Sparse Polynomial Chaos Surrogate for ACME Land Model via Iterative Bayesian Compressive Sensing

SAND2015-10667C

K. Sargsyan[1], D. Ricciuto[2], P. Thornton[2]
C. Safta[1], B.Debusschere[1], H. Najm[1]

Sandia National Laboratories

Oak Ridge National Laboratory

[1]Sandia National Laboratories
Livermore, CA

[2]Oak Ridge National Laboratory
Oak Ridge, TN

ACME Accelerated Climate Modeling for Energy

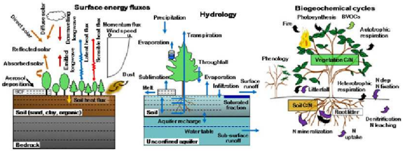
Sponsored by DOE, Biological and Environmental Research,
under Accelerated Climate Modeling for Energy (ACME).

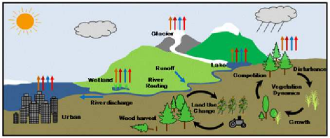
Sandia National Laboratories is a multi-program laboratory operated by Sandia Corporation,
a wholly owned subsidiary of Lockheed Martin Corporation,
for the U.S. Department of Energy's National Nuclear Security Administration
under contract DE-AC04-94AL85000.

OUTLINE

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

Application of Interest: ACME 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

Surrogate construction: scope and challenges

Construct surrogate for a complex model $f(\boldsymbol{\lambda})$ to enable

- Global sensitivity analysis
- Optimization
- Forward uncertainty propagation
- Input parameter calibration
- $\cdots$

- 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 terms in the polynomial expansion

Surrogate construction: scope and challenges

Construct surrogate for a complex model $f(\boldsymbol{\lambda})$ to enable

- Global sensitivity analysis
- Optimization
- Forward uncertainty propagation
- Input parameter calibration
- $\cdots$

- 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 terms in the polynomial expansion

Surrogate construction: scope and challenges

Construct surrogate for a complex model $f(\boldsymbol{\lambda})$ to enable

- Global sensitivity analysis
- Optimization
- Forward uncertainty propagation
- Input parameter calibration
- $\cdots$

- 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 terms in the polynomial expansion

Polynomial Chaos surrogate for $f(\boldsymbol{\lambda})$

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

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

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

$$u = f(\boldsymbol{\lambda}(\boldsymbol{x})) \quad \approx \quad \sum_{k=0}^{K-1} c_k \Psi_k(\boldsymbol{x}) \equiv g(\boldsymbol{x})$$

- Global sensitivity information for free
  - Sobol indices, variance-based decomposition.
- Bayesian inference useful for finding c_k:

$$P(c_k | u(\boldsymbol{x}_j)) \propto P(u(\boldsymbol{x}_j) | c_k) P(c_k)$$

Polynomial Chaos surrogate for $f(\boldsymbol{\lambda})$

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

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

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

$$u = f(\boldsymbol{\lambda}(\boldsymbol{x})) \qquad \approx \qquad \sum_{k=0}^{K-1} c_k \Psi_k(\boldsymbol{x}) \equiv g(\boldsymbol{x})$$

- Global sensitivity information for free
  - Sobol indices, variance-based decomposition.
- Bayesian inference useful for finding c_k:

$$P(c_k | u(\boldsymbol{x}_j)) \propto P(u(\boldsymbol{x}_j) | c_k) P(c_k)$$

Polynomial Chaos surrogate for $f(\boldsymbol{\lambda})$

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

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

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

$$u = f(\boldsymbol{\lambda}(\boldsymbol{x})) \qquad \approx \qquad \sum_{k=0}^{K-1} c_k \Psi_k(\boldsymbol{x}) \equiv g(\boldsymbol{x})$$

- Global sensitivity information for free
  - Sobol indices, variance-based decomposition.
- Bayesian inference useful for finding c_k:

$$P(c_k|u(\boldsymbol{x}_j)) \propto P(u(\boldsymbol{x}_j)|c_k)P(c_k)$$

Polynomial Chaos surrogate for $f(\boldsymbol{\lambda})$

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

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

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

$$u = f(\boldsymbol{\lambda}(\boldsymbol{x})) \qquad \approx \qquad \sum_{k=0}^{K-1} c_k \Psi_k(\boldsymbol{x}) \equiv g(\boldsymbol{x})$$

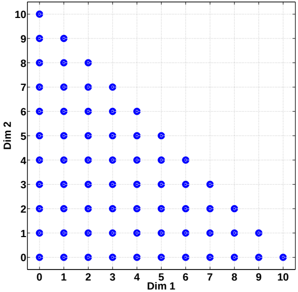
- Global sensitivity information for free
  - Sobol indices, variance-based decomposition.
- Bayesian inference useful for finding c_k:

$$P(c_k | u(\boldsymbol{x}_j)) \propto P(u(\boldsymbol{x}_j) | c_k) P(c_k)$$

Bayesian inference of PC surrogate: high-d, low-data regime

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

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



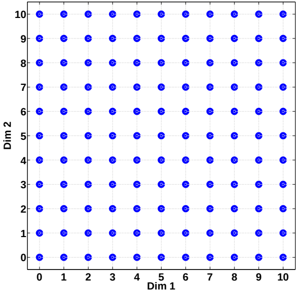
- Issues:
  - how to properly choose the basis set?
  - need to work in underdetermined regime $N < K$: fewer data than bases (d.o.f.)

- Discover the underlying low-d structure in the model
  - get help from the machine learning community

Bayesian inference of PC surrogate: high-d, low-data regime

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

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

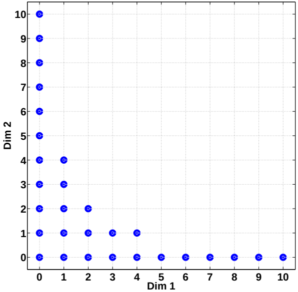


- Issues:
  - how to properly choose the basis set?
  - need to work in underdetermined regime $N < K$: fewer data than bases (d.o.f.)
- Discover the underlying low-d structure in the model
  - get help from the machine learning community

Bayesian inference of PC surrogate: high-d, low-data regime

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

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

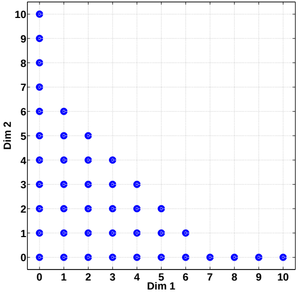


- Issues:
  - how to properly choose the basis set?
  - need to work in underdetermined regime $N < K$: fewer data than bases (d.o.f.)
- Discover the underlying low-d structure in the model
  - get help from the machine learning community

Bayesian inference of PC surrogate: high-d, low-data regime

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

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

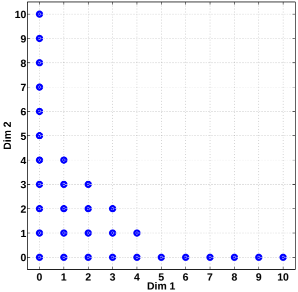


- Issues:
  - how to properly choose the basis set?
  - need to work in underdetermined regime $N < K$: fewer data than bases (d.o.f.)
- Discover the underlying low-d structure in the model
  - get help from the machine learning community

Bayesian inference of PC surrogate: high-d, low-data regime

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

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



- 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 $(\boldsymbol{x}_n, u_n)$ and K basis terms $\Psi_k(\cdot)$
- Projection matrix $\boldsymbol{P}^{N\times K}$ with $\boldsymbol{P}_{nk} = \Psi_k(\boldsymbol{x}_n)$
- Find regression weights $\boldsymbol{c} = (c_0, \ldots, c_{K-1})$ so that

$$\boldsymbol{u} \approx \boldsymbol{P}\boldsymbol{c} \qquad \text{or} \qquad u_n \approx \sum_k c_k \Psi_k(\boldsymbol{x}_n)$$

- 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 $\quad argmin_{\boldsymbol{c}} \{||\boldsymbol{u} - \boldsymbol{P}\boldsymbol{c}||_2\}$
- The 'sparsest' $\quad argmin_{\boldsymbol{c}} \{||\boldsymbol{u} - \boldsymbol{P}\boldsymbol{c}||_2 + \alpha||\boldsymbol{c}||_0\}$
- Compressive sensing $\quad argmin_{\boldsymbol{c}} \{||\boldsymbol{u} - \boldsymbol{P}\boldsymbol{c}||_2 + \alpha||\boldsymbol{c}||_1\}$

In a different language....

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

$$\boldsymbol{u} \approx \boldsymbol{P}\boldsymbol{c} \qquad \text{or} \qquad u_n \approx \sum_k c_k \Psi_k(\boldsymbol{x}_n)$$

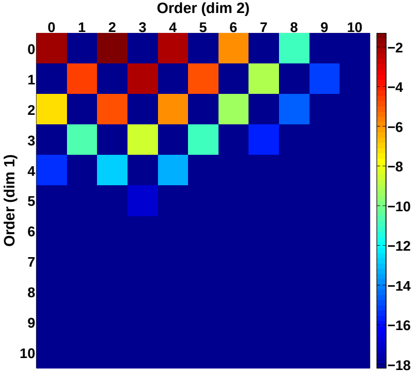
- The number of polynomial basis terms grows fast; a p-th order, d-dimensional basis has a total of $K = (p+d)!/(p!d!)$ terms.
- For limited data and large basis set ($N < K$) this is a sparse signal recovery problem $\Rightarrow$ need some regularization/constraints.

- Least-squares $\quad$ *argmin*$_{\boldsymbol{c}} \{||\boldsymbol{u} - \boldsymbol{P}\boldsymbol{c}||_2\}$
- The 'sparsest' $\quad$ *argmin*$_{\boldsymbol{c}} \{||\boldsymbol{u} - \boldsymbol{P}\boldsymbol{c}||_2 + \alpha||\boldsymbol{c}||_0\}$
- Compressive sensing $\quad$ *argmin*$_{\boldsymbol{c}} \{||\boldsymbol{u} - \boldsymbol{P}\boldsymbol{c}||_2 + \alpha||\boldsymbol{c}||_1\}$

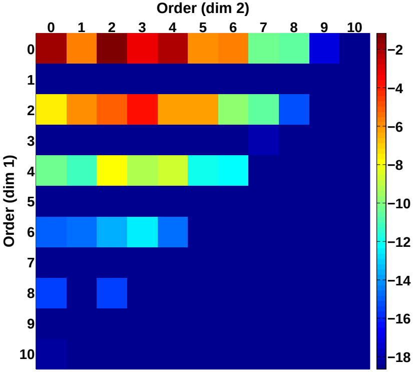
Bayesian $\quad$ Likelihood $\quad$ Prior

BCS removes unnecessary basis terms

$$f(x, y) = \cos(x + 4y)$$



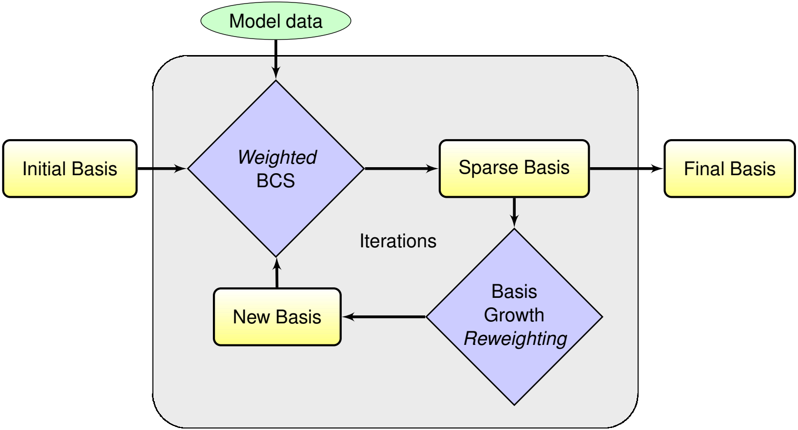
$$f(x, y) = \cos(x^2 + 4y)$$



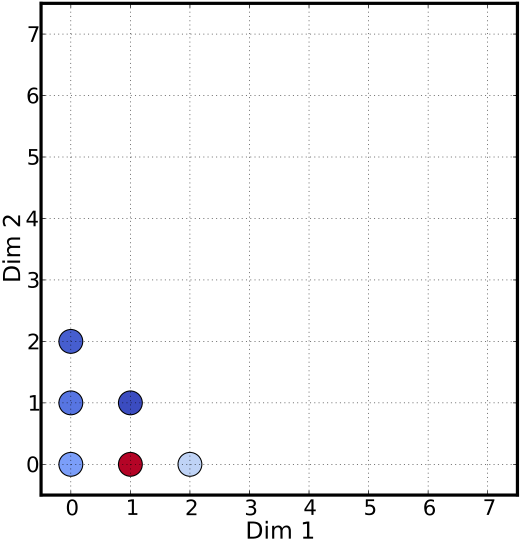
The square (i, j) represents the (log) spectral coefficient for the basis term $\psi_i(x)\psi_j(y)$.

Iterative Bayesian Compressive Sensing (iBCS)

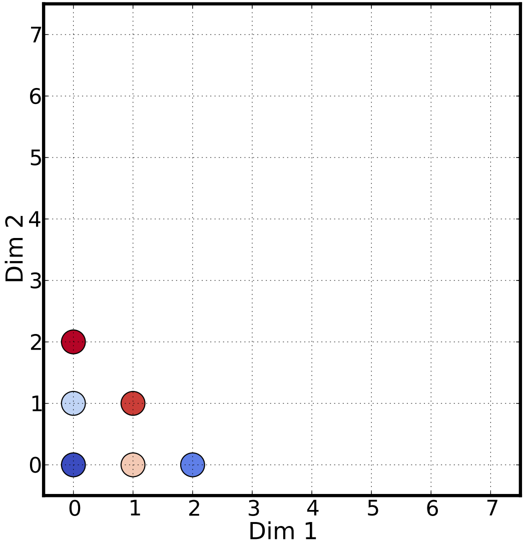
- *Iterative BCS*: We implement an iterative procedure that allows increasing the order for the relevant basis terms while maintaining the dimensionality reduction [Sargsyan *et al.* 2014], [Jakeman *et al.* 2015].
- Combine basis growth and reweighting!



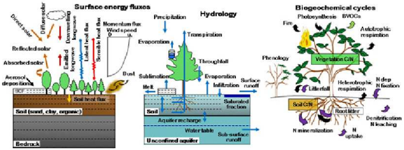
Basis set growth: simple anisotropic function

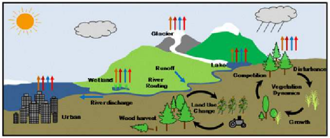


Basis set growth: ... added outlier term



Application of Interest: ACME Land Model

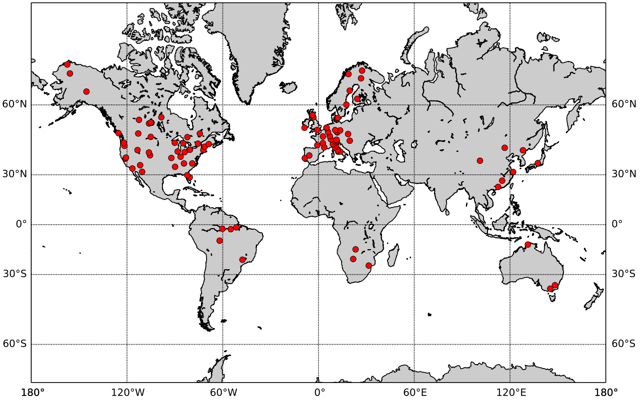




http://www.cesm.ucar.edu/models/clm/

- 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

FLUXNET experiment

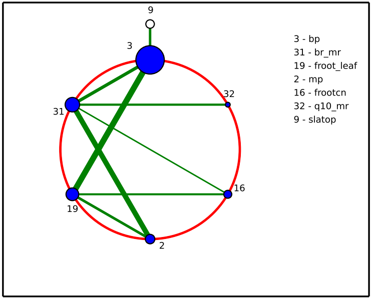


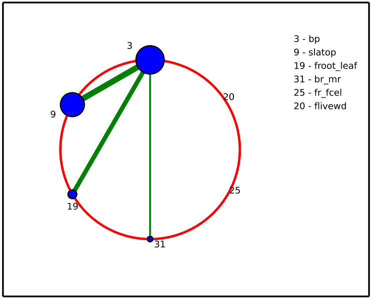
FLUXNET sites

- 96 FLUXNET sites covering major biomes and plant functional types
- Varying 68 input parameters over given ranges; 5 steady state outputs
- Ensemble of 3000 runs on Titan, DoE Leadership Computing Facility at Oak Ridge National Lab

Sparse PC surrogate and uncertainty decomposition for the ACME Land Model

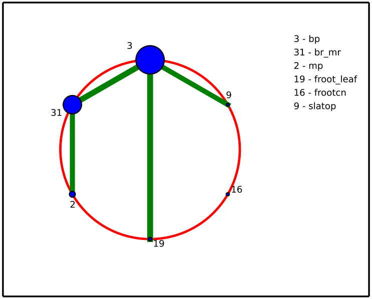
- Main effect sensitivities : rank input parameters
- Joint sensitivities : most influential input couplings
- About 200 polynomial basis terms in the 68-dimensional space
- Sparse PC will further be used for
  - sampling in a reduced space
  - parameter calibration against experimental data

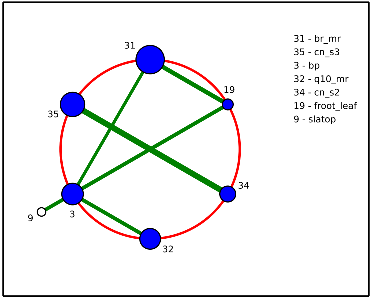




Sparse PC surrogate and uncertainty decomposition for the ACME Land Model

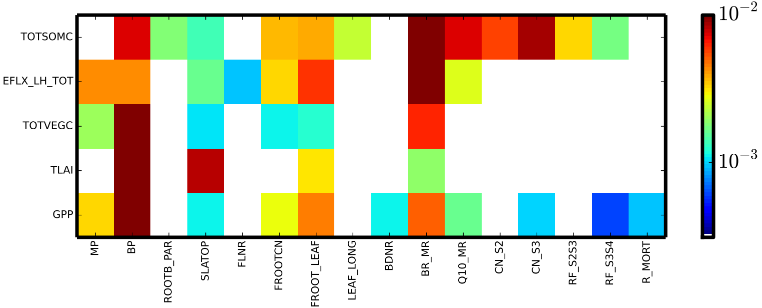
- Main effect sensitivities : rank input parameters
- Joint sensitivities : most influential input couplings
- About 200 polynomial basis terms in the 68-dimensional space
- Sparse PC will further be used for
  - sampling in a reduced space
  - parameter calibration against experimental data





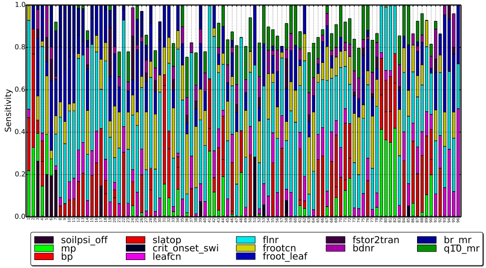
Sparse PC surrogate and uncertainty decomposition for the ACME Land Model

- Main effect sensitivities : rank input parameters
- Joint sensitivities : most influential input couplings
- About 200 polynomial basis terms in the 68-dimensional space
- Sparse PC will further be used for
  - sampling in a reduced space
  - parameter calibration against experimental data



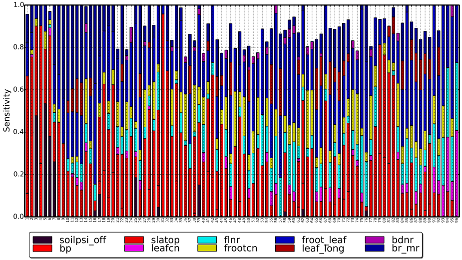
Sparse PC surrogate and uncertainty decomposition for the ACME Land Model

- Main effect sensitivities : rank input parameters
- Joint sensitivities : most influential input couplings
- About 200 polynomial basis terms in the 68-dimensional space
- Sparse PC will further be used for
  - sampling in a reduced space
  - parameter calibration against experimental data



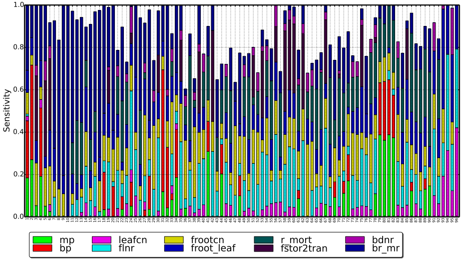
Sparse PC surrogate and uncertainty decomposition for the ACME Land Model

- Main effect sensitivities : rank input parameters
- Joint sensitivities : most influential input couplings
- About 200 polynomial basis terms in the 68-dimensional space
- Sparse PC will further be used for
  - sampling in a reduced space
  - parameter calibration against experimental data



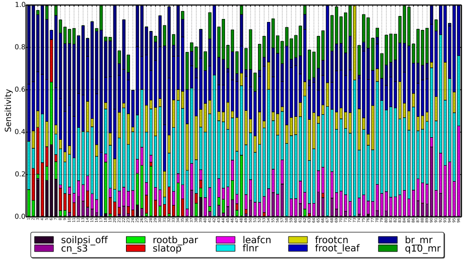
Sparse PC surrogate and uncertainty decomposition for the ACME Land Model

- Main effect sensitivities : rank input parameters
- Joint sensitivities : most influential input couplings
- About 200 polynomial basis terms in the 68-dimensional space
- Sparse PC will further be used for
  - sampling in a reduced space
  - parameter calibration against experimental data



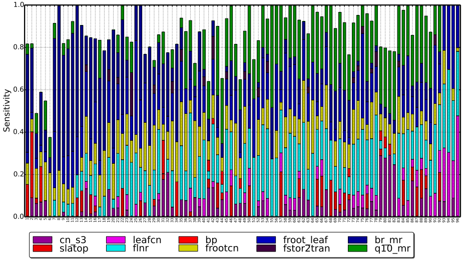
Sparse PC surrogate and uncertainty decomposition for the ACME Land Model

- Main effect sensitivities : rank input parameters
- Joint sensitivities : most influential input couplings
- About 200 polynomial basis terms in the 68-dimensional space
- Sparse PC will further be used for
  - sampling in a reduced space
  - parameter calibration against experimental data



Sparse PC surrogate and uncertainty decomposition for the ACME Land Model

- Main effect sensitivities : rank input parameters
- Joint sensitivities : most influential input couplings
- About 200 polynomial basis terms in the 68-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 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 *iteratively* reweighting Bayesian CS algorithm that reduces dimensionality and increases order on-the-fly.

- Open issues
  - Computational design. What is the best sampling strategy?
  - Overfitting still present. Cross-validation techniques help.

Literature

- M. Tipping, "Sparse Bayesian learning and the relevance vector machine", *J Machine Learning Research*, 1, pp. 211-244, 2001.
- S. Ji, Y. Xue and L. Carin, "Bayesian compressive sensing", *IEEE Trans. Signal Proc.*, 56:6, 2008.
- S. Babacan, R. Molina and A. Katsaggelos, "Bayesian compressive sensing using Laplace priors", *IEEE Trans. Image Proc.*, 19:1, 2010.
- E. J. Candes, M. Wakin and S. Boyd. "Enhancing sparsity by reweighted ℓ_1 minimization", *J. Fourier Anal. Appl.*, 14 877-905, 2007.
- 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*, 4(1), pp. 63-93, 2014.

- J. Jakeman, M. Eldred and K. Sargsyan, "Enhancing ℓ_1-minimization estimates of polynomial chaos expansions using basis selection", *J Comp Phys*, 289, pp. 18-34, 2015.

Random variables represented by Polynomial Chaos

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

- $\boldsymbol{\eta} = (\eta_1, \cdots, \eta_d)$ standard i.i.d. r.v.
  Ψ_k standard polynomials, orthogonal w.r.t. $\pi(\boldsymbol{\eta})$.

$$\Psi_k(\eta_1, \eta_2, \ldots, \eta_d) = \psi_{k_1}(\eta_1)\psi_{k_2}(\eta_2)\cdots\psi_{k_d}(\eta_d)$$

- Typical truncation rule: total-order p, $k_1 + k_2 + \ldots k_d \leq p$.
  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]

Bayesian inference of PC surrogate

$$u \simeq \sum_{k=0}^{K-1} c_k \Psi_k(\boldsymbol{x}) \equiv g_{\boldsymbol{c}}(\boldsymbol{x}) \qquad \overbrace{P(\boldsymbol{c}|\mathcal{D})}^{\text{Posterior}} \propto \overbrace{P(\mathcal{D}|\boldsymbol{c})}^{\text{Likelihood}} \overbrace{P(\boldsymbol{c})}^{\text{Prior}}$$

- Data consists of *training runs*

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

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

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

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

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

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

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

Sensitivity information comes free with PC surrogate,

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

- Main effect sensitivity indices

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

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

Sensitivity information comes free with PC surrogate,

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

- Main effect sensitivity indices

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

- Joint sensitivity indices

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

$\mathbb{I}_{ij}$ 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

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

- Main effect sensitivity indices

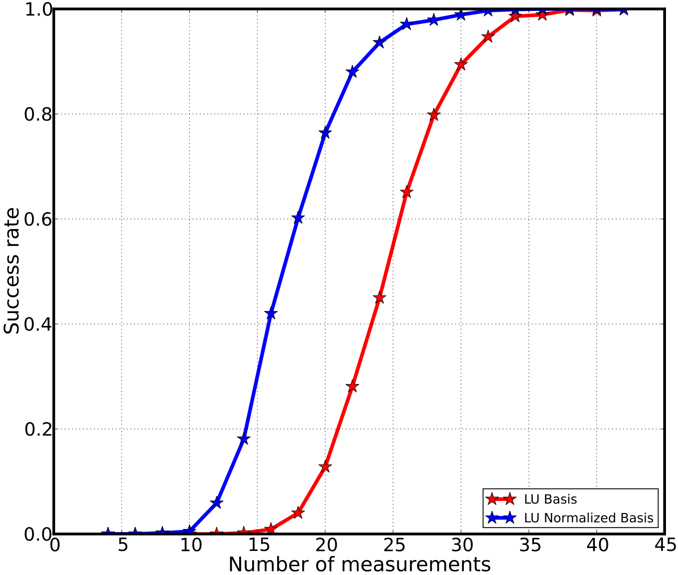
$$S_i=\frac{Var[\mathbb{E}(g(\boldsymbol{x}|x_i)]}{Var[g(\boldsymbol{x})]}=\frac{\sum_{k\in\mathbb{I}_i} c_k^2||\Psi_k||^2}{\sum_{k>0} c_k^2||\Psi_k||^2}$$

- Joint sensitivity indices

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

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

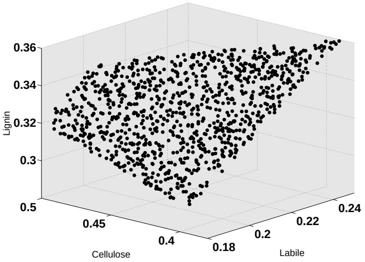
Basis normalization helps the success rate



Input correlations: Rosenblatt transformation

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

$$
\begin{aligned}
x_1 &= F_1(\lambda_1) \\
x_2 &= F_{2|1}(\lambda_2|\lambda_1) \\
x_3 &= F_{3|2,1}(\lambda_3|\lambda_2, \lambda_1) \\
&\vdots \\
x_d &= F_{d|d-1,\ldots,1}(\lambda_d|\lambda_{d-1}, \ldots, \lambda_1)
\end{aligned}
$$



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

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

- 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

Piecewise PC expansion with classification

- 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 $g_i(\boldsymbol{x}) = \sum_k c_{ik}\Psi_k(\boldsymbol{x})$ using each dataset individually ($i = 1, 2$)
- Declare a surrogate

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

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

- Caveat: the sensitivity information is harder to obtain.