

Bayesian compressive sensing and dimensionality reduction for high-dimensional models

Khachik Sargsyan



Sandia National Laboratories
Livermore, CA



Main Collaborators:

*Habib Najm, Bert Debusschere, Cosmin Safta (SNL)
Daniel Ricciuto, Peter Thornton (ORNL)*

*Thanks to DOE ASCR, DOE BER,
under Climate Science for Sustainable Energy Future (CSSEF)*

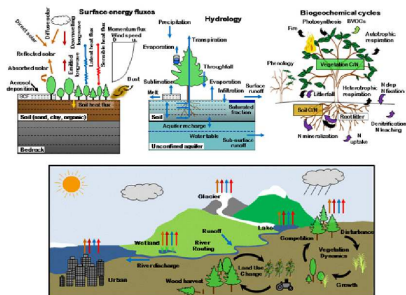
Sandia National Laboratories is a multi-program laboratory operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract

DE-AC04-94AL85000

OUTLINE

- **Surrogates** needed for complex models
- **Polynomial Chaos (PC)** surrogates work well with uncertain inputs
- **Bayesian regression** provides results with uncertainty certificate
- **Compressive sensing** ideas deal with high-dimensionality

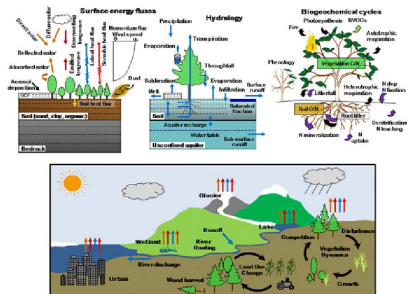
Application of Interest: Community Land Model



<http://www.cesm.ucar.edu/models/clm/>

- Nested computational grid hierarchy
- A single-site, 1000-yr simulation takes ~ 10 hrs on 1 CPU
- Involves ~ 70 input parameters; some dependent
- Non-smooth input-output relationship

Application of Interest: Community Land Model

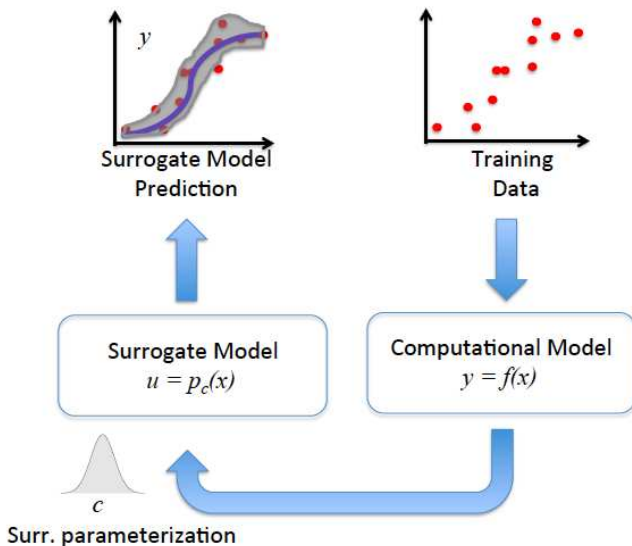


<http://www.cesm.ucar.edu/models/clm/>

UQ challenges:

- Computationally expensive
- High dimensionality
- Non-smooth/nonlinear behavior

Surrogate model construction



Surrogates are necessary for computationally expensive models

Construct surrogate for a complex model $f(\mathbf{x})$ to enable sampling-intensive studies:

- Global sensitivity analysis
 - Optimization
 - Uncertainty propagation (Forward UQ)
 - Input parameter calibration (Inverse UQ)
 - ...
-
- Computationally expensive model simulations, data sparsity
 - Need to build accurate surrogates with as few training runs as possible
 - High-dimensional input space
 - Too many samples needed to cover the space
 - Too many parameters in the surrogate parameterization

Surrogates are necessary for computationally expensive models

Construct surrogate for a complex model $f(\mathbf{x})$ to enable sampling-intensive studies:

- Global sensitivity analysis
- Optimization
- Uncertainty propagation (Forward UQ)
- Input parameter calibration (Inverse UQ)
- ...
- Computationally expensive model simulations, data sparsity
 - Need to build accurate surrogates with as few training runs as possible
- High-dimensional input space
 - Too many samples needed to cover the space
 - Too many parameters in the surrogate parameterization

Surrogates are necessary for computationally expensive models

Construct surrogate for a complex model $f(\mathbf{x})$ to enable sampling-intensive studies:

- Global sensitivity analysis
- Optimization
- Uncertainty propagation (Forward UQ)
- Input parameter calibration (Inverse UQ)
- ...
- Computationally expensive model simulations, data sparsity
 - Need to build accurate surrogates with as few training runs as possible
- High-dimensional input space
 - Too many samples needed to cover the space
 - Too many parameters in the surrogate parameterization

Random variables represented by Polynomial Chaos

$$Y \simeq \sum_{k=0}^{K-1} c_k \Psi_k(\boldsymbol{\xi})$$

- $\boldsymbol{\xi} = (\xi_1, \dots, \xi_d)$ standard i.i.d. r.v.
 Ψ_k standard polynomials, orthogonal w.r.t. $\pi(\boldsymbol{\xi})$.

$$\Psi_k(\xi_1, \xi_2, \dots, \xi_d) = \psi_{k_1}(\xi_1) \psi_{k_2}(\xi_2) \cdots \psi_{k_d}(\xi_d)$$

- Typical truncation rule: total-order p , $k_1 + k_2 + \dots + k_d \leq p$.
Number of terms is $K = \frac{(d+p)!}{d!p!}$.
- Essentially, a parameterization of a r.v. by deterministic spectral modes c_k .
- Most common standard Polynomial-Variable pairs:
(continuous) Gauss-Hermite, Legendre-Uniform,
(discrete) Poisson-Charlier.

[Wiener, 1938; Ghanem & Spanos, 1991; Xiu & Karniadakis, 2002; Le Maître & Knio, 2010]

Polynomial Chaos surrogate construction

- Scale the input parameters $x_i \in [a_i, b_i]$

$$x_i = \frac{a_i + b_i}{2} + \frac{b_i - a_i}{2} \xi_i$$

- Forward function $f(\cdot)$, output u

$$y = f(\mathbf{x}) \quad \approx \quad u = p(\mathbf{x}) \equiv \sum_{k=0}^{K-1} c_k \Psi_k(\boldsymbol{\xi})$$

- A lot of information for free:
 - Global sensitivity (Sobol) indices, variance-based decomposition
 - Moments of u , as a random variable

Alternative methods to obtain PC coefficients

$$y = f(x) \simeq \sum_{k=0}^{K-1} c_k \Psi_k(\mathbf{x})$$

- Projection $c_k = \frac{\langle f(\mathbf{x}) \Psi_k(\mathbf{x}) \rangle}{\langle \Psi_k^2(\mathbf{x}) \rangle}$

The integral $\langle f(\mathbf{x}) \Psi_k(\mathbf{x}) \rangle = \int f(\mathbf{x}) \Psi_k(\mathbf{x}) d\mathbf{x}$ can be estimated by

- Monte-Carlo

$$\frac{1}{N} \sum_{j=1}^N f(\mathbf{x}_j) \Psi_k(\mathbf{x}_j)$$



many(!) random samples

- Quadrature

$$\sum_{j=1}^Q f(\mathbf{x}_j) \Psi_k(\mathbf{x}_j) w_j$$



samples at quadrature

Alternative methods to obtain PC coefficients

$$y = f(x) \simeq \sum_{k=0}^{K-1} c_k \Psi_k(\mathbf{x})$$

- Projection $c_k = \frac{\langle f(\mathbf{x}) \Psi_k(\mathbf{x}) \rangle}{\langle \Psi_k^2(\mathbf{x}) \rangle}$

The integral $\langle f(\mathbf{x}) \Psi_k(\mathbf{x}) \rangle = \int f(\mathbf{x}) \Psi_k(\mathbf{x}) d\mathbf{x}$ can be estimated by

- Monte-Carlo

$$\frac{1}{N} \sum_{j=1}^N f(\mathbf{x}_j) \Psi_k(\mathbf{x}_j)$$



many(!) random samples

- Quadrature

$$\sum_{j=1}^Q f(\mathbf{x}_j) \Psi_k(\mathbf{x}_j) w_j$$



samples at quadrature

- Bayesian regression

$$P(c_k | f(\mathbf{x}_j)) \propto P(f(\mathbf{x}_j) | c_k) P(c_k)$$



any (number of) samples

Alternative methods to obtain PC coefficients

$$y = f(\mathbf{x}) \simeq \sum_{k=0}^{K-1} c_k \Psi_k(\mathbf{x})$$

- Projection $c_k = \frac{\langle f(\mathbf{x}) \Psi_k(\mathbf{x}) \rangle}{\langle \Psi_k^2(\mathbf{x}) \rangle}$

The integral $\langle f(\mathbf{x}) \Psi_k(\mathbf{x}) \rangle = \int f(\mathbf{x}) \Psi_k(\mathbf{x}) d\mathbf{x}$ can be estimated by

- Monte-Carlo

$$\frac{1}{N} \sum_{j=1}^N f(\mathbf{x}_j) \Psi_k(\mathbf{x}_j)$$



many(!) random samples

- Quadrature

$$\sum_{j=1}^Q f(\mathbf{x}_j) \Psi_k(\mathbf{x}_j) w_j$$



samples at quadrature

- Bayesian regression

$$\underbrace{P(\mathbf{c}|\mathcal{D})}_{\text{Posterior}} \propto \underbrace{P(\mathcal{D}|\mathbf{c})}_{\text{Likelihood}} \underbrace{P(\mathbf{c})}_{\text{Prior}}$$



any (number of) samples

Bayesian inference of PC surrogate

$$y = f(\mathbf{x}) \approx \sum_{k=0}^{K-1} c_k \Psi_k(\mathbf{x})$$

$$\underbrace{P(\mathbf{c}|\mathcal{D})}_{\text{Posterior}} \propto \underbrace{P(\mathcal{D}|\mathbf{c})}_{\text{Likelihood}} \underbrace{P(\mathbf{c})}_{\text{Prior}}$$

- Data consists of *training runs*

$$\mathcal{D} \equiv \{(\mathbf{x}_i, u_i)\}_{i=1}^N$$

- Likelihood with a gaussian noise model with σ^2 fixed or inferred,

$$L(\mathbf{c}) = P(\mathcal{D}|\mathbf{c}) = \left(\frac{1}{\sigma\sqrt{2\pi}}\right)^N \prod_{i=1}^N \exp\left(-\frac{(u_i - g\mathbf{c}(\mathbf{x}))^2}{2\sigma^2}\right)$$

- Prior on \mathbf{c} is chosen to be conjugate, uniform or gaussian.
- Posterior is a *multivariate normal*

$$\mathbf{c} \in \mathcal{MVN}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$$

- The (uncertain) surrogate is a *Gaussian process*

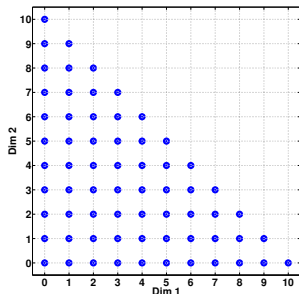
$$\sum_{k=0}^{K-1} c_k \Psi_k(\mathbf{x}) = \boldsymbol{\Psi}(\mathbf{x})^T \mathbf{c} \in \mathcal{GP}(\boldsymbol{\Psi}(\mathbf{x})^T \boldsymbol{\mu}, \boldsymbol{\Psi}(\mathbf{x}) \boldsymbol{\Sigma} \boldsymbol{\Psi}(\mathbf{x}')^T)$$

Bayesian inference of PC surrogate: low-data regime

$$y = f(\mathbf{x}) \approx \sum_{k=0}^{K-1} c_k \Psi_k(\mathbf{x})$$

$$\Psi_k(x_1, x_2, \dots, x_d) = \psi_{k_1}(x_1) \psi_{k_2}(x_2) \cdots \psi_{k_d}(x_d)$$

- Issues:
 - how to properly choose the basis set?
 - need to work in underdetermined regime $N < K$: fewer data than bases (d.o.f.)
- Discover the underlying low-d structure in the model
 - get help from the machine learning community

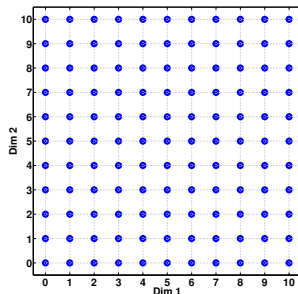


Bayesian inference of PC surrogate: low-data regime

$$y = f(\mathbf{x}) \approx \sum_{k=0}^{K-1} c_k \Psi_k(\mathbf{x})$$

$$\Psi_k(x_1, x_2, \dots, x_d) = \psi_{k_1}(x_1) \psi_{k_2}(x_2) \cdots \psi_{k_d}(x_d)$$

- Issues:
 - how to properly choose the basis set?
 - need to work in underdetermined regime
 $N < K$: fewer data than bases (d.o.f.)
- Discover the underlying low-d structure in the model
 - get help from the machine learning community

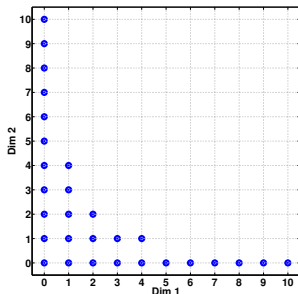


Bayesian inference of PC surrogate: low-data regime

$$y = f(\mathbf{x}) \approx \sum_{k=0}^{K-1} c_k \Psi_k(\mathbf{x})$$

$$\Psi_k(x_1, x_2, \dots, x_d) = \psi_{k_1}(x_1) \psi_{k_2}(x_2) \cdots \psi_{k_d}(x_d)$$

- Issues:
 - how to properly choose the basis set?
 - need to work in underdetermined regime
 $N < K$: fewer data than bases (d.o.f.)
- Discover the underlying low-d structure in the model
 - get help from the machine learning community

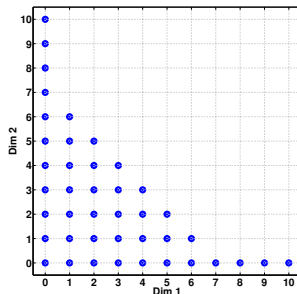


Bayesian inference of PC surrogate: low-data regime

$$y = f(\mathbf{x}) \approx \sum_{k=0}^{K-1} c_k \Psi_k(\mathbf{x})$$

$$\Psi_k(x_1, x_2, \dots, x_d) = \psi_{k_1}(x_1) \psi_{k_2}(x_2) \cdots \psi_{k_d}(x_d)$$

- Issues:
 - how to properly choose the basis set?
 - need to work in underdetermined regime $N < K$: fewer data than bases (d.o.f.)
- Discover the underlying low-d structure in the model
 - get help from the machine learning community

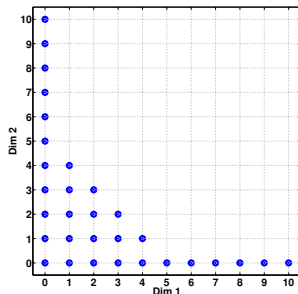


Bayesian inference of PC surrogate: low-data regime

$$y = f(\mathbf{x}) \approx \sum_{k=0}^{K-1} c_k \Psi_k(\mathbf{x})$$

$$\Psi_k(x_1, x_2, \dots, x_d) = \psi_{k_1}(x_1) \psi_{k_2}(x_2) \cdots \psi_{k_d}(x_d)$$

- Issues:
 - how to properly choose the basis set?
 - need to work in underdetermined regime
 $N < K$: fewer data than bases (d.o.f.)
- Discover the underlying low-d structure in the model
 - get help from the machine learning community



In a different language....

- N training data points (\mathbf{x}_n, y_n) and K basis terms $\Psi_k(\cdot)$
- Projection matrix $\mathbf{P}^{N \times K}$ with $\mathbf{P}_{nk} = \Psi_k(\mathbf{x}_n)$
- Find regression weights $\mathbf{c} = (c_0, \dots, c_{K-1})$ so that

$$\mathbf{y} \approx \mathbf{P}\mathbf{c}$$

or

$$y_n \approx \sum_k c_k \Psi_k(\mathbf{x}_n)$$

- The number of polynomial basis terms grows fast; a p -th order, d -dimensional basis has a total of $K = (p + d)!/(p!d!)$ terms.
- For limited data and large basis set ($N < K$) this is a sparse signal recovery problem \Rightarrow need some regularization/constraints.
- Least-squares $\operatorname{argmin}_{\mathbf{c}} \{ \|\mathbf{y} - \mathbf{P}\mathbf{c}\|_2 \}$
- The 'sparsest' $\operatorname{argmin}_{\mathbf{c}} \{ \|\mathbf{y} - \mathbf{P}\mathbf{c}\|_2 + \alpha \|\mathbf{c}\|_0 \}$
- Compressive sensing $\operatorname{argmin}_{\mathbf{c}} \{ \|\mathbf{y} - \mathbf{P}\mathbf{c}\|_2 + \alpha \|\mathbf{c}\|_1 \}$

In a different language....

- N training data points (\mathbf{x}_n, y_n) and K basis terms $\Psi_k(\cdot)$
- Projection matrix $\mathbf{P}^{N \times K}$ with $\mathbf{P}_{nk} = \Psi_k(\mathbf{x}_n)$
- Find regression weights $\mathbf{c} = (c_0, \dots, c_{K-1})$ so that

$$\mathbf{y} \approx \mathbf{P}\mathbf{c}$$

or

$$y_n \approx \sum_k c_k \Psi_k(\mathbf{x}_n)$$

- The number of polynomial basis terms grows fast; a p -th order, d -dimensional basis has a total of $K = (p + d)!/(p!d!)$ terms.
- For limited data and large basis set ($N < K$) this is a sparse signal recovery problem \Rightarrow need some regularization/constraints.
- Least-squares $\mathit{argmin}_{\mathbf{c}} \{ \|\mathbf{y} - \mathbf{P}\mathbf{c}\|_2 \}$
- The 'sparsest' $\mathit{argmin}_{\mathbf{c}} \{ \|\mathbf{y} - \mathbf{P}\mathbf{c}\|_2 + \alpha \|\mathbf{c}\|_0 \}$
- Compressive sensing $\mathit{argmin}_{\mathbf{c}} \{ \|\mathbf{y} - \mathbf{P}\mathbf{c}\|_2 + \alpha \|\mathbf{c}\|_1 \}$

In a different language....

- N training data points (\mathbf{x}_n, y_n) and K basis terms $\Psi_k(\cdot)$
- Projection matrix $\mathbf{P}^{N \times K}$ with $\mathbf{P}_{nk} = \Psi_k(\mathbf{x}_n)$
- Find regression weights $\mathbf{c} = (c_0, \dots, c_{K-1})$ so that

$$\mathbf{y} \approx \mathbf{P}\mathbf{c}$$

or

$$y_n \approx \sum_k c_k \Psi_k(\mathbf{x}_n)$$

- The number of polynomial basis terms grows fast; a p -th order, d -dimensional basis has a total of $K = (p + d)!/(p!d!)$ terms.
- For limited data and large basis set ($N < K$) this is a sparse signal recovery problem \Rightarrow need some regularization/constraints.
- Least-squares $\argmin_{\mathbf{c}} \{ \|\mathbf{y} - \mathbf{P}\mathbf{c}\|_2 \}$
- The 'sparsest' $\argmin_{\mathbf{c}} \{ \|\mathbf{y} - \mathbf{P}\mathbf{c}\|_2 + \alpha \|\mathbf{c}\|_0 \}$
- Compressive sensing $\argmin_{\mathbf{c}} \{ \|\mathbf{y} - \mathbf{P}\mathbf{c}\|_2 + \alpha \|\mathbf{c}\|_1 \}$

In a different language....

- N training data points (\mathbf{x}_n, y_n) and K basis terms $\Psi_k(\cdot)$
- Projection matrix $\mathbf{P}^{N \times K}$ with $\mathbf{P}_{nk} = \Psi_k(\mathbf{x}_n)$
- Find regression weights $\mathbf{c} = (c_0, \dots, c_{K-1})$ so that

$$\mathbf{y} \approx \mathbf{P}\mathbf{c}$$

or

$$y_n \approx \sum_k c_k \Psi_k(\mathbf{x}_n)$$

- The number of polynomial basis terms grows fast; a p -th order, d -dimensional basis has a total of $K = (p + d)!/(p!d!)$ terms.
- For limited data and large basis set ($N < K$) this is a sparse signal recovery problem \Rightarrow need some regularization/constraints.

- Least-squares $\operatorname{argmin}_{\mathbf{c}} \{ \|\mathbf{y} - \mathbf{P}\mathbf{c}\|_2 \}$

- The 'sparsest' $\operatorname{argmin}_{\mathbf{c}} \{ \|\mathbf{y} - \mathbf{P}\mathbf{c}\|_2 + \alpha \|\mathbf{c}\|_0 \}$

- Compressive sensing $\operatorname{argmin}_{\mathbf{c}} \{ \|\mathbf{y} - \mathbf{P}\mathbf{c}\|_2 + \alpha \|\mathbf{c}\|_1 \}$
Bayesian Likelihood Prior

Bayesian Compressive Sensing (BCS), or Relevance Vector Machine (RVM)

- Dimensionality reduction by using hierarchical priors

$$p(c_k|\sigma_k^2) = \frac{1}{\sqrt{2\pi}\sigma_k} e^{-\frac{c_k^2}{2\sigma_k^2}} \quad p(\sigma_k^2|\alpha) = \frac{\alpha}{2} e^{-\frac{\alpha\sigma_k^2}{2}}$$

- Effectively, one obtains Laplace *sparsity* prior

$$p(\mathbf{c}|\alpha) = \int \prod_{k=0}^{K-1} p(c_k|\sigma_k^2)p(\sigma_k^2|\alpha)d\sigma_k^2 = \prod_{k=0}^{K-1} \frac{\sqrt{\alpha}}{2} e^{-\sqrt{\alpha}|c_k|}$$

- The parameter α can be further modeled hierarchically, or fixed.
- Evidence maximization dictates values for $\sigma_k^2, \alpha, \sigma^2$ and allows exact Bayesian solution

$$\mathbf{c} \sim \mathcal{MVN}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$$

with

$$\boldsymbol{\mu} = \sigma^{-2} \boldsymbol{\Sigma} \mathbf{P}^T \mathbf{u} \quad \boldsymbol{\Sigma} = \sigma^2 (\mathbf{P}^T \mathbf{P} + \text{diag}(\sigma^2/\sigma_k^2))^{-1}$$

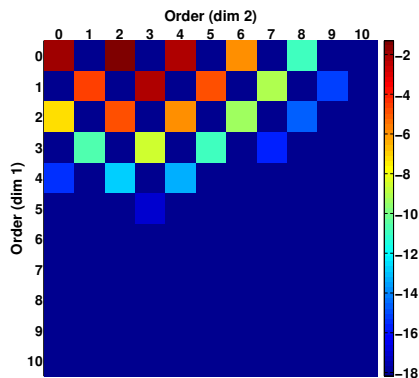
- KEY: Some $\sigma_k^2 \rightarrow 0$, hence the corresponding basis terms are dropped.

[Tipping, 2001, Ji *et al.*, 2008; Babacan *et al.*, 2010]

BCS removes unnecessary basis terms

$$f(\mathbf{x}) = \cos(x_1 + 4x_2)$$

$$f(\mathbf{x}) = \cos(x_1^2 + 4x_2)$$



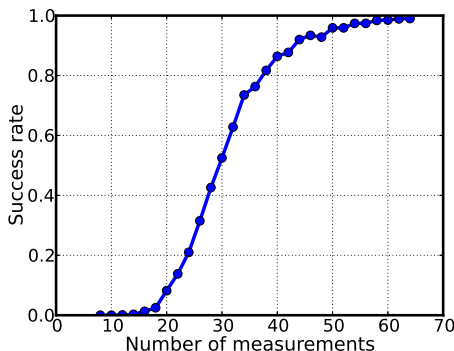
Success rate grows with more data and ‘sparser’ model

Consider test function

$$f(\mathbf{x}) = \sum_{k=0}^{K-1} c_k \Psi_k(\mathbf{x})$$

where only S coefficients c_k are non-zero. Typical setting is

$$S < N < K$$



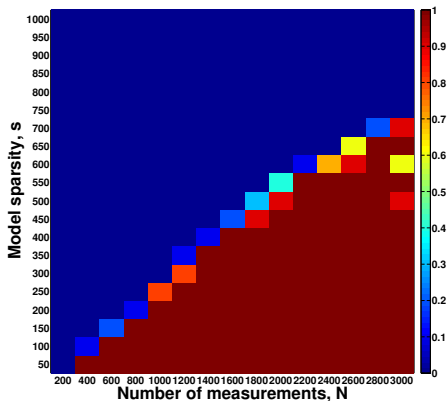
Success rate grows with more data and ‘sparser’ model

Consider test function

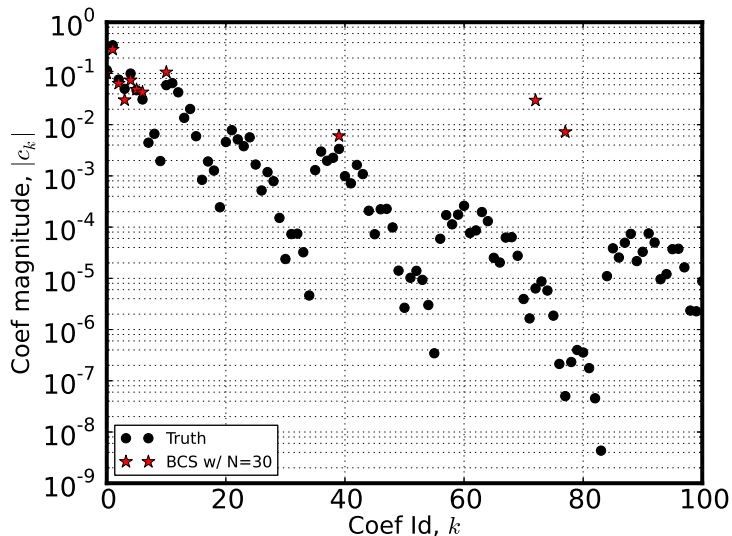
$$f(\mathbf{x}) = \sum_{k=0}^{K-1} c_k \Psi_k(\mathbf{x})$$

where only S coefficients c_k are non-zero. Typical setting is

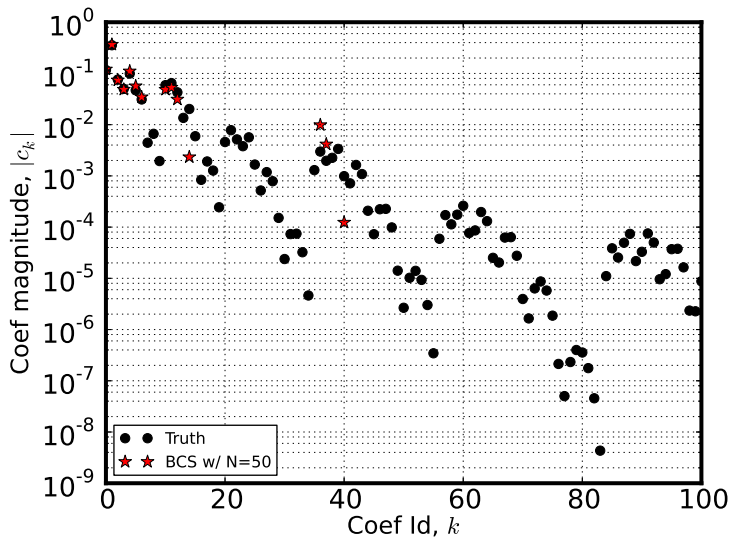
$$S < N < K$$



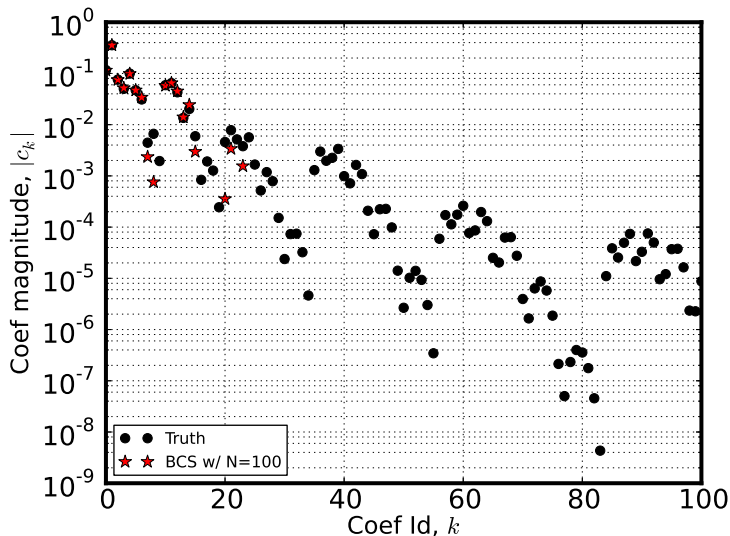
Recovering true PC coefficients



Recovering true PC coefficients

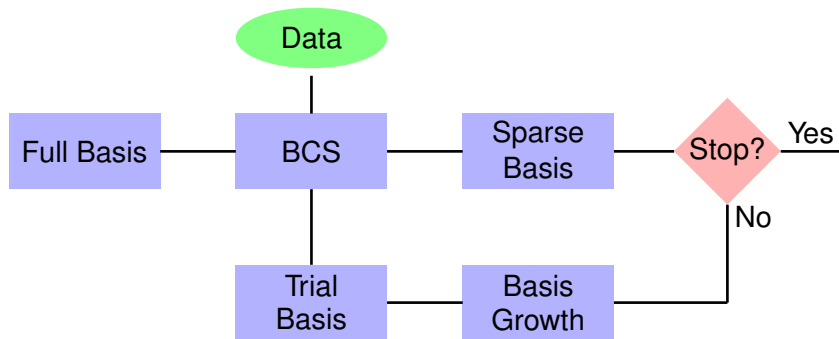


Recovering true PC coefficients



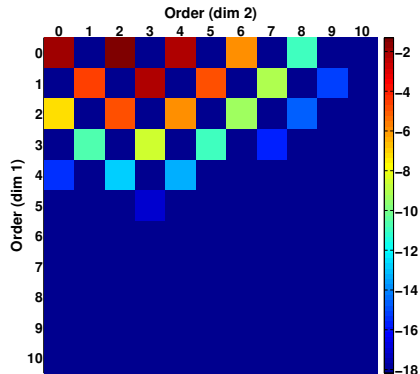
Iterative Bayesian Compressive Sensing (iBCS)

- *Iterative BCS*: We implement an iterative procedure that allows increasing the order for the relevant basis terms while maintaining the dimensionality reduction [Sargsyan *et al.* 2013].



Basis set growth with iterative BCS

$$f(\mathbf{x}) = \cos(x_1 + 4x_2)$$



Piecewise-PC expansion deals with nonlinearities:

use data classification methods

- Cluster the training dataset into non-overlapping subsets \mathcal{D}_1 and \mathcal{D}_2 , where the behavior of function is smoother
- Construct global PC expansions $g_i(\mathbf{x}) = \sum_k c_{ik} \Psi_k(\mathbf{x})$ using each dataset individually ($i = 1, 2$)
- Declare a surrogate

$$g_s(\mathbf{x}) = \begin{cases} g_1(\mathbf{x}) & \text{if } \mathbf{x} \in^* \mathcal{D}_1 \\ g_2(\mathbf{x}) & \text{if } \mathbf{x} \in^* \mathcal{D}_2 \end{cases}$$

* Requires a classification step to find out which cluster \mathbf{x} belongs to. We applied Random Decision Forests (RDF).

- Caveat: the sensitivity information is harder to obtain.

Illustration of piecewise PC construction

Global 5-th order surrogate fails

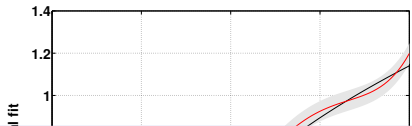
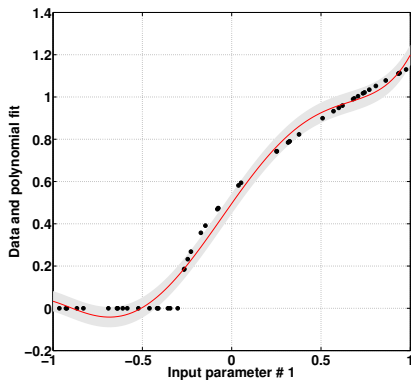


Illustration of piecewise PC construction

Piecewise 2-nd order surrogate

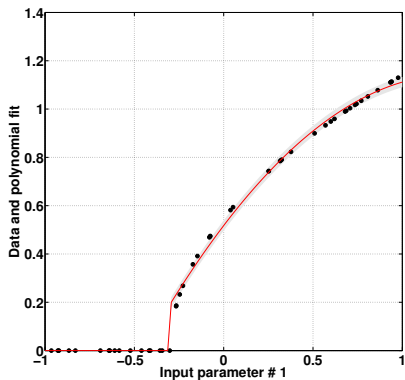


Illustration of piecewise PC construction

Piecewise 5-th order surrogate

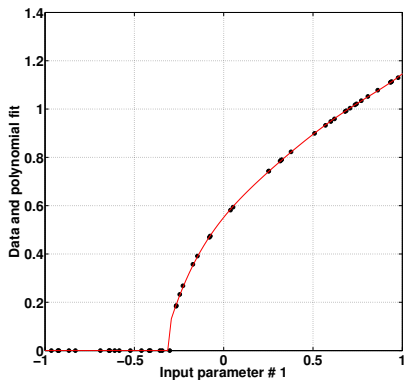


Illustration of piecewise PC construction

Piecewise 5-th order surrogate

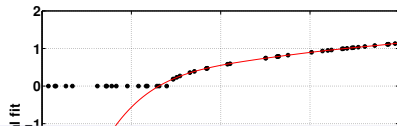
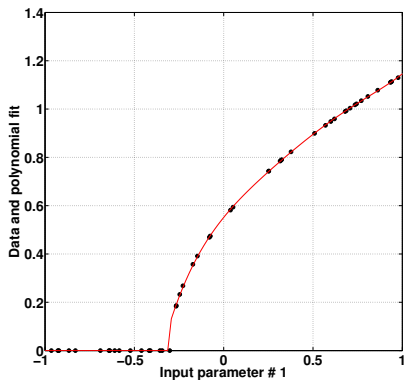


Illustration of piecewise PC construction

Piecewise 5-th order surrogate

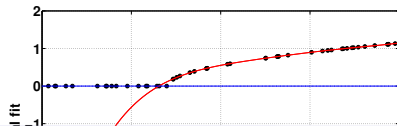
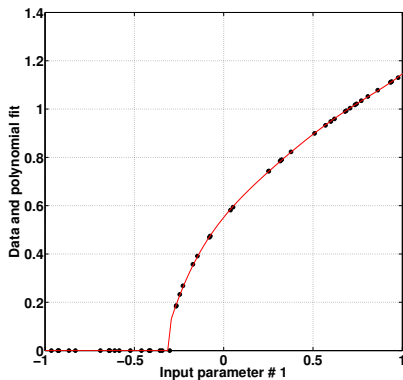
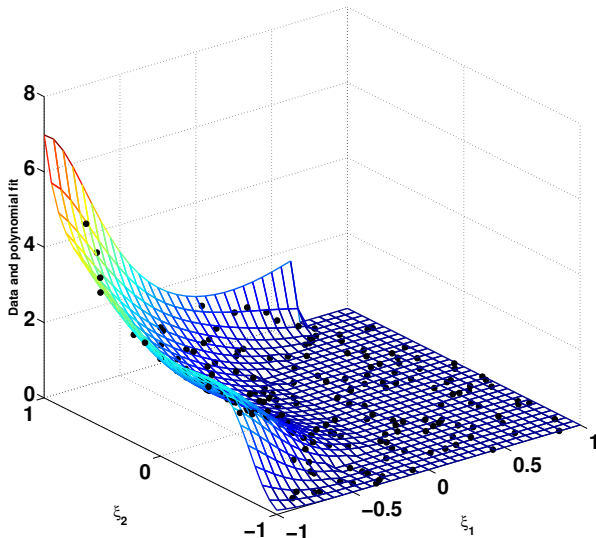
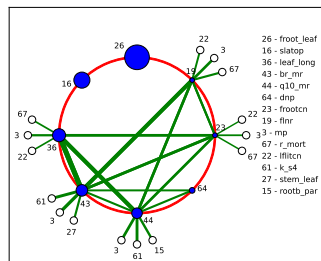
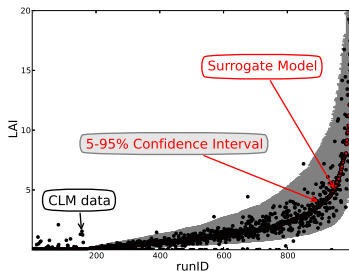


Illustration of piecewise PC construction



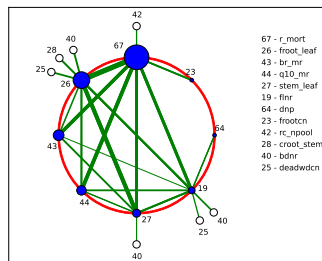
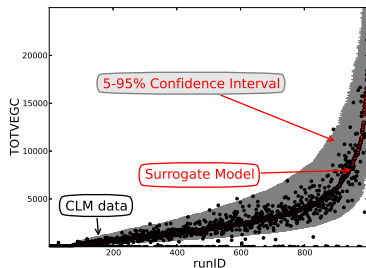
Sparse PC surrogate for the Community Land Model

- Main effect sensitivities : rank input parameters
- Joint sensitivities : most influential input couplings
- About 200 polynomial basis terms in the 70-dimensional space
- Sparse PC will further be used for
 - sampling in a reduced space
 - parameter calibration against experimental data



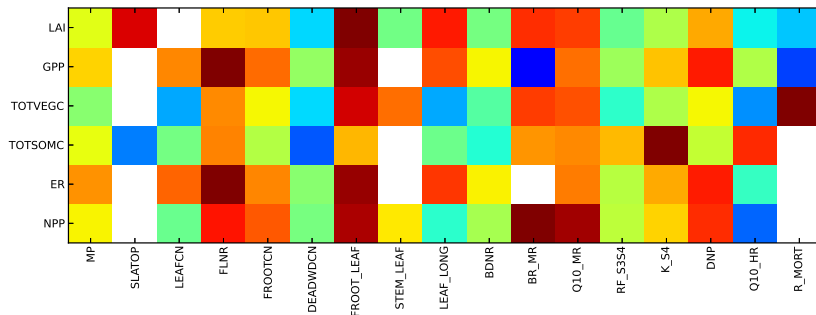
Sparse PC surrogate for the Community Land Model

- Main effect sensitivities : rank input parameters
- Joint sensitivities : most influential input couplings
- About 200 polynomial basis terms in the 70-dimensional space
- Sparse PC will further be used for
 - sampling in a reduced space
 - parameter calibration against experimental data



Sparse PC surrogate for the Community Land Model

- Main effect sensitivities : rank input parameters
- Joint sensitivities : most influential input couplings
- About 200 polynomial basis terms in the 70-dimensional space
- Sparse PC will further be used for
 - sampling in a reduced space
 - parameter calibration against experimental data



Summary

- Surrogate models are necessary for UQ studies of complex models
 - Replace the full model for both forward and inverse UQ
 - Uncertain inputs
 - Polynomial Chaos surrogates well-suited
 - Limited training dataset
 - Bayesian methods handle limited information well
 - Curse of dimensionality
 - The hope is that not too many dimensions matter
 - Compressive sensing (CS) ideas ported from machine learning
 - We implemented *iterative* Bayesian CS algorithm that reduces dimensionality and increases order on-the-fly.
 - Nonlinear behavior
 - Data clustering and classification-driven piecewise PC
-
- Future work, open issues
 - Computational design. What is the best sampling strategy?
 - Weighted l_1 minimization to accomodate natural coefficient decay.

Literature

- O. Le Maître and O. Knio, “Spectral Methods for Uncertainty Quantification with Applications to Computational Fluid Dynamics”, Springer, 2010.
- M. Tipping, “Sparse Bayesian learning and the relevance vector machine”, *J Machine Learning Research*, 1, pp. 211-244, 2001.
- S. Babacan, R. Molina and A. Katsaggelos, “Bayesian compressive sensing using Laplace priors”, *IEEE Trans. Image Proc.*, 19:1, 2010.
- A. Saltelli, “Making best use of model evaluations to compute sensitivity indices”, *Comp Phys Comm*, 145, 2002.
- K. Sargsyan, C. Safta, H. Najm, B. Debusschere, D. Ricciuto and P. Thornton, “Dimensionality reduction for complex models via Bayesian compressive sensing”, *Int J for Uncertainty Quantification*, in press, 2013.
- K. Sargsyan, C. Safta, R. Berry, J. Ray, B. Debusschere and H. Najm, “Efficient uncertainty quantification methodologies for high-dimensional climate land models”, Sandia Report, SAND2011-8757, Nov. 2011.

Thank You

Input correlations: Rosenblatt transformation

- Rosenblatt transformation maps any (not necessarily independent) set of random variables $\lambda = (\lambda_1, \dots, \lambda_d)$ to uniform i.i.d.'s $\{\eta_i\}_{i=1}^d$ [Rosenblatt, 1952].

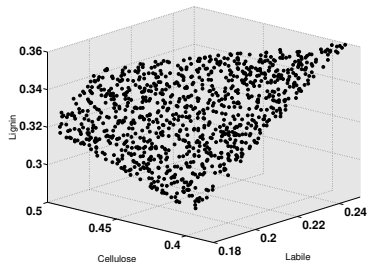
$$\eta_1 = F_1(\lambda_1)$$

$$\eta_2 = F_{2|1}(\lambda_2|\lambda_1)$$

$$\eta_3 = F_{3|2,1}(\lambda_3|\lambda_2, \lambda_1)$$

$$\vdots$$

$$\eta_d = F_{d|d-1,\dots,1}(\lambda_d|\lambda_{d-1}, \dots, \lambda_1)$$



- Inverse Rosenblatt transformation $\lambda = R^{-1}(\eta)$ ensures a well-defined input PC construction

$$\lambda_i = \sum_{k=0}^{K-1} \lambda_{ik} \Psi_k(\eta)$$

- Caveat: the conditional distributions are often hard to evaluate accurately.

Strong discontinuities/nonlinearities challenge global polynomial expansions

- Basis enrichment [Ghosh & Ghanem, 2005]
- Stochastic domain decomposition
 - Wiener-Haar expansions,
Multiblock expansions,
Multiwavelets, [Le Maître *et al*, 2004,2007]
 - also known as Multielement PC [Wan & Karniadakis, 2009]
- Smart splitting, discontinuity detection
[Archibald *et al*, 2009; Chantrasmi, 2011; Sargsyan *et al*, 2011; Jakeman *et al*, 2012]
- Data domain decomposition,
 - Mixture PC expansions [Sargsyan *et al*, 2010]
- Data clustering, classification,
 - Piecewise PC expansions

Sensitivity information comes free with PC surrogate,

$$g(x_1, \dots, x_d) = \sum_{k=0}^{K-1} c_k \Psi_k(\mathbf{x})$$

- Main effect sensitivity indices

$$S_i = \frac{\text{Var}[\mathbb{E}(g(\mathbf{x}|x_i))]}{\text{Var}[g(\mathbf{x})]} = \frac{\sum_{k \in \mathbb{I}_i} c_k^2 \|\Psi_k\|^2}{\sum_{k > 0} c_k^2 \|\Psi_k\|^2}$$

\mathbb{I}_i is the set of bases with only x_i involved

Sensitivity information comes free with PC surrogate,

$$g(x_1, \dots, x_d) = \sum_{k=0}^{K-1} c_k \Psi_k(\mathbf{x})$$

- Main effect sensitivity indices

$$S_i = \frac{\text{Var}[\mathbb{E}(g(\mathbf{x}|x_i))]}{\text{Var}[g(\mathbf{x})]} = \frac{\sum_{k \in \mathbb{I}_i} c_k^2 \|\Psi_k\|^2}{\sum_{k > 0} c_k^2 \|\Psi_k\|^2}$$

- Joint sensitivity indices

$$S_{ij} = \frac{\text{Var}[\mathbb{E}(g(\mathbf{x}|x_i, x_j))]}{\text{Var}[g(\mathbf{x})]} - S_i - S_j = \frac{\sum_{k \in \mathbb{I}_{ij}} c_k^2 \|\Psi_k\|^2}{\sum_{k > 0} c_k^2 \|\Psi_k\|^2}$$

\mathbb{I}_{ij} is the set of bases with only x_i and x_j involved

Sensitivity information comes free with PC surrogate, but not with piecewise PC

$$g(x_1, \dots, x_d) = \sum_{k=0}^{K-1} c_k \Psi_k(\mathbf{x})$$

- Main effect sensitivity indices

$$S_i = \frac{\text{Var}[\mathbb{E}(g(\mathbf{x}|x_i))]}{\text{Var}[g(\mathbf{x})]} = \frac{\sum_{k \in \mathbb{I}_i} c_k^2 \|\Psi_k\|^2}{\sum_{k > 0} c_k^2 \|\Psi_k\|^2}$$

- Joint sensitivity indices

$$S_{ij} = \frac{\text{Var}[\mathbb{E}(g(\mathbf{x}|x_i, x_j))]}{\text{Var}[g(\mathbf{x})]} - S_i - S_j = \frac{\sum_{k \in \mathbb{I}_{ij}} c_k^2 \|\Psi_k\|^2}{\sum_{k > 0} c_k^2 \|\Psi_k\|^2}$$

- For piecewise PC, need to resort to Monte-Carlo estimation
[\[Saltelli, 2002\]](#).