers 1, 2, …, K. Let $n[i,j]$ be the jth number in the ith permutation.
- Reorder the ith variable so that the first sample value is $x_i^{(n[i,1])}$, the second is $x_i^{(n[i,2])}$, and so on.
- Construct the vectors of input quantities as

$$\mathbf{x}^{(j)} = \left\{ x_1^{(n[1,j])}, x_2^{(n[2,j])} \ldots, x_N^{(n[N,j])} \right\}^{\mathrm{T}}$$

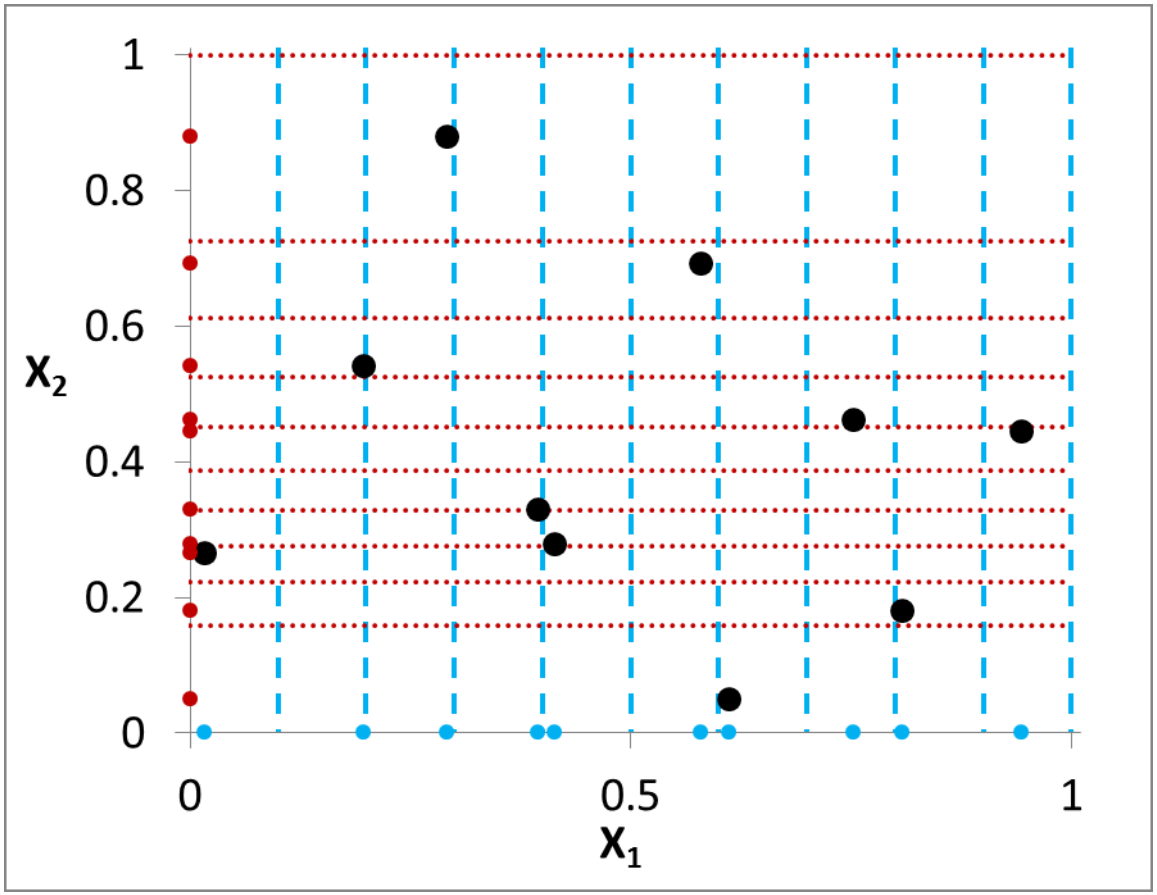
- Evaluate model and process as for random sampling.

Latin hypercube sampling: how



Divide each axis into ten regions of equal probability.
Sample once within each region.

Latin hypercube sampling: how



Pair the points up randomly.

$$\bar{Y}_{LHS} = 0.62, \quad s_{LHS} = 0.72$$

Latin hypercube sampling: pros & cons

- Plus points:
  - simple,
  - general,
  - can impose correlation structures on input samples.
- Minus points:
  - sample size not flexible.

Polynomial chaos

- Has elements of sampling and elements of surrogate modelling.
- Idea is to
  - treat the model output quantities as random quantities directly,
  - approximate model output as an expansion of polynomials Ψ of random variables **ξ**, $Y \approx \sum_{i=1}^{q} a_i \Psi_i(\boldsymbol{\xi})$
  - evaluate expansion coefficients a_i from model evaluations at well-chosen points,
  - can derive statistics directly from these coefficients.

Coefficients and evaluation

- Coefficients depend on integrals of the form

$$\int_{R^N} \Psi_i(s) w(s)\, F\big(\mathbf{x}(s)\big) ds$$

- where Ψ_k is the polynomial and *w* is a weighting function associated with the polynomial.
- Need numerical quadrature to evaluate.
  - Degree of polynomial & dimension of space affect number of evaluations required.
  - Can use Gaussian quadrature for small *N*.
  - Need to use sparse grid methods as *N* increases.

Polynomial chaos: pros & cons

- Plus points:
  - Excellent convergence: get very good approximation from low order polynomials.
  - Minimal processing.
  - Can use as a surrogate.
- Minus points:
  - Distribution restrictions: no correlation, non-standard distributions may suffer from poor convergence.
  - Can get expensive for more input quantities.
  - Quite mathematically involved, although software implementations are becoming more widespread.

Surrogate models: what

Choose training points $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \ldots, \mathbf{x}^{(K)}$

↓

Evaluate model $\mathbf{y}^{(1)}=\mathbf{F}(\mathbf{x}^{(1)}), \mathbf{y}^{(2)}=\mathbf{F}(\mathbf{x}^{(2)}), \ldots, \mathbf{y}^{(K)}=\mathbf{F}(\mathbf{x}^{(K)})$

↓

Construct surrogate model $\widehat{\mathbf{Y}} = \mathbf{G}(\mathbf{X}, \beta) \approx \mathbf{F}(\mathbf{X})$

↓

Use surrogate to evaluate uncertainties

Why?

- Replace the computationally expensive model with a model that is quicker to run and that gives the same results.
  - Can then use analytical methods or MC sampling for uncertainty evaluation.
- Replace the computationally expensive model with one that captures the random nature of the model solution directly.
  - Polynomial chaos, as discussed previously.

How?

- Nearest neighbour interpolation.
- Response surface methodology.
- Gaussian process emulators.

- Extra knowledge: could have a known expected model and add some form of error term.
- Could use local approximations similar to FE approach.

Training point selection

- Aim to fill input space and have more points where model output quantities are sensitive to input quantity values.
- Analytical methods.

  Regular grids, sparse grid methods, Hammersley sampling….
- Random methods.

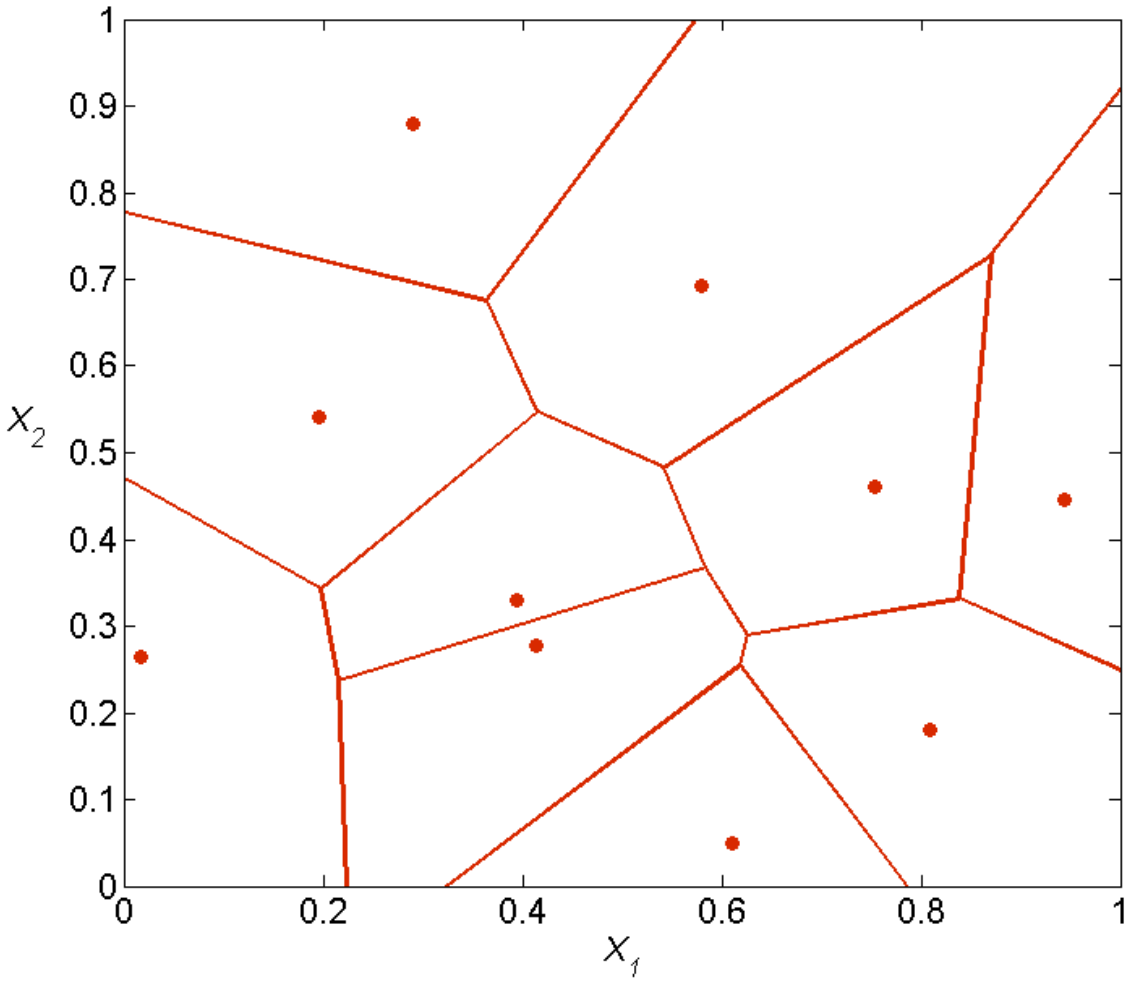
  Sampling methods described previously.

- Can reuse historical model evaluations.

Nearest neighbour: how

- Each training point has an associated neighbourhood.
- If defining neighbourhoods first, define a non-overlapping subdivision of the input space and choose one training point in each region.
- If choosing training points first, define neighbourhood of each point as the region that is closer to this training point than to any other training point (Voronoi polygons).
- To evaluate the surrogate at a new point, identify the neighbourhood in which the new point lies.
- New point has the function value of the training point associated with that volume.

Toy problem: nearest neighbour



$$\bar{Y}_{NN} = 0.62,$$

$$s_{NN} = 0.63$$

from 10^4 randomly chosen points. Mean is area-weighted average of training point values

Nearest neighbour: pros & cons

- Plus points
  - Simple to set up, particularly if neighbourhoods defined first.
  - Simple to understand.
- Minus points:
  - Less accurate, particularly for rapidly varying models.
  - Very sensitive to choice of training points.
  - Needs care with scaling to define “nearest”.

Response surface methodology

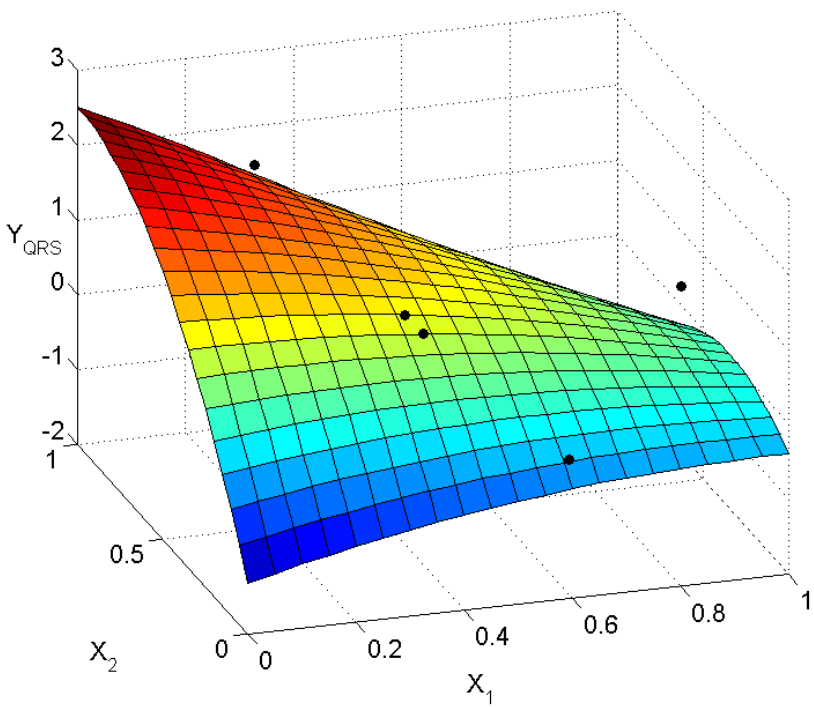
- Fit a simple polynomial surface $P(\mathbf{x})$ to the training points:
  - Minimise

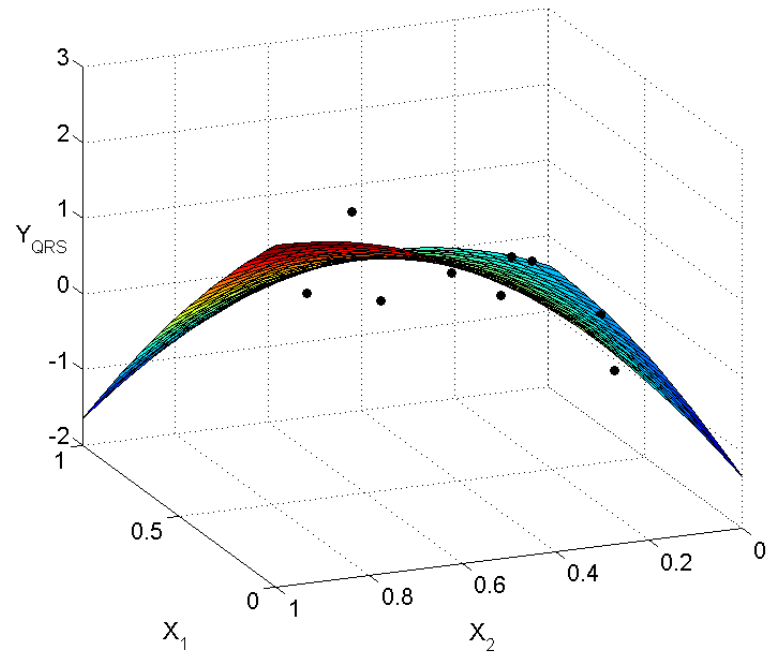
$$\sum_{k=1}^{K} \left(y_i^k - P\left(\mathbf{x}^{(k)}\right) \right)^2$$

- Evaluate the polynomial as a surrogate for the main model.

- Requires that K is large enough to uniquely define the coefficients of P.
- Best to normalise/rescale values before fitting.

Toy example: QRSM





$$\bar{Y}_{RSM} = 0.62, \quad s_{RSM} = 0.61 \qquad \Delta\bar{y} \approx 0.4$$

Reponse surface methodology

- Plus points
  - Fairly simple mathematically.
  - Works well for a lot of models.
  - No limits on distributions etc.
  - Can extend to other forms if likely behaviour is known.
- Minus points
  - Requires more point evaluations and can become less stable for higher order or large values of *N.*
  - QRS will not capture rapid changes in value & oscillatory behaviour.

Gaussian process emulation

- Approximate as a mean function β plus a zero-mean Gaussian process *Z* with covariance function C.
  - Gaussian process is a set of random variables with any subset having a Gaussian joint pdf

$$Y \approx \beta(\mathbf{X}) + Z(\mathbf{X})$$

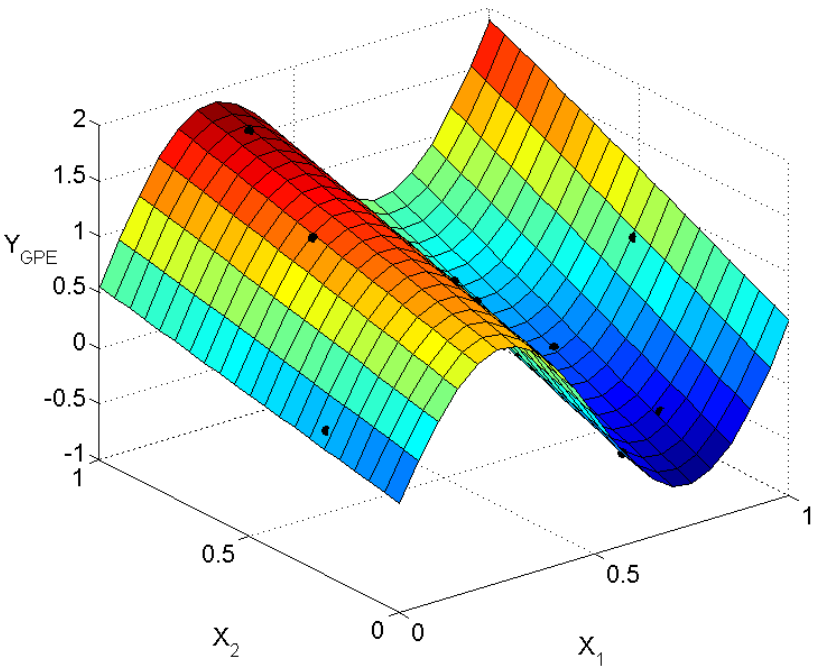
- Needs definition of a covariance function for the joint pdfs.
  - Common choice with useful smoothness properties:

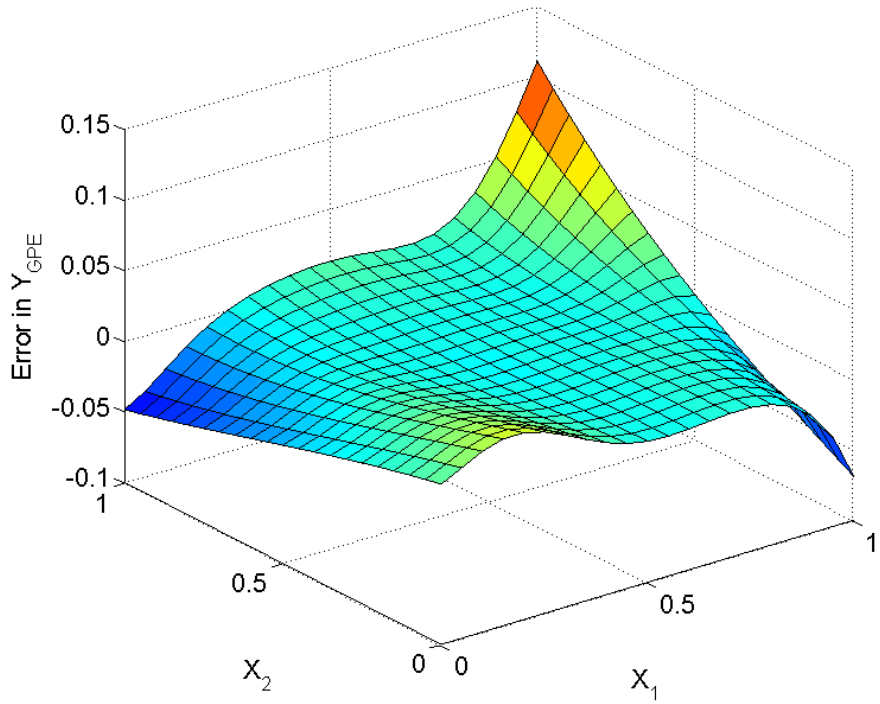
$$C(\mathbf{X}, \mathbf{X}') = \sigma^2 \prod_{i=1}^{N} \exp\left(-\left[\frac{X_i - X_i'}{\ell_i}\right]^2\right)$$

GPE: How to

- Select a correlation function.
- Identify suitable hyperparameter values.
  - Typically involves nonlinear optimisation.
- If the mean function is parameterised, determine mean function parameters.
- Evaluate the covariance matrix for every pair of training points: $C_{ij} = C(\mathbf{x}^{(i)}, \mathbf{x}^{(j)})$
- Given a new point, use the correlation matrix and best fit parameters to evaluate the surrogate (and an error estimate).

Toy example





$$\bar{Y}_{GPE} = 0.66, \quad s_{GPE} = 0.57$$

Gaussian process emulation: pros & cons

- Plus points
  - Can capture very general behaviour.
  - Provides error estimate at new points.
  - Not that sensitive to training point values.
- Minus points
  - Quite mathematically involved.
  - Hyperparameter determination not always easy.

Multi-run toy model comparison

Ten samples of size ten for all methods.
Reference values mean 0.66, std dev 0.57

Random	0.58	0.22	0.51	0.05
Stratified	0.56	0.14	0.54	0.05
Latin hyp	0.65	0.03	0.58	0.05
Nearest	0.64	0.04	0.57	0.05
QRSM	0.68	0.07	0.66	0.26
Gauss proc	0.66	0.006	0.57	0.005

Presenting results

Present results

Presenting results

- Summarising distributions (see GUM).
- Summarising sampling or surrogate models.
  - Give enough information to reproduce procedure.
- Visualisation methods
  - Ways to visualise repeatability.
  - Ways to compare distributions.

Thanks to

NPL National Physical Laboratory

- Johan Bunde Kondrup, Kurt Rasmussen (FORCE)
- Alexandre Allard, Severine Demeyer, Nicolas Fischer (LNE)
- Elena Barton, Clare Matthews, Dale Partridge (NPL)
- Markus Bär, Hermann Gross, Sebastian Heidenreich, Mark-A. Henn, Regine Model, Sonja Schmelter (PTB)
- Gertjan Kok, Nikola Pelevic (VSL)

NPL
National Physical Laboratory

Department for
Business, Energy
& Industrial Strategy

FUNDED BY BEIS

EURAMET
European Association of National Metrology Institutes

The National Physical Laboratory is operated by NPL Management Ltd, a wholly-owned company of the Department for Business, Energy and Industrial Strategy (BEIS).