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)

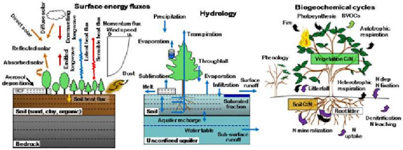
Thanks to DOE ASCR, DOE BER,
under Climate Science for Sustainable Energy Future (CSSEF)

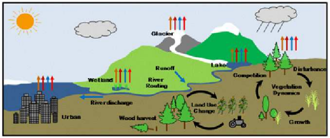
Sandia National Laboratories is a multi-program laboratory operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000

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

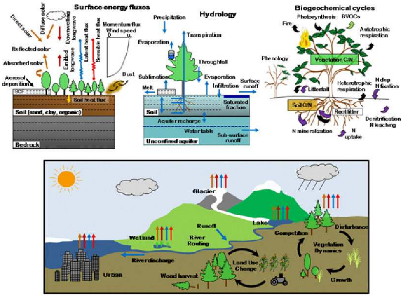




http://www.cesm.ucar.edu/models/clm/

- 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

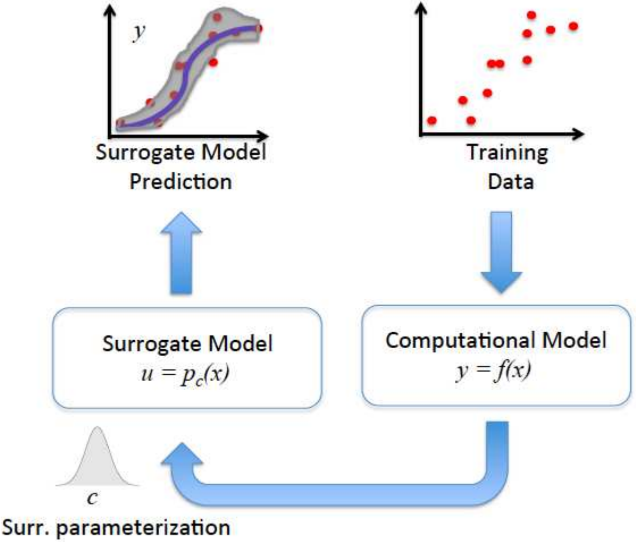


http://www.cesm.ucar.edu/models/clm/

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 $f(\boldsymbol{x})$ to enable sampling-intensive studies:

- Global sensitivity analysis
- Optimization
- Uncertainty propagation (Forward UQ)
- Input parameter calibration (Inverse UQ)
- $\cdots$

- 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 parameters in the surrogate parameterization

Surrogates are necessary for computationally expensive models

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

- Global sensitivity analysis
- Optimization
- Uncertainty propagation (Forward UQ)
- Input parameter calibration (Inverse UQ)
- $\cdots$

- 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 parameters in the surrogate parameterization

Surrogates are necessary for computationally expensive models

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

- Global sensitivity analysis
- Optimization
- Uncertainty propagation (Forward UQ)
- Input parameter calibration (Inverse UQ)
- $\cdots$

- 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 parameters in the surrogate parameterization

Random variables represented by Polynomial Chaos

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

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

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

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

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

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

- 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

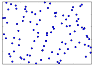
$$y = f(x) \simeq \sum_{k=0}^{K-1} c_k \Psi_k(\boldsymbol{x})$$

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

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

  - Monte-Carlo

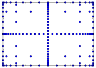
$$\frac{1}{N}\sum_{j=1}^{N} f(\boldsymbol{x}_j)\Psi_k(\boldsymbol{x}_j)$$



many(!) random samples

  - Quadrature

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



samples at quadrature

Alternative methods to obtain PC coefficients

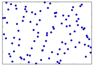
$$y = f(x) \simeq \sum_{k=0}^{K-1} c_k \Psi_k(\boldsymbol{x})$$

- <u>Projection</u> $\qquad c_k = \frac{\langle f(\boldsymbol{x})\Psi_k(\boldsymbol{x})\rangle}{\langle \Psi_k^2(\boldsymbol{x})\rangle}$

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

  - Monte-Carlo

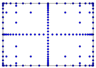
$$\frac{1}{N}\sum_{j=1}^{N} f(\boldsymbol{x}_j)\Psi_k(\boldsymbol{x}_j)$$



many(!) random samples

  - Quadrature

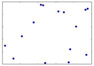
$$\sum_{j=1}^{Q} f(\boldsymbol{x}_j)\Psi_k(\boldsymbol{x}_j)w_j$$



samples at quadrature

- <u>Bayesian regression</u>

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



any (number of) samples

Alternative methods to obtain PC coefficients

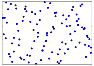
$$y = f(x) \simeq \sum_{k=0}^{K-1} c_k \Psi_k(\boldsymbol{x})$$

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

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

  - Monte-Carlo

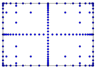
$$\frac{1}{N}\sum_{j=1}^{N} f(\boldsymbol{x}_j)\Psi_k(\boldsymbol{x}_j)$$



many(!) random samples

  - Quadrature

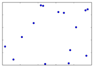
$$\sum_{j=1}^{Q} f(\boldsymbol{x}_j)\Psi_k(\boldsymbol{x}_j)w_j$$



samples at quadrature

- Bayesian regression

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



any (number of) samples

Bayesian inference of PC surrogate

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

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

- 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*

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

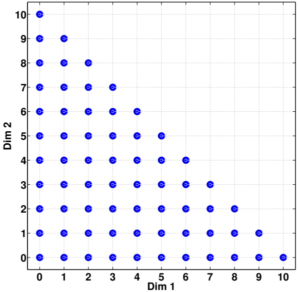
- The (uncertain) surrogate is *a Gaussian process*

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

Bayesian inference of PC surrogate: low-data regime

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

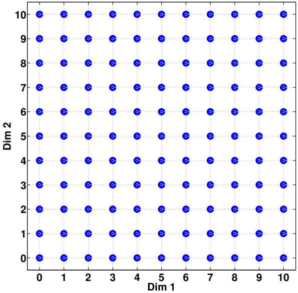
$$\Psi_k(x_1, x_2, ..., x_d) = \psi_{k_1}(x_1)\psi_{k_2}(x_2)\cdots\psi_{k_d}(x_d)$$



- 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

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

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

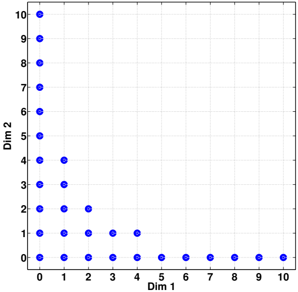


- Issues:
  - how to properly choose the basis set?
  - need to work in underdetermined regime $N < K$: fewer data than bases (d.o.f.)
- Discover the underlying low-d structure in the model
  - get help from the machine learning community

Bayesian inference of PC surrogate: low-data regime

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

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



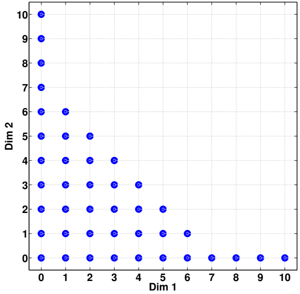
- Issues:
  - how to properly choose the basis set?
  - need to work in underdetermined regime $N < K$: fewer data than bases (d.o.f.)
- Discover the underlying low-d structure in the model
  - get help from the machine learning community

Bayesian inference of PC surrogate: low-data regime

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

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

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

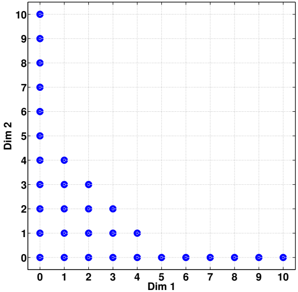


- Discover the underlying low-d structure in the model
  - get help from the machine learning community

Bayesian inference of PC surrogate: low-data regime

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

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



- 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 $(\boldsymbol{x}_n, y_n)$ 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 $\boldsymbol{c} = (c_0, \dots, c_{K-1})$ so that

$$\boldsymbol{y} \approx \boldsymbol{P}\boldsymbol{c} \quad \text{or} \quad y_n \approx \sum_k c_k \Psi_k(\boldsymbol{x}_n)$$

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

In a different language....

- N training data points $(\boldsymbol{x}_n, y_n)$ 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 $\boldsymbol{c} = (c_0, \dots, c_{K-1})$ so that

$$\boldsymbol{y} \approx \boldsymbol{Pc} \quad \text{or} \quad y_n \approx \sum_k c_k \Psi_k(\boldsymbol{x}_n)$$

- 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 *argmin*$_{\boldsymbol{c}}$ $\{||\boldsymbol{y} - \boldsymbol{Pc}||_2\}$
- The 'sparsest' *argmin*$_{\boldsymbol{c}}$ $\{||\boldsymbol{y} - \boldsymbol{Pc}||_2 + \alpha||\boldsymbol{c}||_0\}$
- Compressive sensing *argmin*$_{\boldsymbol{c}}$ $\{||\boldsymbol{y} - \boldsymbol{Pc}||_2 + \alpha||\boldsymbol{c}||_1\}$

In a different language....

- N training data points $(\boldsymbol{x}_n, y_n)$ 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

$$\boldsymbol{y} \approx \boldsymbol{P}\boldsymbol{c} \quad \text{or} \quad y_n \approx \sum_k c_k \Psi_k(\boldsymbol{x}_n)$$

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

In a different language....

- N training data points $(\boldsymbol{x}_n, y_n)$ 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

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

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

Bayesian $\quad$ Likelihood $\quad$ Prior

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

- Dimensionality reduction by using hierarchical priors

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

- Effectively, one obtains Laplace *sparsity* prior

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

- 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

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

  with

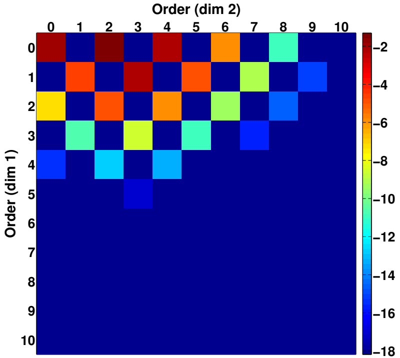
$$\boldsymbol{\mu} = \sigma^{-2}\boldsymbol{\Sigma}\boldsymbol{P}^T\boldsymbol{u} \qquad\qquad \boldsymbol{\Sigma} = \sigma^2(\boldsymbol{P}^T\boldsymbol{P} + \mathrm{diag}(\sigma^2/\sigma_k^2))^{-1}$$

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

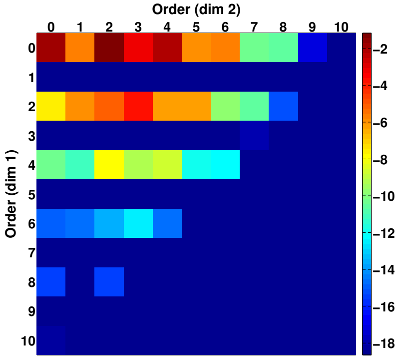
[Tipping, 2001, Ji *et al.*, 2008; Babacan *et al.*, 2010]

BCS removes unnecessary basis terms

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



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



The square (i,j) represents the (log) spectral coefficient for the basis term $\psi_i(x_1)\psi_j(x_2)$.

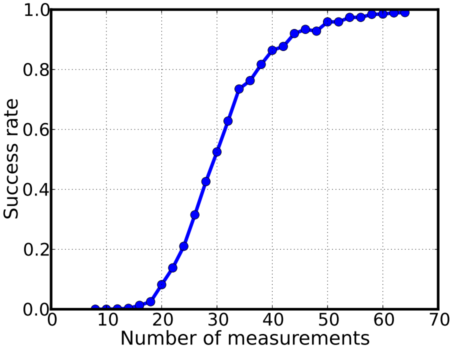
Success rate grows with more data and ‘sparser’ model

Consider test function

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

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

$$S < N < K$$



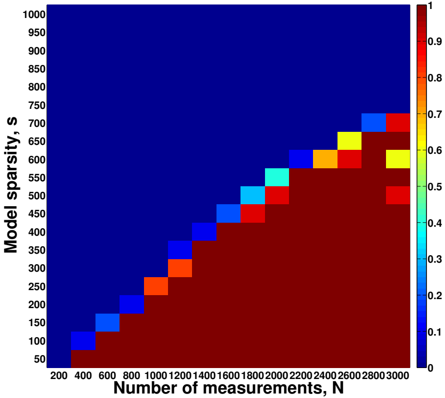
Success rate grows with more data and 'sparser' model

Consider test function

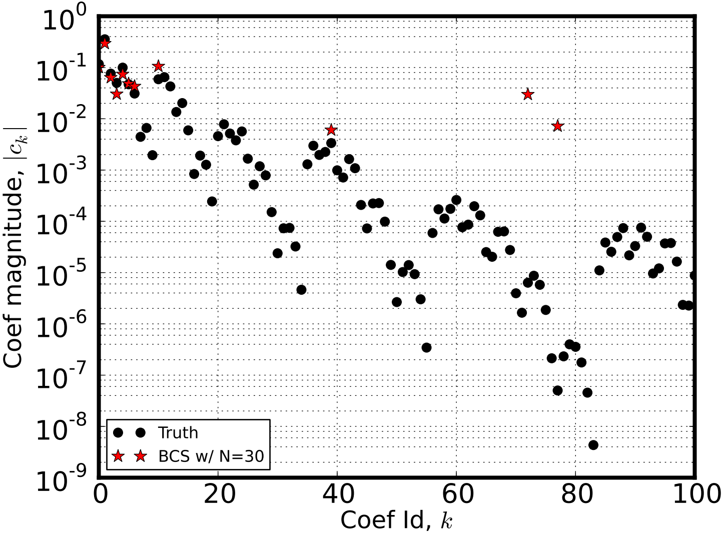
$$f(\boldsymbol{x}) = \sum_{k=0}^{K-1} c_k \Psi_k(\boldsymbol{x})$$

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

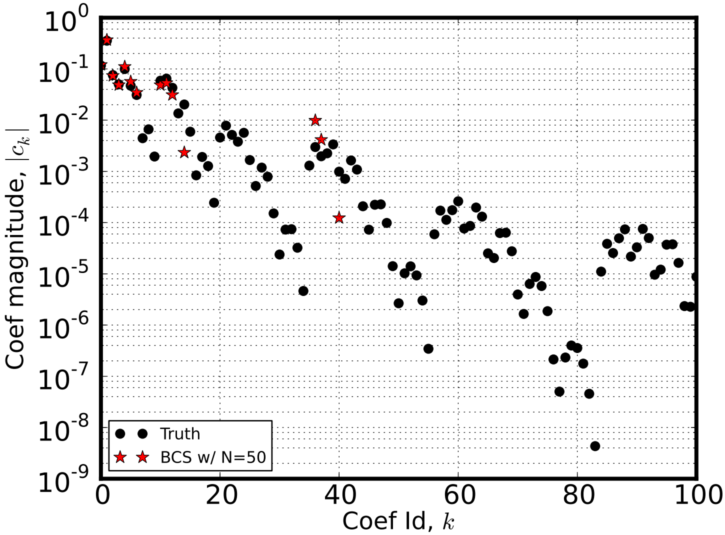
$$S < N < K$$



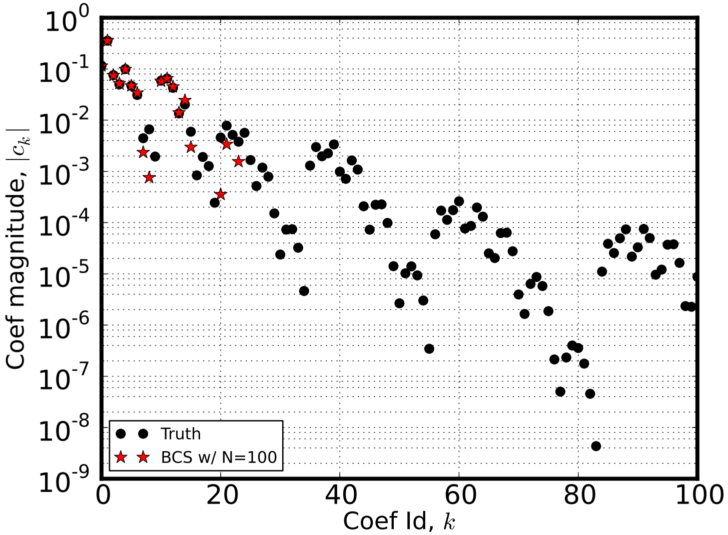
Recovering true PC coefficients



Recovering true PC coefficients

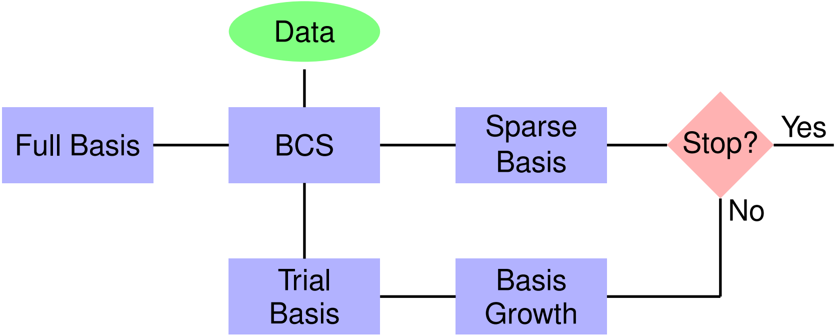


Recovering true PC coefficients



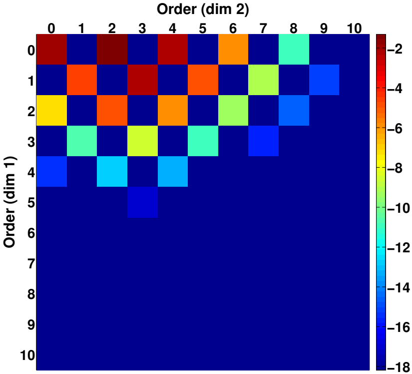
Iterative Bayesian Compressive Sensing (iBCS)

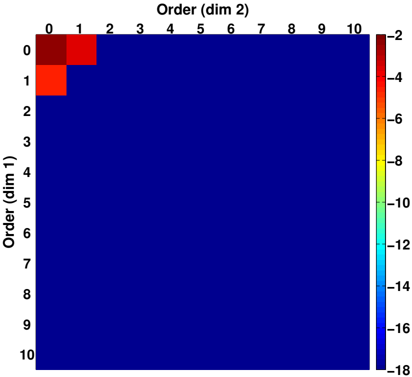
- *Iterative BCS*: We implement an iterative procedure that allows increasing the order for the relevant basis terms while maintaining the dimensionality reduction [Sargsyan *et al.* 2013].



Basis set growth with iterative BCS

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





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

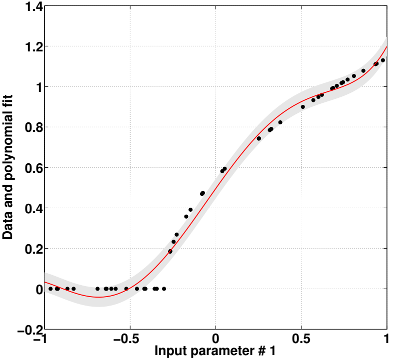
$$g_s(\boldsymbol{x}) = \begin{cases} g_1(\boldsymbol{x}) & \text{if } \boldsymbol{x} \in^* \mathcal{D}_1 \\ g_2(\boldsymbol{x}) & \text{if } \boldsymbol{x} \in^* \mathcal{D}_2 \end{cases}$$

* 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.

Illustration of piecewise PC construction

Global 5-th order surrogate fails



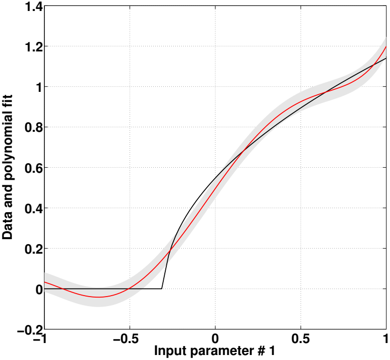


Illustration of piecewise PC construction

Piecewise 2-nd order surrogate

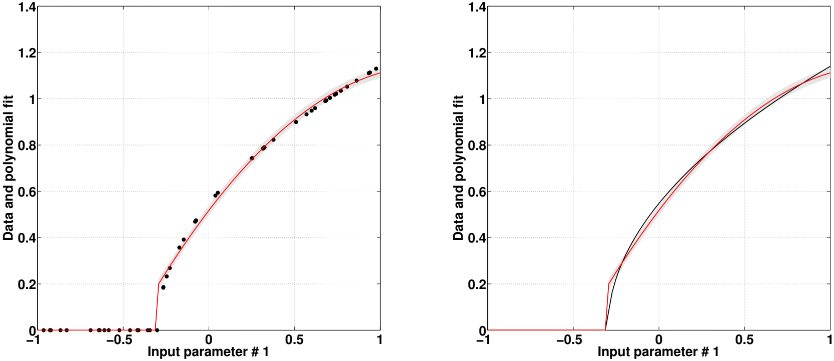
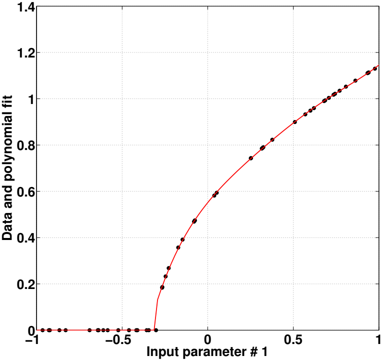


Illustration of piecewise PC construction

Piecewise 5-th order surrogate



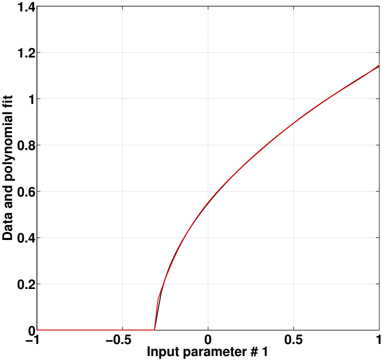


Illustration of piecewise PC construction

Piecewise 5-th order surrogate

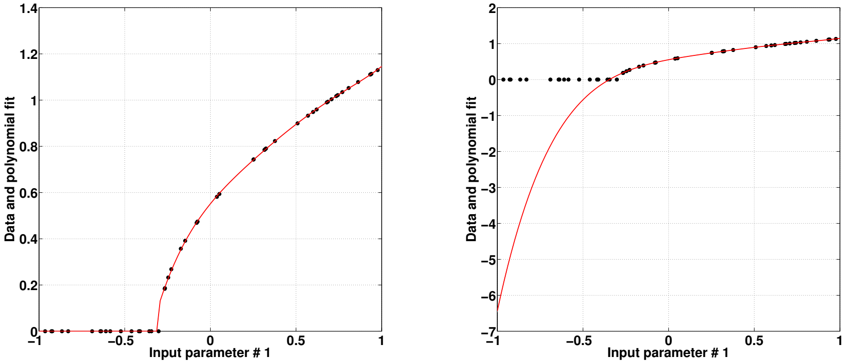
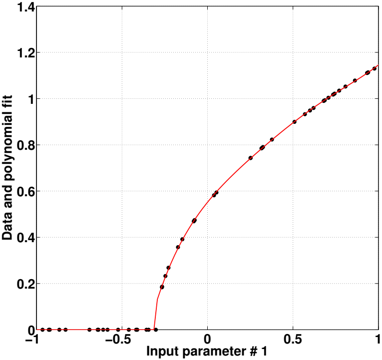


Illustration of piecewise PC construction

Piecewise 5-th order surrogate



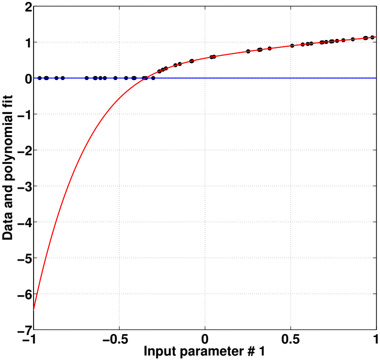
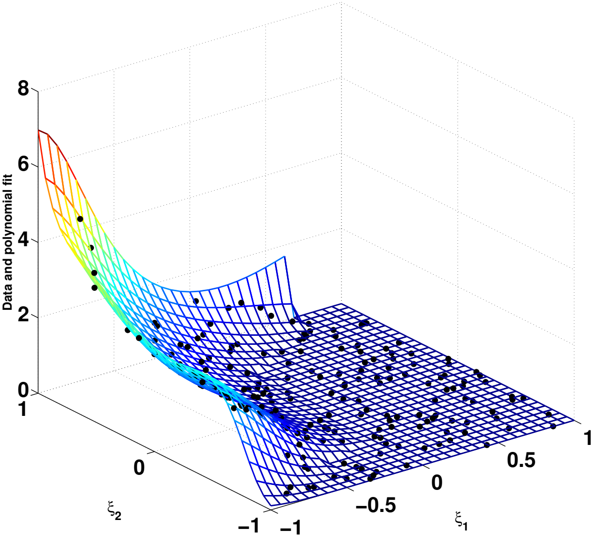
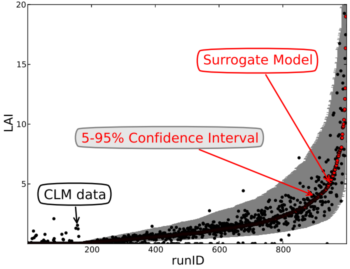


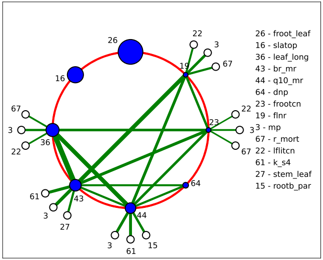
Illustration of piecewise PC construction



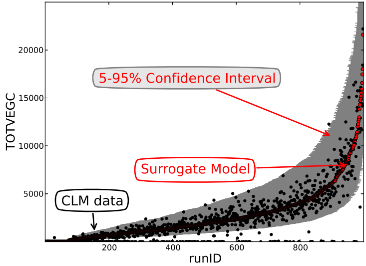
Sparse PC surrogate for the Community Land Model

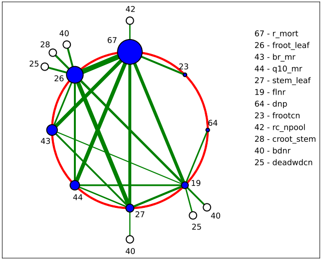
- Main effect sensitivities : rank input parameters
- Joint sensitivities : most influential input couplings
- About 200 polynomial basis terms in the 70-dimensional space
- Sparse PC will further be used for
  - sampling in a reduced space
  - parameter calibration against experimental data





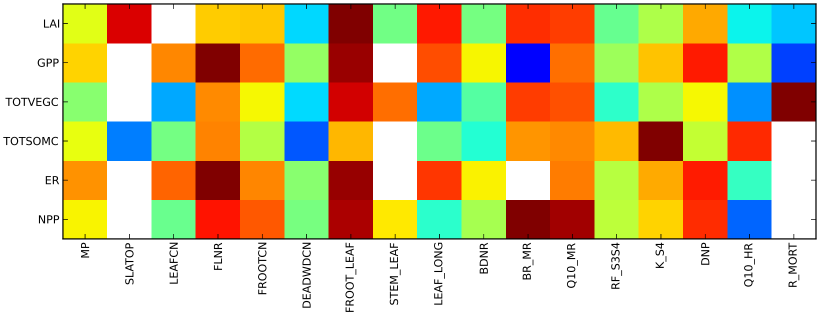
- Main effect sensitivities : rank input parameters
- Joint sensitivities : most influential input couplings
- About 200 polynomial basis terms in the 70-dimensional space
- Sparse PC will further be used for
    - sampling in a reduced space
    - parameter calibration against experimental data





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.

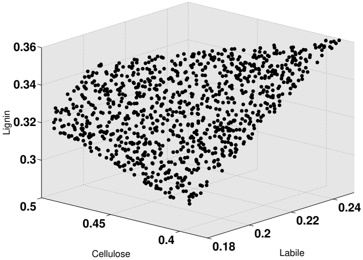
- K. Sargsyan, C. Safta, H. Najm, B. Debusschere, D. Ricciuto and P. Thornton, “Dimensionality reduction for complex models via Bayesian compressive sensing”, *Int J for Uncertainty Quantification*, in press, 2013.
- K. Sargsyan, C. Safta, R. Berry, J. Ray, B. Debusschere and H. Najm, “Efficient uncertainty 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 $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_d)$ to uniform i.i.d.'s $\{\eta_i\}_{i=1}^{d}$ [Rosenblatt, 1952].

$$
\begin{aligned}
\eta_1 &= F_1(\lambda_1) \\
\eta_2 &= F_{2|1}(\lambda_2|\lambda_1) \\
\eta_3 &= F_{3|2,1}(\lambda_3|\lambda_2, \lambda_1) \\
&\vdots \\
\eta_d &= F_{d|d-1,\ldots,1}(\lambda_d|\lambda_{d-1}, \ldots, \lambda_1)
\end{aligned}
$$



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

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

- 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,

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

- 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].