Surrogate modelling meets machine learning

Other Conference Item

Author(s):
Sudret, Bruno; Lataniotis, Christos; Lüthen, Nora; Marelli, Stefano; Torre, Emiliano

Publication date:
2019-06-25

Permanent link:
https://doi.org/10.3929/ethz-b-000359596

Surrogate modelling meets machine learning

Bruno Sudret

Contributions from: C. Lataniotis, N. Lüthen, S. Marelli, E. Torre

Chair of Risk, Safety and Uncertainty Quantification
ETH Zurich

3rd International Conference on Uncertainty Quantification in Computational Sciences and Engineering
June 24-26, 2019

How to cite?

This presentation is the semi-plenary lecture given at the 3rd International Conference on Uncertainty Quantification in Computational Sciences and Engineering (UNCECOMP2019) in Hersonissos, Crete (Greece), June 24-26, 2019
(`https://2019.uncecomp.org/`)

How to cite

Sudret, B. *Surrogate modelling meets machine learning* (2019), Proc. 3rd Int. Conf. on Uncertainty Quantification in Computational Sciences and Engineering (UNCECOMP2019), Hersonissos, Crete (Greece), June 24-26.

Introduction: supervised learning

I. Goodfellow, Y. Bengio, A. Courville, *Deep learning*, MIT Press (2017)

- Machine learning aims at making predictions by building a model based on data
- Unsupervised learning aims at discovering a hidden structure within unlabelled data $\left\{\boldsymbol{x}^{(i)},\, i=1, \ldots, n\right\}$
- Supervised learning considers a training data set:

$$\mathcal{X}=\left\{(\boldsymbol{x}^{(i)}, y^{(i)}),\, i=1, \ldots, n\right\}$$

where:

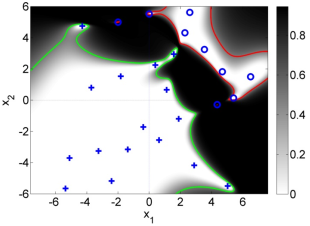
  - $\boldsymbol{x}^{(i)}$'s are the attributes / features (input space)
  - $y^{(i)}$'s are the labels (output space)

Classical problems and algorithms

Classification

- In classification problems, the labels are discrete, *e.g.* $y^{(i)} \in \{-1, 1\}$. The goal is to predict the class of a new point $\boldsymbol{x}$

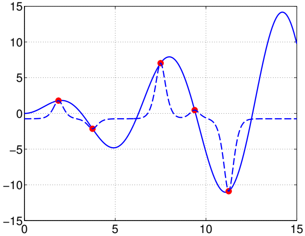
  Logistic regression - Support vector machines - (Deep) neural networks



Regression

- In regression problems, the labels are continuous, say $y^{(i)} \in \mathcal{D}_Y \subset \mathbb{R}$. The goal is to predict the value $\hat{y} = \tilde{\mathcal{M}}(\boldsymbol{x})$ for a new point $\boldsymbol{x}$

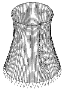
  Neural networks - Gaussian process models - Support vector regression



Uncertainty quantification

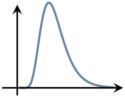
- A computational model is defined as a map:

$$x \in \mathcal{D}_{\boldsymbol{X}} \mapsto y = \mathcal{M}(\boldsymbol{x})$$

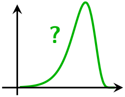


- Uncertainties in the input are represented by a probabilistic model:

$$\boldsymbol{X} \sim f_{\boldsymbol{X}} \qquad \text{(joint PDF)}$$



- Uncertainty propagation aims at estimating the statistics of $Y = \mathcal{M}(\boldsymbol{X})$



Surrogate models for uncertainty quantification

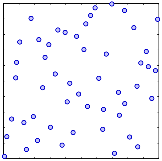
A surrogate model $\tilde{\mathcal{M}}$ is an approximation of the original computational model $\mathcal{M}$ with the following features:

- It is built from a limited set of runs of the original model $\mathcal{M}$ called the experimental design $\mathcal{X} = \left\{ \boldsymbol{x}^{(i)}, \, i = 1, \, \dots \, , n \right\}$
- It assumes some regularity of the model $\mathcal{M}$ and some general functional shape

Name	Shape	Parameters
Polynomial chaos expansions	$\tilde{\mathcal{M}}(\boldsymbol{x}) = \sum_{\boldsymbol{\alpha} \in \mathcal{A}} a_{\boldsymbol{\alpha}} \, \Psi_{\boldsymbol{\alpha}}(\boldsymbol{x})$	$\boldsymbol{a}_{\boldsymbol{\alpha}}$
Low-rank tensor approximations	$\tilde{\mathcal{M}}(\boldsymbol{x}) = \sum_{l=1}^{R} b_l \left(\prod_{i=1}^{M} v_l^{(i)}(x_i) \right)$	$b_l, \; z_{k,l}^{(i)}$
Kriging (a.k.a Gaussian processes)	$\tilde{\mathcal{M}}(\boldsymbol{x}) = \boldsymbol{\beta}^{\mathsf{T}} \cdot \boldsymbol{f}(\boldsymbol{x}) + Z(\boldsymbol{x}, \omega)$	$\boldsymbol{\beta} \, , \, \sigma_Z^2 \, , \, \boldsymbol{\theta}$
Support vector machines	$\tilde{\mathcal{M}}(\boldsymbol{x}) = \sum_{i=1}^{m} a_i \, K(\boldsymbol{x}_i, \boldsymbol{x}) + b$	$\boldsymbol{a} \, , \, b$
Neural networks	$\tilde{\mathcal{M}}(\boldsymbol{x}) = f_2 \left(b_2 + f_1 \left(b_1 + \boldsymbol{w}_1 \cdot \boldsymbol{x} \right) \cdot \boldsymbol{w}_2 \right)$	$\boldsymbol{w}, \boldsymbol{b}$

Ingredients for building a surrogate model

- Select an experimental design $\mathcal{X}$ that covers at best the domain of input parameters: Latin hypercube sampling (LHS), low-discrepancy sequences



- Run the computational model $\mathcal{M}$ onto $\mathcal{X}$ exactly as in Monte Carlo simulation

- Smartly post-process the data $\{\mathcal{X}\,,\,\mathcal{M}(\mathcal{X})\}$ through a learning algorithm

Name	**Learning method**
Polynomial chaos expansions	sparse grid integration, least-squares, compressive sensing
Low-rank tensor approximations	alternate least squares
Kriging	maximum likelihood, Bayesian inference
Support vector machines	quadratic programming

- Validate the surrogate model, *e.g.* estimate a global error $\varepsilon = \mathbb{E}\left[\left(\mathcal{M}(\boldsymbol{X}) - \tilde{\mathcal{M}}(\boldsymbol{X})\right)^2\right]$

Bridging supervised learning and PC expansions

Features	Supervised learning	Surrogate modelling
Computational model $\mathcal{M}$	✗	✔
Probabilistic model of the input $\boldsymbol{X} \sim f_{\boldsymbol{X}}$	✗	✔
Training data: $\mathcal{X} = \{(\boldsymbol{x}_i, y_i),\ i = 1, \ldots, n\}$	✔ Training data set (big data)	✔ Experimental design (small data)
Prediction goal: for a new $\boldsymbol{x} \notin \mathcal{X}$, $y(\boldsymbol{x})$?	$\sum_{i=1}^{m} y_i\, K(\boldsymbol{x}_i, \boldsymbol{x}) + b$	$\sum_{\boldsymbol{\alpha} \in \mathcal{A}} y_{\boldsymbol{\alpha}}\, \Psi_{\boldsymbol{\alpha}}(\boldsymbol{x})$
Validation (resp. cross-validation)	✔ Validation set	✔ Leave-one-out CV

Outline

Polynomial chaos expansions in a nutshell

Ghanem & Spanos (1991); S. & Der Kiureghian (2000); Xiu & Karniadakis (2002); Soize & Ghanem (2004)

- Consider the input random vector $\boldsymbol{X}$ ($\dim \boldsymbol{X} = M$) with given probability density function (PDF) $f_{\boldsymbol{X}}(\boldsymbol{x}) = \prod_{i=1}^{M} f_{X_i}(x_i)$

- Assuming that the random output $Y = \mathcal{M}(\boldsymbol{X})$ has finite variance, it can be cast as the following polynomial chaos expansion:

$$Y = \sum_{\boldsymbol{\alpha} \in \mathbb{N}^M} y_{\boldsymbol{\alpha}} \, \Psi_{\boldsymbol{\alpha}}(\boldsymbol{X})$$

  where :

  - $\Psi_{\boldsymbol{\alpha}}(\boldsymbol{X})$: basis functions
  - $y_{\boldsymbol{\alpha}}$: coefficients to be computed (coordinates)

- The PCE basis $\left\{ \Psi_{\boldsymbol{\alpha}}(\boldsymbol{X}), \, \boldsymbol{\alpha} \in \mathbb{N}^M \right\}$ is made of multivariate orthonormal polynomials

$$\Psi_{\boldsymbol{\alpha}}(\boldsymbol{x}) \stackrel{\text{def}}{=} \prod_{i=1}^{M} \Psi_{\alpha_i}^{(i)}(x_i) \qquad \mathbb{E}\left[\Psi_{\boldsymbol{\alpha}}(\boldsymbol{X}) \Psi_{\boldsymbol{\beta}}(\boldsymbol{X}) \right] = \delta_{\boldsymbol{\alpha}\boldsymbol{\beta}}$$

Isoprobabilistic transform

Premise

Classical orthogonal polynomials are defined for reduced variables *e.g.* standard normal variables $\mathcal{N}(0,1)$ (Hermite polynomials) or standard uniform variables $\mathcal{U}(-1,1)$ (Legendre polynomials)

How to handle arbitrary distributions?

Independent variables with given CDF F_{X_i}

- Use arbitrary PCE — Wan & Karniadakis (2006); Oladyshkin & Nowak (2012)

  Univariate polynomials $\left\{\Psi_k^{(i)}\right\}_{k\geq 0}$ are constructed numerically so as to be orthogonal w.r.t f_{X_i}

- Use a one-to-one mapping to reduced variables: — Berveiller *et al.* (2006)

$$X_i = F_{X_i}^{-1}\left(\frac{\xi_i+1}{2}\right) \qquad \text{if } \xi_i \sim \mathcal{U}(-1\,,\,1)$$

$$X_i = F_{X_i}^{-1}\left(\Phi(\xi_i)\right) \qquad \text{if } \xi_i \sim \mathcal{N}(0,1)$$

Isoprobabilistic transform

Dependence: copula representation

- Copula theory allows one to represent the joint CDF $F_{\boldsymbol{X}}$ by the set of marginal distributions $\{F_{X_1}, \ldots, F_{X_M}\}$ and a copula $\mathcal{C}$
- Sklar's theorem:

$$F_{\boldsymbol{X}}(\boldsymbol{x}) = \mathcal{C}\left(F_{X_1}(x_1), \ldots, F_{X_M}(x_M)\right)$$

Example: Gaussian copula

$$\mathcal{C}^{\mathcal{N}}(\boldsymbol{u}; \boldsymbol{\Theta}) = \Phi_M\left(\Phi^{-1}(u_1), \ldots, \Phi^{-1}(u_M); \boldsymbol{\Theta}\right)$$

where Φ_M is the multivariate Gaussian CDF

Inference

- In practice, marginals and copulas are inferred from data sequentially
- Rosenblatt or Nataf isoprobabilistic transforms can be used to map $\boldsymbol{X}$ to a vector $\boldsymbol{Z}$ with independent components

Outline

1. Introduction
2. Sparse polynomial chaos expansions
   - Spectral expansion
   - Computing the coefficients
   - Sparse PCE
3. Applications in machine learning
4. Surrogate models in high dimensions

Computing the coefficients by least-square minimization

Principle

Isukapalli (1999); Berveiller, S. & Lemaire (2006)

The exact (infinite) series expansion is considered as the sum of a truncated series and a residual:

$$Y = \mathcal{M}(\boldsymbol{X}) = \sum_{\boldsymbol{\alpha} \in \mathcal{A}} y_{\boldsymbol{\alpha}} \Psi_{\boldsymbol{\alpha}}(\boldsymbol{X}) + \varepsilon_P \equiv \mathbf{Y}^{\mathsf{T}} \boldsymbol{\Psi}(\boldsymbol{X}) + \varepsilon_P(\boldsymbol{X})$$

where : $\mathbf{Y} = \{y_{\boldsymbol{\alpha}},\, \boldsymbol{\alpha} \in \mathcal{A}\} \equiv \{y_0, \dots, y_{P-1}\}$ (P unknown coef.)

$$\boldsymbol{\Psi}(\boldsymbol{x}) = \{\Psi_0(\boldsymbol{x}), \dots, \Psi_{P-1}(\boldsymbol{x})\}$$

Least-square minimization

The unknown coefficients are estimated by minimizing the mean square residual error:

$$\hat{\mathbf{Y}} = \arg\min \mathbb{E}\left[\left(\mathbf{Y}^{\mathsf{T}} \boldsymbol{\Psi}(\boldsymbol{X}) - \mathcal{M}(\boldsymbol{X})\right)^2\right]$$

$$= \arg\min_{\mathbf{Y} \in \mathbb{R}^P} \frac{1}{n} \sum_{i=1}^{n} \left(\mathbf{Y}^{\mathsf{T}} \boldsymbol{\Psi}(\boldsymbol{x}^{(i)}) - \mathcal{M}(\boldsymbol{x}^{(i)})\right)^2$$

Validation: error estimators

- In least-squares analysis, the generalization error is defined as:

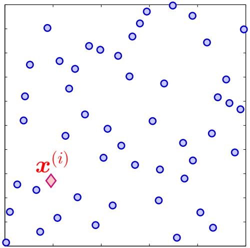
$$E_{gen} = \mathbb{E}\left[\left(\mathcal{M}(\boldsymbol{X}) - \mathcal{M}^{\mathsf{PC}}(\boldsymbol{X})\right)^2\right] \qquad \mathcal{M}^{\mathsf{PC}}(\boldsymbol{X}) = \sum_{\boldsymbol{\alpha}\in\mathcal{A}} y_{\boldsymbol{\alpha}}\,\Psi_{\boldsymbol{\alpha}}(\boldsymbol{X})$$

Leave-one-out cross validation

- From statistical learning theory, model validation shall be carried out using independent data
- LOO cross-validation for PCE emulates it using all data at once

$$E_{LOO} = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{\mathcal{M}(\boldsymbol{x}^{(i)}) - \mathcal{M}^{PC}(\boldsymbol{x}^{(i)})}{1-h_i}\right)^2$$

where h_i is the i-th diagonal term of matrix $\mathbf{A}(\mathbf{A}^{\mathsf{T}}\mathbf{A})^{-1}\mathbf{A}^{\mathsf{T}}, \quad \mathbf{A}_{ij} = \Psi_j(\boldsymbol{x}^{(i)})$

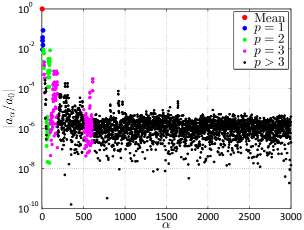


Outline

Curse of dimensionality and sparsity-inducing truncation

- The cardinality of the truncation scheme $\mathcal{A}^{M,p}$ is $P = \dfrac{(M+p)!}{M!\,p!}$
- Typical computational requirements: $n = OSR \cdot P$ where the oversampling rate is $OSR = 2 - 3$

However ... most coefficients are close to zero !



Sparsity-of-effects principle

Only low-order interactions between the input variables are relevant

Hyperbolic truncation

$$\mathcal{A}_q^{M,p} = \{\boldsymbol{\alpha} \in \mathbb{N}^M \ : \ ||\boldsymbol{\alpha}||_q \leq p\}$$

where $||\boldsymbol{\alpha}||_q \equiv \left(\sum_{i=1}^{M} \alpha_i^q\right)^{1/q} \quad , \quad 0 < q \leq 1$

Blatman & S., Prob. Eng. Mech (2010); J. Comp. Phys (2011)

Compressive sensing approaches

Blatman & S. (2011); Doostan & Owhadi (2011); Ian, Guo, Xiu (2012); Sargsyan *et al.* (2014); Jakeman *et al.* (2015)

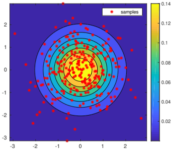
- Sparsity in the solution can be induced by ℓ_1-regularization:

$$\boldsymbol{y}_{\boldsymbol{\alpha}} = \arg\min \frac{1}{n} \sum_{i=1}^{n} \left(\mathsf{Y}^{\mathsf{T}} \boldsymbol{\Psi}(\boldsymbol{x}^{(i)}) - \mathcal{M}(\boldsymbol{x}^{(i)}) \right)^2 + \lambda \parallel \boldsymbol{y}_{\boldsymbol{\alpha}} \parallel_1$$

- Different solvers: LASSO, LAR, orthogonal matching pursuit, convex optimization (SPGL1), Bayesian compressive sensing

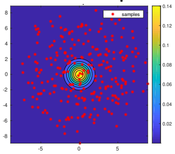
- Different sampling schemes

LHS



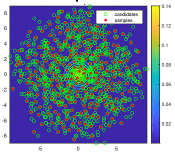
McKay, Beckman, Conover 1979

Coherence-optimal



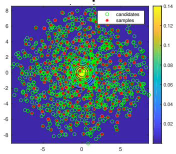
Hampton, Doostan 2015

D-optimal



Diaz, Doostan, Hampton 2018

Near-optimal



Alemazkoor, Meidani 2018

→ Talk by Nora Lüthen: "*Literature survey and benchmarking of sparse polynomial chaos expansions*" Tuesday 14:00 (MS5-II) in Room 2

Outline

Sparse PCE for supervised learning

Premise

Polynomial chaos expansions are built based on the PDF of the input parameters, assuming independence

Overview

- Build a probabilistic model of the input data, say $\hat{F}_{\boldsymbol{X}}$
- Transform into auxiliary independent variables $\boldsymbol{Z}$:

$$\mathcal{T} : \boldsymbol{X} \mapsto \boldsymbol{Z} \sim \prod_{i=1}^{M} f_{Z_i}(z_i)$$

- Map the data into the auxiliary space: $\boldsymbol{x}^{(i)} \longrightarrow \boldsymbol{z}^{(i)}$
- Use the new data set $\mathcal{Z} = \left\{ \left(\boldsymbol{z}^{(i)}, y^{(i)} \right), i = 1, \ldots, n \right\}$ for building a PCE

Probabilistic modelling of raw data

Copula representation

- Non-parametric estimation of the marginals

  For each univariate sample $\mathcal{X}_k \stackrel{\text{def}}{=} \left\{x_k^{(1)}, \ldots, x_k^{(n)}\right\}$ a kernel smoothing technique is used:

$$\hat{f}_{X_k}(x) = \frac{1}{n\,h_k} \sum_{i=1}^{n} K\left(\frac{x - x_k^{(i)}}{h}\right)$$

  - K: kernel function, *e.g.* the Gaussian kernel $\varphi(t) = e^{-t^2/2}/\sqrt{2\pi}$
  - h_k: bandwidth to be adapted to the data

- Estimation of the copula: requires flexibility in high-dimensions:

Vine copulas

Torre, Marelli, Embrechts, Sudret (2019), Prob. Eng. Mech. (2019)

Outline

Approach #1: PCE in the auxiliary space

Procedure

- Data: $\mathcal{X} = \left\{(\boldsymbol{x}^{(i)}, y^{(i)}),\ i = 1, \ldots, n\right\}$
- Use kernel smoothing for setting marginals
- Fit a copula *e.g.* Gaussian, vines
- Transform the data into the resulting auxiliary space, *e.g.* $[-1, 1]$, to use Legendre polynomials

Case of Gaussian copula (with correlation matrix $\hat{\boldsymbol{\Theta}}$)

Sudret *et al.* (2015), Int. Symp. on Big Data and Predictive Computational Modeling, Munich (Germany)

Gaussianize: $z_k^{(i)} = \Phi^{-1}(\hat{F}_{X_k}(x_k^{(i)}))$

Decorrelate $\boldsymbol{z}$'s: $\tilde{\boldsymbol{z}}^{(i)} = \mathsf{L}^{-1} \cdot \boldsymbol{z}^{(i)}$ where $\hat{\boldsymbol{\Theta}} = \mathsf{L} \cdot \mathsf{L}^{\mathsf{T}}$

General case: vine copulas

Torre, Marelli, Embrechts, Sudret (2019), J. Comput. Phys. (2019)

→ Talk by Emiliano Torre: "Representation of complex dependencies with copulas in UQLab" Tuesday 14:00 (MS10-I) in Room 7

Approach #2: aPCE on marginals

Premise

- Approach #1 captures (some of) the complex data dependence, yet requires a non linear isop. transform into the auxiliary space
- The PC expansion approximates the combination of the “true model” and the isop. transform

$$\mathcal{M}(\boldsymbol{x}) = \mathcal{M}(\mathcal{T}(\boldsymbol{z})) = \sum_{\boldsymbol{\alpha} \in \mathcal{A}} a_{\boldsymbol{\alpha}} \, \Psi_{\boldsymbol{\alpha}}(\boldsymbol{z})$$

Alternative

Torre, Marelli, Embrechts, Sudret (2019), J. Comput. Phys.

Disregard the dependencies and work in the original space using arbitrary PCE based on non-parametric distributions

- Use kernel smoothing for representing the marginals $\left\{ \hat{F}_{X_i}, \, i = 1, \, \dots \, , M \right\}$
- Compute polynomials that are orthonormal to the PDF in each dimension
- Use least-square analysis with sparse aPCE

Outline

Combined cycle power plant (CCPP)

Data set

UC Irvine Machine Learning Repository

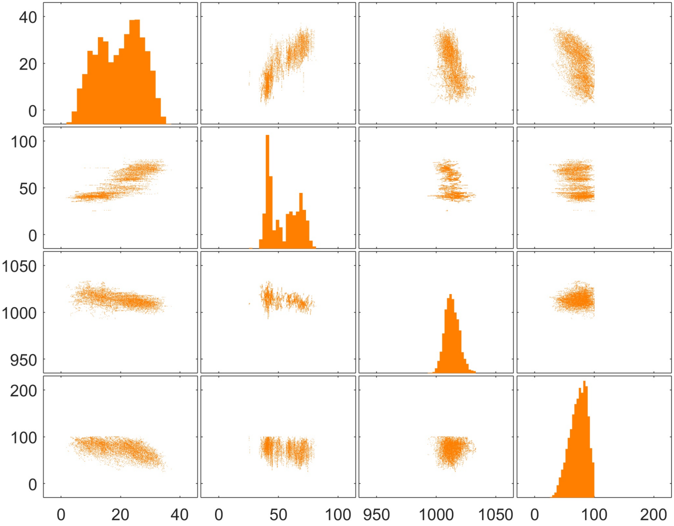
- 9,568 data points
- 4 features:
  - Temperature $T \in [1.81, 37.11]$ °C
  - Exhaust vacuum in the steam turbine $V \in [25.36, 81.56]$ cm Hg
  - Ambient pressure $P \in [992.89, 1033.30]$ mB
  - Relative humidity in the gas turbine $RH \in [25.56 - 100.16]$%
- Output: net hourly electrical energy output $EP \in [420.26, 495.76]$ MW

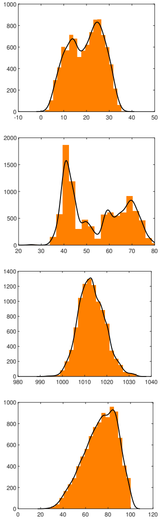
Reference approach

Tüfekci, P. (2014), *Int. J. Elec. Power & Energy Systems*

- 13 ML techniques including regression trees, ANN and SVR
- 10 pairs of training / validation sets of size 4,784
- Best approach: *bagging reduced error pruning (BREP) regression tree*

CCPP: Training data (X-space)



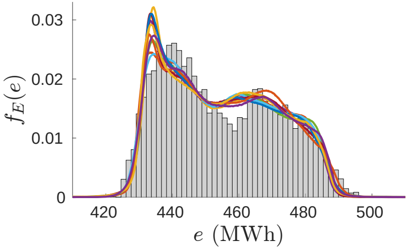


CCPP: Results

(Relative) mean absolute error

	MAE	min. MAE	mean-min	rMAE (%)
aPCEonX	3.11 ± 0.03	3.05	0.06	0.68 ± 0.007
BREP-NN†	3.22 ± n.a.	2.82	0.40	n.a.

† Tüfekci *et al.* (2014)



Estimated PDF of the energy produced by the CCPP:

- Histogram of raw data
- PDF obtained by PCE (10 diff. training sets) for input dependencies modelled by C-vines

Quality of *vinho verde* wines (Portugal)

Data set

http://www3.dsi.uminho.pt/pcortez/wine/

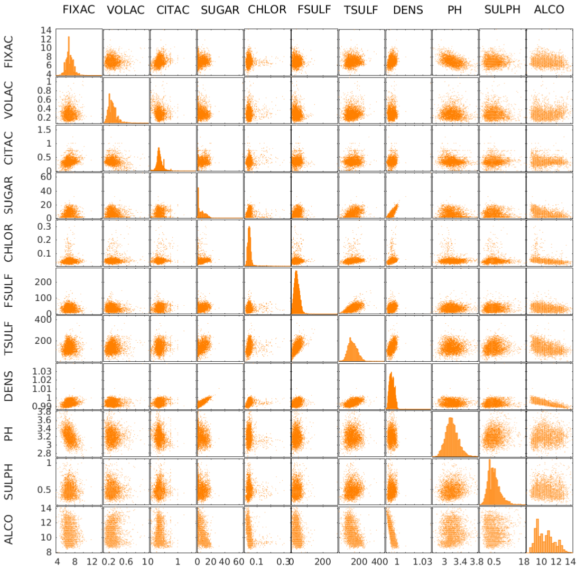
- 6,497 wine samples (1,599 red and 4,898 white) analyzed in laboratory for physico-chemical parameters, then graded by experts
- 11 features:
  - Fixed acidity
  - Volatile acidity
  - Citric acid
  - Residual sugar
  - Chlorides
  - Free sulfur dioxide
  - Total sulfur dioxide
  - Density
  - pH
  - Sulphates
  - Alcohol
- Output: Quality score Q, which is the median of 3 (integer) grades between 0 and 10 given by experts

Reference approach

Cortez *et al.* , Decision Support Systems (2009)

- Multilinear regression, single-layer NN, SVM
- 20 $\times$ 5-fold randomized cross validation

Quality of *vinho verde* wines: Training data (X-space)

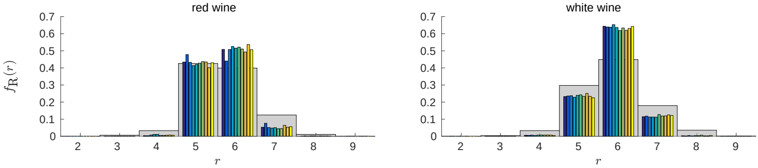


Quality of *vinho verde* wines: Results

(Relative) mean absolute error (MAE)

	Red wine		White wine	
	MAE	rMAE (%)	MAE	rMAE (%)
aPCEonX	0.44 ± 0.03	8.0 ± 0.6	0.50 ± 0.02	8.8 ± 0.3
SVM[†]	0.46 ± 0.00	n.a.	0.45 ± 0.00	n.a.
Best NN[†]	0.51 ± 0.00	n.a.	0.58 ± 0.00	n.a.

[†] Cortez *et al.* (2009)



- PCE predictions rounded to the closest integer

Airfoil

Data set

UC Irvine Machine Learning Repository

- 750 training points, 750 validation points
- 41 features:
  - Frequency, in Hertz
  - Angle of attack, in degrees
  - Chord length, in meters
  - Free-stream velocity, in meters per second.
  - Suction side displacement thickness, in meters
  - 36 noise variables (standard normal)
- Output: Scaled sound pressure level, in decibels

Reference approach

K. Kandasamy & Y. Yu, ICML16 Proc. of the 33rd Int. Conf. on Machine Learning (2016)

- Sparse LASSO regression (SALSA)
- Beats 13 other regression models, incl. neural networks

Airfoil: Results

(Relative) mean absolute error (MAE)

	MAE (dB)	rMAE (%)
aPCEonX	3.04 ± 0.07	2.4±0.06
SALSA†	3.81 ± 0.06	3.1±0.04

† Kandasamy & Yu (2016)

Outline

Challenges of modern engineering simulations

Medium-dimensional inputs

Typically $\mathcal{O}(10 - 100)$ (possibly dependent) input parameters

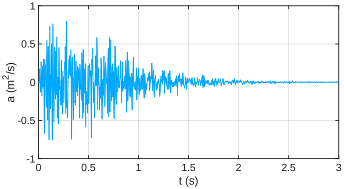
Sparse polynomial chaos expansions, low-rank tensor representations

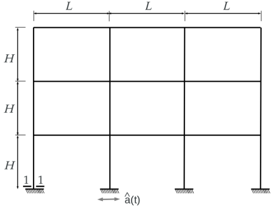
Functional inputs

- Time-series inputs, *e.g.* to represent climatic loads such as temperature history, wind velocity, etc.
- Maps of measured geometry/material properties: land elevation (river hydraulics), surface rugosity (contact problems in mechanical engineering), thermal conductivity, permeability, etc.

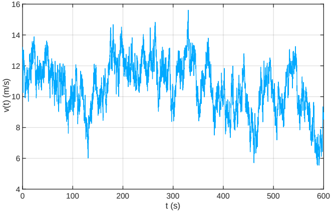
Challenges of modern engineering simulations: examples

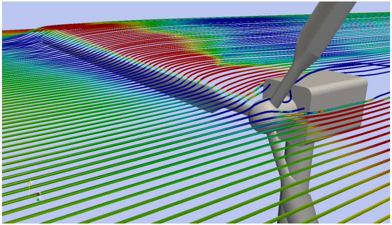
- Time-series accelerograms in structural dynamics (earthquake engineering)





- Wind velocity fields in the design of wind turbines





Common features

- High-dimensional inputs: 10^{3-5} time steps / pixels per input
- Underlying probabilistic model not necessarily available (data-driven UQ), *e.g.* when a catalog of recorded input signals is used (earthquake engineering)

> In order to use classical surrogate modelling techniques, dimensionality reduction must be used as a pre-processing

Dimensionality reduction

Dimensionality reduction

A mapping $g: \mathcal{X} \in \mathbb{R}^M \mapsto \mathcal{Z} \in \mathbb{R}^m$ $(m << M)$ of the form:

$$\mathbf{z} = g(\mathbf{x}; \mathbf{w})$$

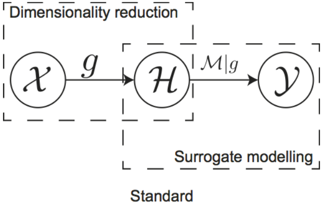
such that:

- It preserves some properties of $\mathcal{X}$ (*e.g.* information content)
- Its parameters $\mathbf{w}$ are inferred from the original data $\mathcal{X}$

Common dimensionality reduction methods

- Principal Component Analysis (PCA)
- Kernel PCA
- Autoencoders

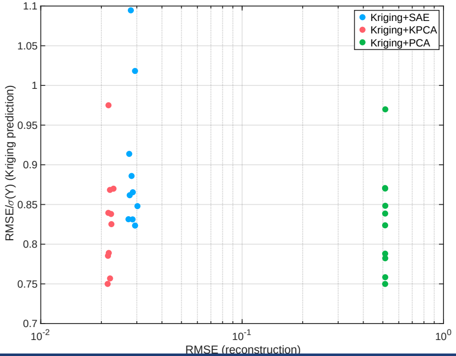
Sequential approach



- Perform dimensionality reduction of $\mathcal{X} \in \mathbb{R}^M$ to $\mathcal{Z} \in \mathbb{R}^m$
- Construct a surrogate model using the compressed data $\mathcal{Z}$, *i.e.* $y \approx \tilde{\mathcal{M}}(\mathbf{z})$

Findings

A good dimensionality reduction (w.r.t reconstruction error) does not mean that an accurate surrogate model can be built in the input space

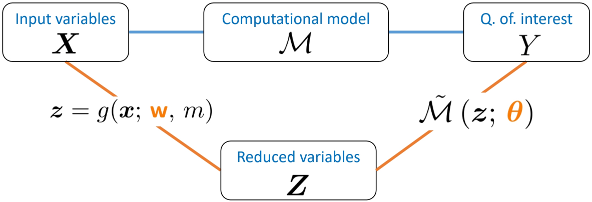


DRSM joint approach

Goal

Lataniotis, Marelli & Sudret (2018), arxiv 1812.06309

Optimize the parameters **w** of the DR step in such a way that the reduced variables $\boldsymbol{z} = g(\boldsymbol{x}; \mathbf{w})$ are suitable to achieve an overall accurate surrogate



Optimization procedure

- Loss function

$$\left\{\widehat{\mathbf{w}}, \widehat{\boldsymbol{\theta}}\right\} = \arg \min_{\mathbf{w} \in \mathcal{D}_{\mathbf{w}},\, \boldsymbol{\theta} \in \mathcal{D}_{\boldsymbol{\theta}}} \ell\left(\mathcal{M}(\cdot), \tilde{\mathcal{M}}\left(g(\cdot; \mathbf{w}), \boldsymbol{\theta}\right)\right)$$

- In practice, a RMS error on a validation set or a leave-one-out error

Block-coordinate descent optimization

Principle

Parameters $\mathbf{w}$ and $\boldsymbol{\theta}$ are updated in an alternating way

Outer loop

Optimize the compression parameters $\mathbf{w}$ so as to minimize the leave-one-out error of the surrogate:

$$\widehat{\mathbf{w}} = \arg\min_{\mathbf{w} \in \mathcal{D}_{\mathbf{w}}} \varepsilon_{LOO}(\mathbf{w}; \widehat{\boldsymbol{\theta}}(\mathbf{w}), \mathcal{X}, \mathcal{Y})$$

Inner loop

Given the current value of the compression parameters $\mathbf{w}$ and related $z(\mathbf{w})$'s, fit the surrogate:

$$\widehat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta} \in \mathcal{D}_{\boldsymbol{\theta}}} \varepsilon_{LOO}(\boldsymbol{\theta}; \mathbf{w}, \mathcal{X}, \mathcal{Y})$$

Computational efficiency

Low-cost intermediate surrogates

During the optimization, cheap-to-calibrate surrogates are used:

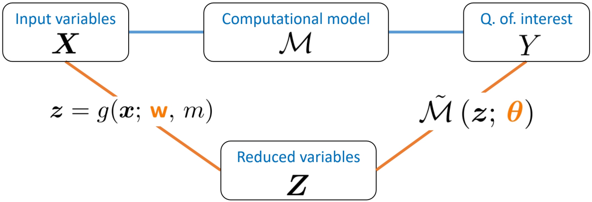
- Kriging: Isotropic kernel is used together with a limited computational budget
- PCE: Low-degree Legendre polynomials

Final surrogate

Once the optimal DR parameters $\mathbf{w}^*$ are obtained, a high-accuracy surrogate model is used:

- Kriging: Anisotropic kernels and global optimization + gradient-based refinement
- PCE: adaptive sparse arbitrary PCE based on the true distribution of the $\mathcal{Z}$'s (kernel density estimation)

Summary



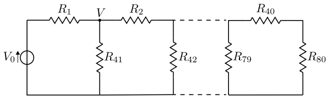
- DRSM is a generic algorithm that combines ML for compression and UQ for surrogating
- Given a data set, multiple combinations can be tested in parallel, *e.g.* {SAE, PCA, KPCAs} $\times$ {PCE, GP, etc.}
- It provides surrogates for models with high dimensional inputs (*e.g.* measured time series / fields)

$\rightarrow$ Talk by Stefano Marelli: "Combining machine learning and surrogate modelling for data-driven uncertainty propagation in high-dimension" Monday 11:30 (MS15-I) in Room 3

Outline

Unstructured data: resistor network

Dataset description

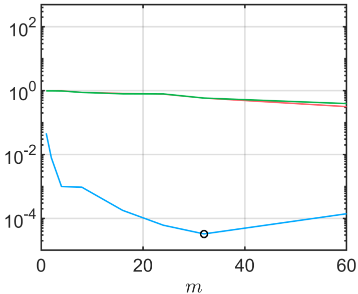


- Network comprised of 80 uncertain resistors
- Output of interest is voltage at V
- Effect of resistors onto V decays with distance

Data courtesy of J. Jakeman (SANDIA National Labs), Jakeman *et al.* , J. Comput. Phys (2015)

Optimal DRSM configuration

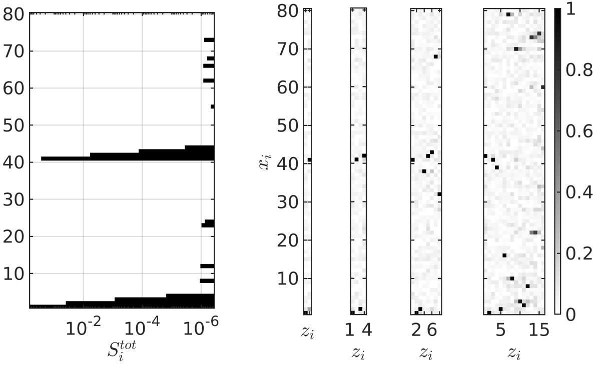
SM method	KPCA kernel	$\widehat{m}$	$\widehat{\varepsilon}_{gen}$
Kriging	Anis. Gaussian	24	$2.40 \cdot 10^{-4}$
PCE	Anis. Gaussian	32	$3.25 \cdot 10^{-5}$



$\widehat{\varepsilon}_{gen}$ **as a function of** m

Unstructured data: resistor network

- Kernel PCA with anisotropic Gaussian kernel combined with PCE yields the most accurate surrogate ($\widehat{\varepsilon}_{gen} = 3.25 \cdot 10^{-5}$)
- The optimal reduced dimension m is equal to 32



- The first auxiliary variables $\{z_1, \ldots, z_m\}$ correspond one-to-one to the important parameters of the problem (based on Sobol' indices)

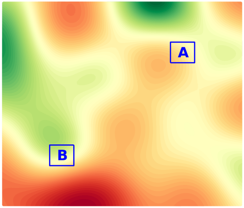
Data-driven heat diffusion problem

Problem statement

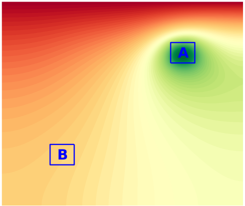
$$-\nabla\cdot(k(\mathbf{v})\nabla T(\mathbf{v})) = 500\, I_A(\mathbf{v}), \quad \mathbf{v} \in [-0.5, 0.5]^2$$

with boundary conditions:

- $T = 0$ on top boundary
- $\nabla T \cdot \boldsymbol{n} = 0$ on other boundaries



Input diffusion coefficient



Output temperature

Lognormal diffusion coefficient $k(\mathbf{v})$

$$k(\mathbf{v}) = \exp\left(a_d + b_d\, g(\mathbf{v})\right)$$

with mean value 1, std. deviation 0.3, square-exponential autocorrelation function:

$$R(\mathbf{v}, \mathbf{v}') = \exp\left(-\left\|\mathbf{v} - \mathbf{v}'\right\|^2 / \ell^2\right)$$

Konakli and Sudret, Prob. Eng. Mech. (2016)

Data-driven heat diffusion problem

Synthetic input maps

Li & Der Kiureghian (1993)

- The underlying Gaussian field is generated from an EOLE expansion

$$\widehat{g}(\mathbf{v}) = \sum_{i=1}^{p} \frac{\xi_i}{\sqrt{l^{(i)}}} \left(\phi^{(i)}\right)^{\top} \boldsymbol{C}_{\mathbf{v}\boldsymbol{v}}(\mathbf{v}),$$

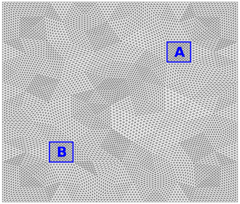
where:

$$C_{\mathbf{v}\boldsymbol{v}}^{(k)} = R(\mathbf{v}, \boldsymbol{v}_k), \qquad C_{\boldsymbol{v}\boldsymbol{v}}^{(i,j)} = R(\boldsymbol{v}_i, \boldsymbol{v}_j)$$

- 500 maps of diffusion coefficient are generated wrt the finite element mesh: $M = 16,000$-dimensional input

  (300 training and 200 validation points)

- Scalar output: average temperature in Domain B



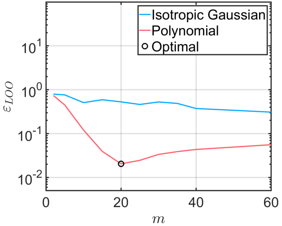
Results

DRSM output

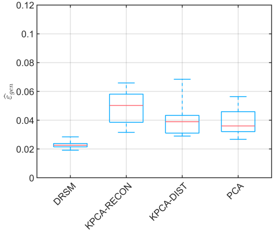
- Kernel PCA with polynomial kernel combined with Kriging yields the most accurate surrogate
- DRSM automatically finds PCA (polynomial degree 1) with $m = 20$ components as the best compression

Conclusion

- Fully data-driven surrogate of the map-to-temperature model
- Accurate estimation of the distribution of T through resampling of the auxiliary variables



LOO error as a function of m

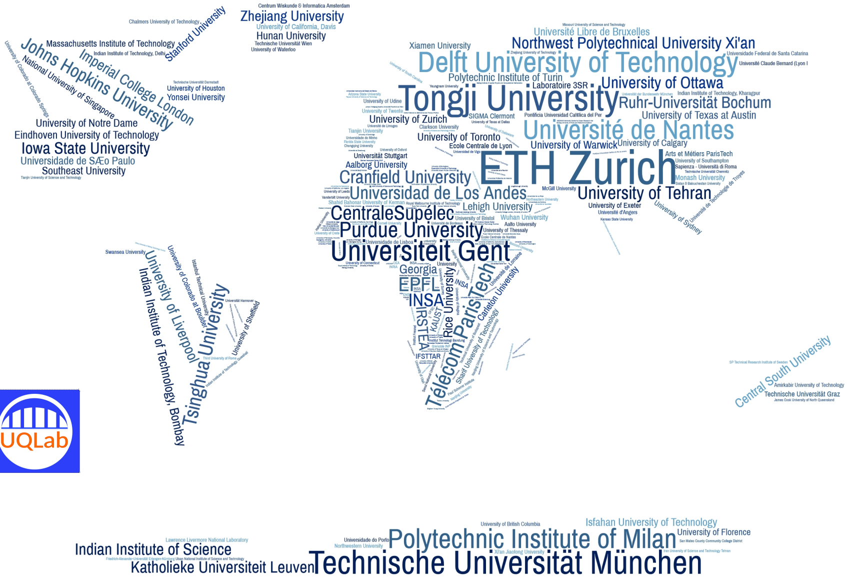


DRSM vs. sequential DR+SM

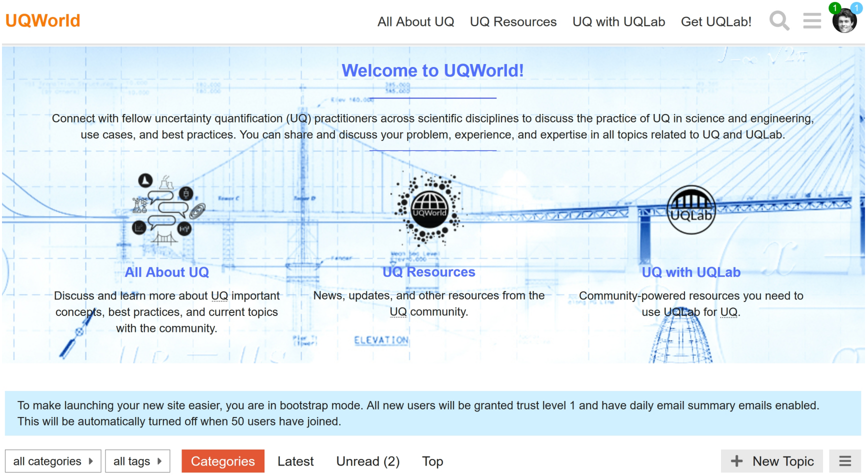
Conclusions

- Machine learning techniques and surrogate modelling for uncertainty quantification are closer than ever
- Sparse PCE can be used efficiently for supervised learning in low/medium dimension (better accuracy, no parameters to tweak)
- Compression techniques (PCA, Kernel PCA, stacked auto-encoders) can be used in combination with classical techniques (PCE, Kriging, etc.) to build data-driven surrogates in extreme input dimensions
- This opens the path to real-time simulation of models with continously measured input parameters

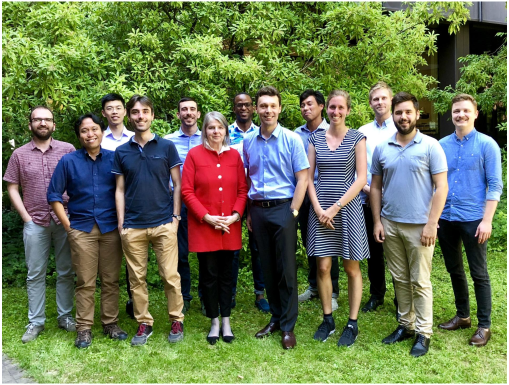
UQLab


Centrum Wiskunde & Informatica Amsterdam
Zhejiang University
University of California, Davis
Hunan University
Technische Universität Wien
University of Waterloo
Chalmers University of Technology
Stanford University
Massachusetts Institute of Technology
Indian Institute of Technology, Delhi
Johns Hopkins University
Imperial College London
National University of Singapore
University of Colorado at Colorado Springs
Technische Universität Darmstadt
University of Houston
Yonsei University
University of Notre Dame
Eindhoven University of Technology
Iowa State University
Universidade de SÆo Paulo
Southeast University
Tianjin University of Science and Technology
Missouri University of Science and Technology
Université Libre de Bruxelles
Xiamen University
Northwest Polytechnical University Xi'an
Zhejiang University of Technology
Universidade Federal de Santa Catarina
Delft University of Technology
Université Claude Bernard (Lyon I
Polytechnic Institute of Turin
Laboratoire 3SR
University of Ottawa
Tongji University
Ruhr-Universität Bochum
Indian Institute of Technology, Kharagpur
University of Udine
University of Twente
University of Zurich
SIGMA Clermont
Pontificia Universidad Católica del Per
University of Texas at Austin
Clarkson University
Université de Nantes
Tianjin University
University of Toronto
Ecole Centrale de Lyon
University of Warwick
University of Calgary
Universität Stuttgart
Aalborg University
ETH Zurich
Arts et Métiers ParisTech
University of Southampton
Sapienza - Università di Roma
Cranfield University
Monash University
Universidad de Los Andes
McGill University
University of Tehran
Shahid Bahonar University of Kerman
Lehigh University
CentraleSupélec
University of Bristol
Wuhan University
University of Exeter
Université d'Angers
Aalto University
Purdue University
University of Thessaly
Université de Technologie de Troyes
University of Sydney
Universidade de Lisboa
Universiteit Gent
Swansea University
University of Liverpool
Indian Institute of Technology, Bombay
University of Colorado at Boulder
Istanbul Technical University
Tsinghua University
Universität Hannover
University of Sheffield
Georgia
EPFL
INSA
IRSTEA
KAUST
Rice University
ParisTech
Télécom
Carleton University
Université de Lorraine
Sharif University of Technology
IFSTTAR
Institut Teknologi Bandung
Central South University
Amirkabir University of Technology
Technische Universität Graz
James Cook University of North Queensland
University of British Columbia
Isfahan University of Technology
Lawrence Livermore National Laboratory
Indian Institute of Science
Universidade do Porto
Northwestern University
Polytechnic Institute of Milan
University of Florence
Xi'an Jiaotong University
Katholieke Universiteit Leuven
Technische Universität München

[NEW] The applied UQ community: uqworld.org



Questions ?



Chair of Risk, Safety & Uncertainty Quantification

www.rsuq.ethz.ch

The Uncertainty Quantification Software

www.uqlab.com

Thank you very much for your attention !