Surrogate construction via Bayesian compressive sensing for the Community Land Model

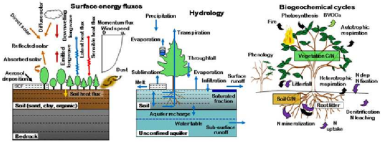
K. Sargsyan[1], C. Safta[1], D. Ricciuto[2],
B.Debusschere[1],H. Najm[1],P. Thornton[2]

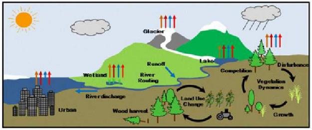
[1]Sandia National Laboratories
Livermore, CA

[2]Oak Ridge National Laboratory
Oak Ridge, TN

Sponsored by DOE, Biological and Environmental Research,
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.

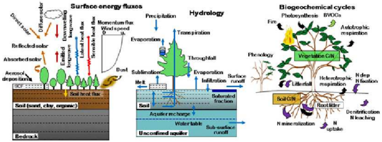


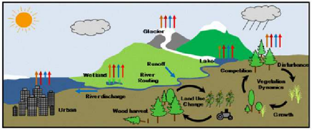


http://www.cesm.ucar.edu/models/clm/

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

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
- Strongly nonlinear input-output relationship

see [Cosmin Safta, Thu 2:35 pm, LMWG joint UQ]

Challenges we tackle

Construct surrogate for a complex model

- 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
- Input parameter correlations/dependences
- Strongly non-linear forward function

Challenges we tackle

Construct surrogate for a complex model

- 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
- Input parameter correlations/dependences
- Strongly non-linear forward function

Challenges we tackle

Construct surrogate for a complex model

- 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
- Input parameter correlations/dependences
- Strongly non-linear forward function

Challenges we tackle

Construct surrogate for a complex model

- 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
- Input parameter correlations/dependences
- Strongly non-linear forward function

Challenges we tackle

Construct surrogate for a complex model

- 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
- Input parameter correlations/dependences
- Strongly non-linear forward function

Challenges we tackle

Construct surrogate for a complex model

- 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
- Input parameter correlations/dependences
- Strongly non-linear forward function

- Global sensitivity analysis
- Optimization
- Forward uncertainty propagation
- Input parameter calibration

Polynomial Chaos surrogate construction

- Build/presume PC for input parameter λ

$$\boldsymbol{\lambda}(\boldsymbol{\eta}) = \sum_{k=0}^{K-1} \boldsymbol{a}_k \Psi_k(\boldsymbol{\eta})$$

with respect to multivariate Legendre polynomials.

Polynomial Chaos surrogate construction

- Build/presume PC for input parameter λ

$$\boldsymbol{\lambda}(\boldsymbol{\eta}) = \sum_{k=0}^{K-1} \boldsymbol{a}_k \Psi_k(\boldsymbol{\eta})$$

  with respect to multivariate Legendre polynomials.

- E.g., uniform on an interval, or gaussian with known moments,

$$\lambda = \lambda_0 + \lambda_1 \eta$$

Polynomial Chaos surrogate construction

- Build/presume PC for input parameter λ

$$\boldsymbol{\lambda}(\boldsymbol{\eta}) = \sum_{k=0}^{K-1} \boldsymbol{a}_k \Psi_k(\boldsymbol{\eta})$$

  with respect to multivariate Legendre polynomials.

- If input parameters are uniform $\lambda_i \sim \text{Uniform}[a_i, b_i]$, then

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

Polynomial Chaos surrogate construction

- Build/presume PC for input parameter λ

$$\boldsymbol{\lambda}(\boldsymbol{\eta}) = \sum_{k=0}^{K-1} \boldsymbol{a}_k \Psi_k(\boldsymbol{\eta})$$

with respect to multivariate Legendre polynomials.

- Input parameters are represented via their cumulative distribution function (CDF) $F(\cdot)$, such that, with $\eta_i \sim$ Uniform$[-1, 1]$

$$\lambda_i = F^{-1}_{\lambda_i}\left(\frac{\eta_i + 1}{2}\right), \qquad\qquad \text{for } i = 1, 2, \ldots, d.$$

Polynomial Chaos surrogate construction

- Build/presume PC for input parameter λ

$$\boldsymbol{\lambda}(\boldsymbol{\eta}) = \sum_{k=0}^{K-1} \boldsymbol{a}_k \Psi_k(\boldsymbol{\eta})$$

  with respect to multivariate Legendre polynomials.

- Input parameters are represented via their cumulative distribution function (CDF) $F(\cdot)$, such that, with $\eta_i \sim$ Uniform$[-1, 1]$

$$\lambda_i = F_{\lambda_i}^{-1}\left(\frac{\eta_i + 1}{2}\right), \qquad \text{for } i = 1, 2, \ldots, d.$$

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

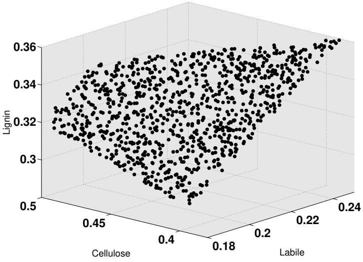
$$u = f(\boldsymbol{\lambda}(\boldsymbol{\eta})) \qquad u = \sum_{k=0}^{K-1} c_k \Psi_k(\boldsymbol{\eta}) \equiv g(\boldsymbol{\eta})$$

- Global sensitivity information for free
  - Sobol indices, variance-based decomposition.

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.

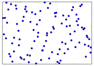
Alternative methods to obtain PC coefficients

$$u \simeq \sum_{k=0}^{K-1} c_k \Psi_k(\boldsymbol{\eta}) \qquad\qquad c_k = \frac{\langle u(\boldsymbol{\eta})\Psi_k(\boldsymbol{\eta})\rangle}{\langle \Psi_k^2(\boldsymbol{\eta})\rangle}$$

The integral $\langle u(\boldsymbol{\eta})\Psi_k(\boldsymbol{\eta})\rangle = \int u(\boldsymbol{\eta})\Psi_k(\boldsymbol{\eta})\pi(\boldsymbol{\eta})d\boldsymbol{\eta}$ can be estimated by

- Monte-Carlo

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



many samples from $\pi(\boldsymbol{\eta})$

- Quadrature

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

samples at quadrature

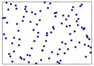
Alternative methods to obtain PC coefficients

$$u \simeq \sum_{k=0}^{K-1} c_k \Psi_k(\boldsymbol{\eta}) \qquad\qquad c_k = \frac{\langle u(\boldsymbol{\eta})\Psi_k(\boldsymbol{\eta})\rangle}{\langle \Psi_k^2(\boldsymbol{\eta})\rangle}$$

The integral $\langle u(\boldsymbol{\eta})\Psi_k(\boldsymbol{\eta})\rangle = \int u(\boldsymbol{\eta})\Psi_k(\boldsymbol{\eta})\pi(\boldsymbol{\eta})d\boldsymbol{\eta}$ can be estimated by

- Monte-Carlo

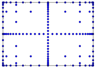
$$\frac{1}{N}\sum_{j=1}^{N} u(\boldsymbol{\eta}_j)\Psi_k(\boldsymbol{\eta}_j)$$



many samples from $\pi(\boldsymbol{\eta})$

- Quadrature

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



samples at quadrature

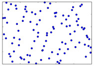
Alternative methods to obtain PC coefficients

$$u \simeq \sum_{k=0}^{K-1} c_k \Psi_k(\boldsymbol{\eta}) \qquad\qquad c_k = \frac{\langle u(\boldsymbol{\eta})\Psi_k(\boldsymbol{\eta})\rangle}{\langle \Psi_k^2(\boldsymbol{\eta})\rangle}$$

The integral $\langle u(\boldsymbol{\eta})\Psi_k(\boldsymbol{\eta})\rangle = \int u(\boldsymbol{\eta})\Psi_k(\boldsymbol{\eta})\pi(\boldsymbol{\eta})d\boldsymbol{\eta}$ can be estimated by

- Monte-Carlo

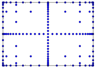
$$\frac{1}{N}\sum_{j=1}^{N} u(\boldsymbol{\eta}_j)\Psi_k(\boldsymbol{\eta}_j)$$



many samples from $\pi(\boldsymbol{\eta})$

- Quadrature

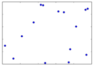
$$\sum_{j=1}^{Q} u(\boldsymbol{\eta}_j)\Psi_k(\boldsymbol{\eta}_j)w_j$$



samples at quadrature

- *Bayesian inference*

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



any (number of) samples

Bayesian inference of PC surrogate

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

- Data consists of *training runs*

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

Bayesian inference of PC surrogate

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

- Data consists of *training runs*

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

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

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

Bayesian inference of PC surrogate

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

- Data consists of *training runs*

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

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

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

- Prior on $\boldsymbol{c}$ is chosen to be conjugate, uniform or gaussian.

Bayesian inference of PC surrogate

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

- Data consists of *training runs*

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

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

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

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

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

Bayesian inference of PC surrogate

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

- Data consists of *training runs*

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

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

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

- 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*

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

In a different language....

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

$$\boldsymbol{u} \approx \boldsymbol{P}\boldsymbol{c}$$

- 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.
- Tikhonov regularization $\quad$ $\mathit{argmin}_{\boldsymbol{c}} \{||\boldsymbol{u} - \boldsymbol{P}\boldsymbol{c}||_2 + \alpha||\boldsymbol{c}||_2\}$
- Lasso regression $\quad$ $\mathit{argmin}_{\boldsymbol{c}} \{||\boldsymbol{u} - \boldsymbol{P}\boldsymbol{c}||_2\}$ subject to $||\boldsymbol{c}||_1 \leq \alpha$
- Compressive sensing $\quad$ $\mathit{argmin}_{\boldsymbol{c}} \{||\boldsymbol{u} - \boldsymbol{P}\boldsymbol{c}||_2 + \alpha||\boldsymbol{c}||_1\}$

In a different language....

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

$$\boldsymbol{u} \approx \boldsymbol{P}\boldsymbol{c}$$

- 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.

- Tikhonov regularization $\quad$ *argmin*$_{\boldsymbol{c}}$ $\{||\boldsymbol{u} - \boldsymbol{P}\boldsymbol{c}||_2 + \alpha||\boldsymbol{c}||_2\}$
- Lasso regression $\quad$ *argmin*$_{\boldsymbol{c}}$ $\{||\boldsymbol{u} - \boldsymbol{P}\boldsymbol{c}||_2\}$ subject to $||\boldsymbol{c}||_1 \leq \alpha$
- Compressive sensing $\quad$ *argmin*$_{\boldsymbol{c}}$ $\{||\boldsymbol{u} - \boldsymbol{P}\boldsymbol{c}||_2 + \alpha||\boldsymbol{c}||_1\}$

Bayesian $\quad$ Likelihood $\quad$ Prior

Bayesian Compressive Sensing (BCS)

- 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

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

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

Bayesian Compressive Sensing (BCS)

- 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

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

with

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

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

Bayesian Compressive Sensing (BCS)

- 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

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

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

Bayesian Compressive Sensing (BCS)

- 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

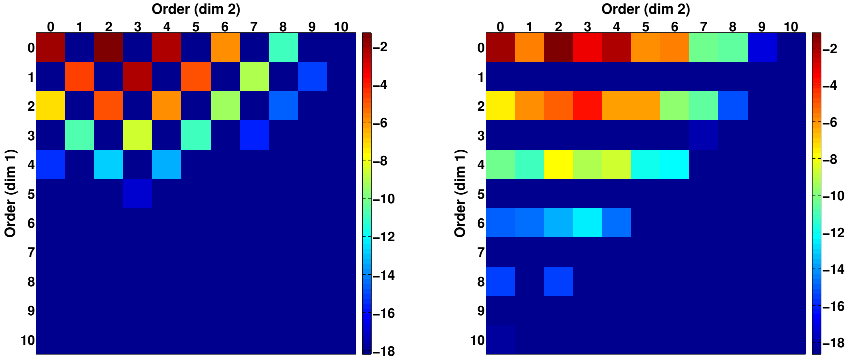
$$\boldsymbol{\mu} = \sigma^{-2}\boldsymbol{\Sigma}\boldsymbol{P}^T\boldsymbol{u} \qquad\qquad \boldsymbol{\Sigma} = \sigma^2(\boldsymbol{P}^T\boldsymbol{P} + \mathrm{diag}(\sigma^2/\sigma_k^2))^{-1}$$

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

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

BCS removes unnecessary basis terms

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



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

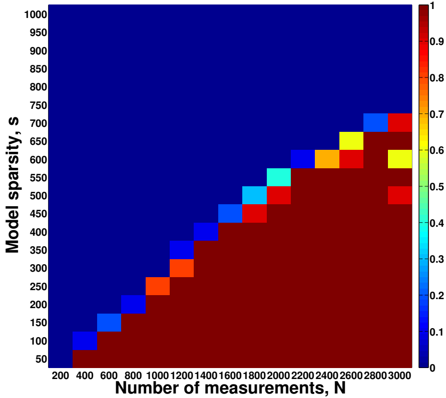
Success rate grows with more data and 'sparser' model

Consider test function

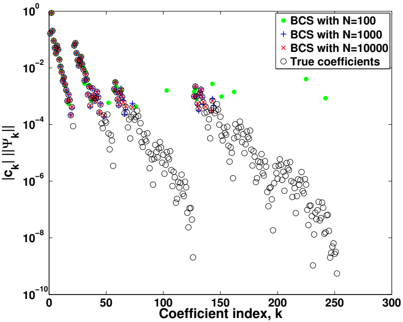
$$f(\boldsymbol{x}) = \sum_{k=0}^{K-1} c_k \Psi_k(\boldsymbol{x})$$

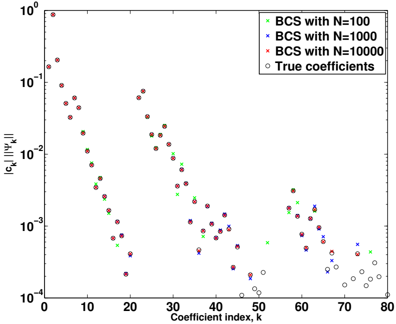
where only S coefficients c_k are non-zero. Typical setting is

$$S < N < K$$



BCS recovers true coefficients with increased number of measurements

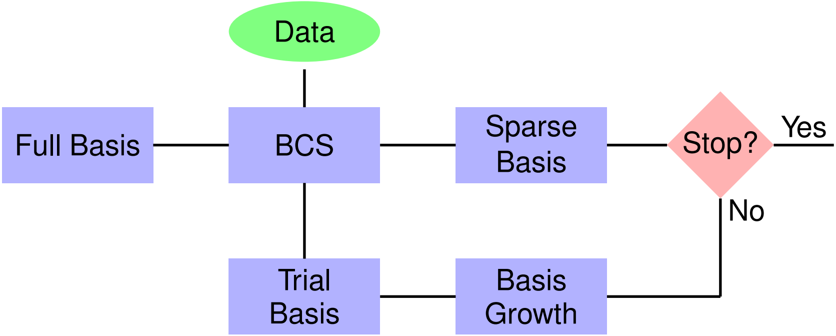




$$f(\boldsymbol{\xi}) = cos\left(2\pi e + \sum_{i=1}^{d} a_i \xi_i\right)$$, for $a_i = i^{-2}$ and $d = 5$.

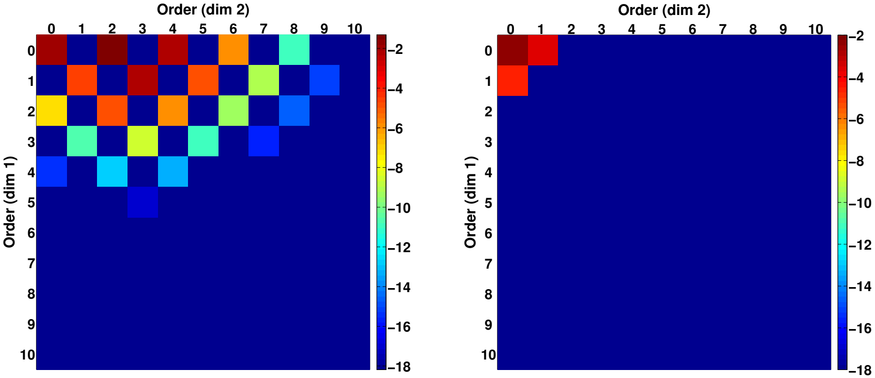
Iterative Bayesian Compressive Sensing (iBCS)

- *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

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

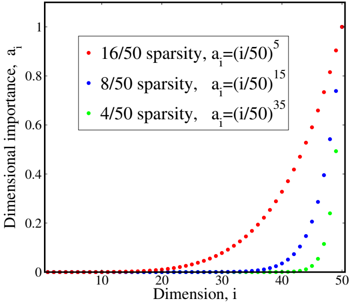


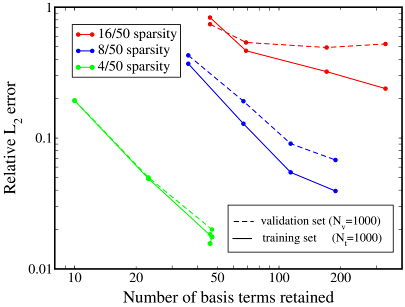
The fewer dimensions matter, the better

$$f(\boldsymbol{x}) = \exp\left(\sum_{i=1}^{d} a_i x_i\right)$$

Dimensionality importance coefficients are chosen so that 90% of energy is in a small subset of dimensions

Validation error increase indicates overfitting. $N_t = 1000$ training runs are sufficient if ~ 10 dimensions matter.





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

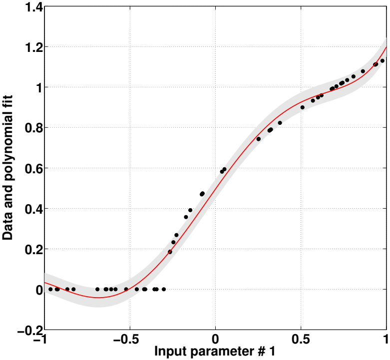
$$g_s(\boldsymbol{x}) = \begin{cases} g_1(\boldsymbol{x}) & \text{if } \boldsymbol{x} \in^* \mathcal{D}_1 \\ g_2(\boldsymbol{x}) & \text{if } \boldsymbol{x} \in^* \mathcal{D}_2 \end{cases}$$

* 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.

Illustration of piecewise PC construction

Global 5-th order surrogate fails



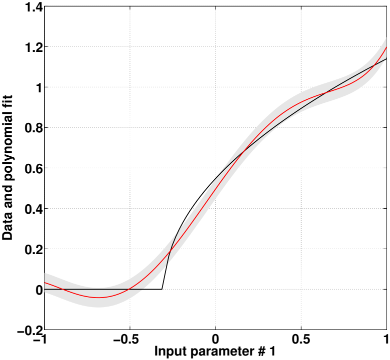


Illustration of piecewise PC construction

Piecewise 2-nd order surrogate

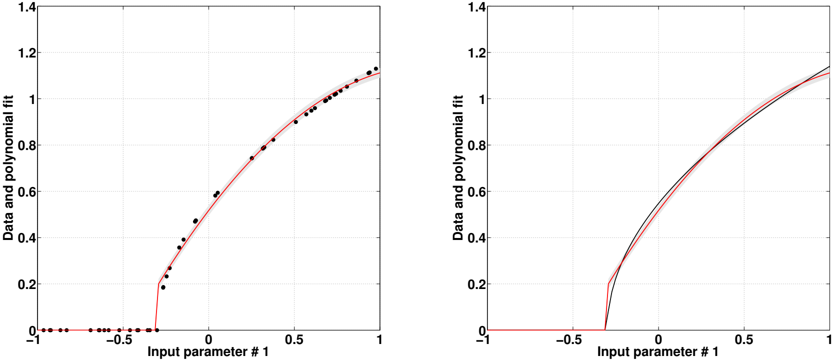


Illustration of piecewise PC construction

Piecewise 5-th order surrogate

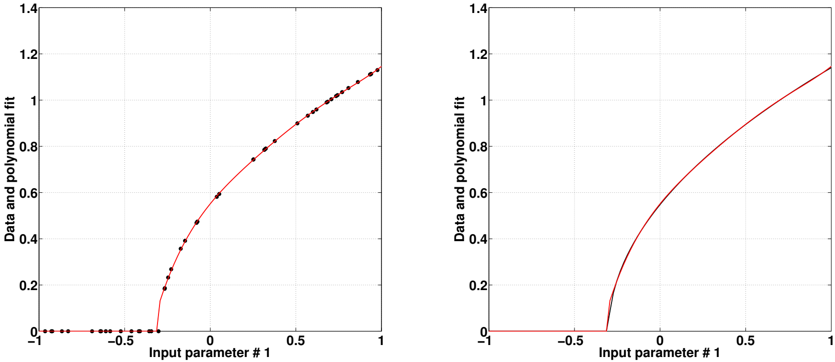


Illustration of piecewise PC construction

Piecewise 5-th order surrogate

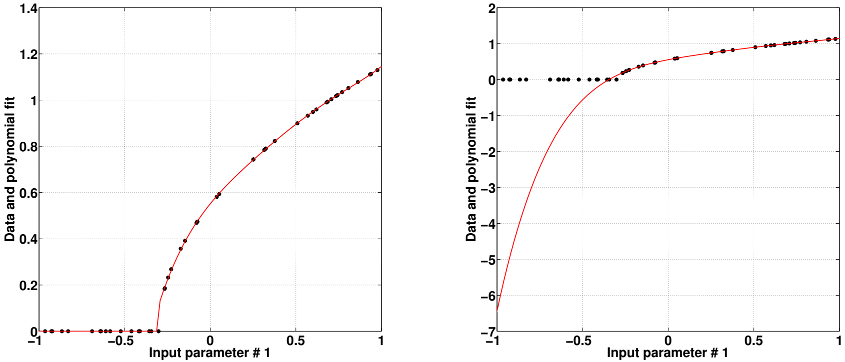
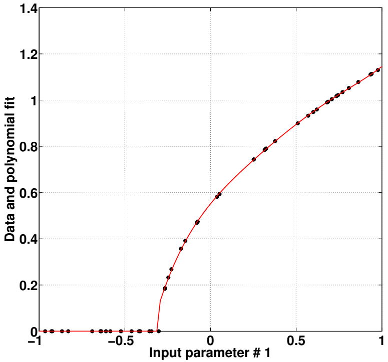


Illustration of piecewise PC construction

Piecewise 5-th order surrogate



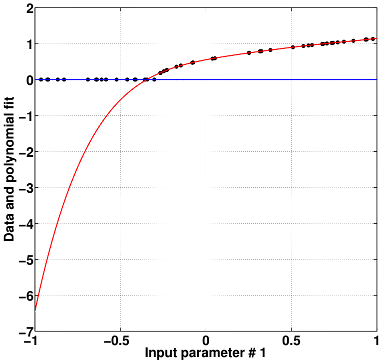
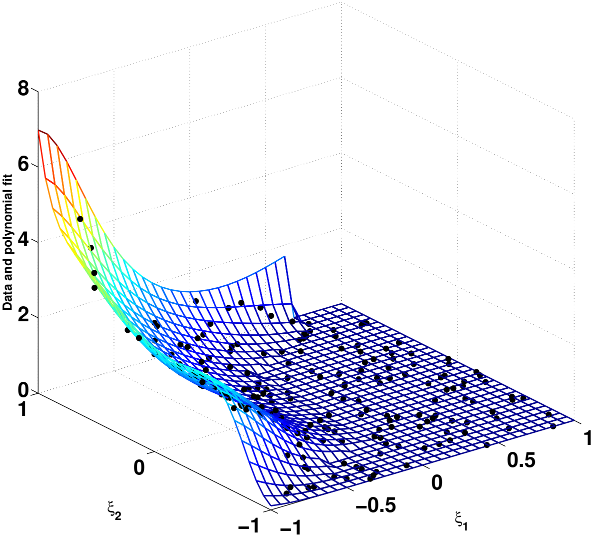
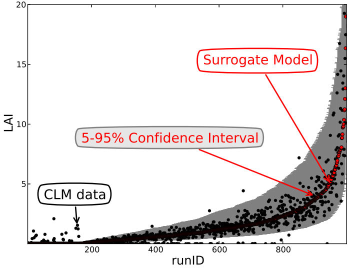


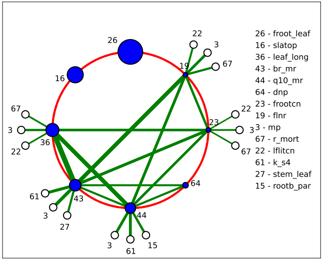
Illustration of piecewise PC construction



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





For more details, see [Cosmin Safta, Thu 2:35 pm, LMWG joint UQ]

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 *iterative* Bayesian CS algorithm that reduces dimensionality and increases order on-the-fly.
- Dependent inputs
  - Rosenblatt transformation
- Nonlinear behavior
  - Data clustering and classification-driven piecewise PC
- Apply to CLM: [Cosmin Safta, Thu 2:35 pm, LMWG joint UQ]

Literature

- M. Rosenblatt, "Remarks on a multivariate transformation", *Ann. Math. Statist.*, 23:3, pp. 470-472, 1952.
- 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.
- 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", submitted to *Int J for Uncertainty Quantification*, 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

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, $k_1 + k_2 + \ldots k_d \le p$.
  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]

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]

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]

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]

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; S. *et al*, 2011]
- Data domain decomposition,
  - Mixture PC expansions [S. *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,

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

- 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].