Prediction Uncertainty in Analysis of a Computer Experiment: Fast Bayesian Inference

Bela Nagy, Jason Loeppky, Will Welch

University of British Columbia

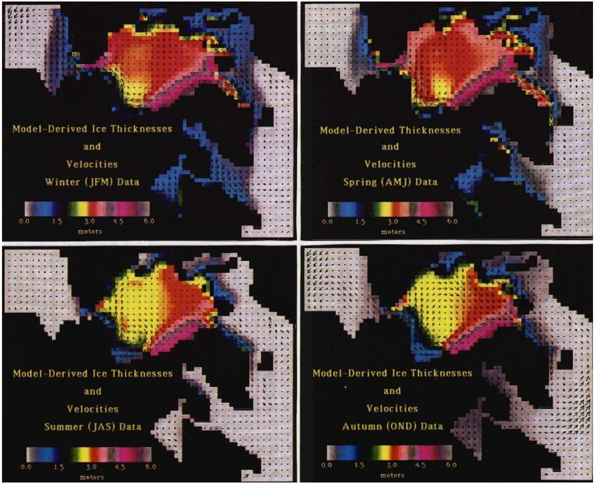
IMAGe/SAMSI Workshop
Application of Statistics to Numerical Models:
New Methods and Case Studies
May 21–23, 2007

Outline of Topics

Arctic Sea Ice Example

Chapman, Welch, Bowman, Sacks and Walsh, "Arctic Sea Ice Variability: Model Sensitivities and a Multidecadal Simulation", *Journal of Geophysical Research*, 99C (1994), 919–935

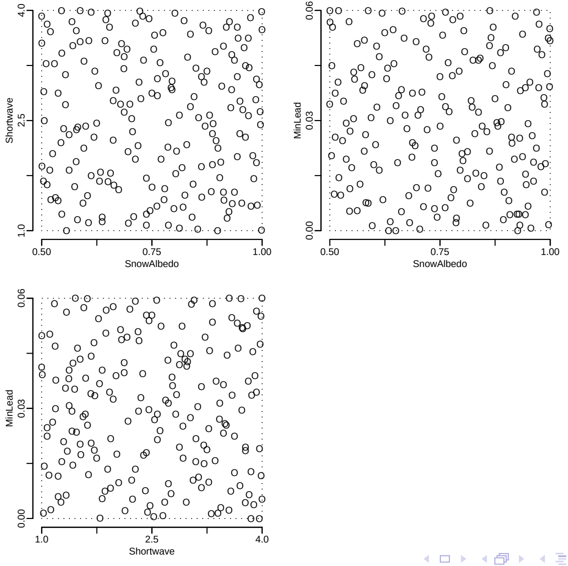
Arctic Sea Ice: Code Output



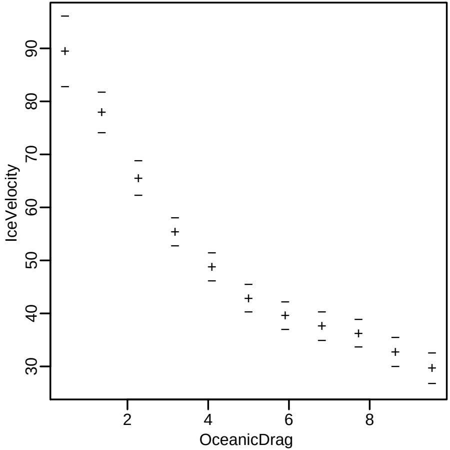
Arctic Sea Ice: Code Input and Output Variables

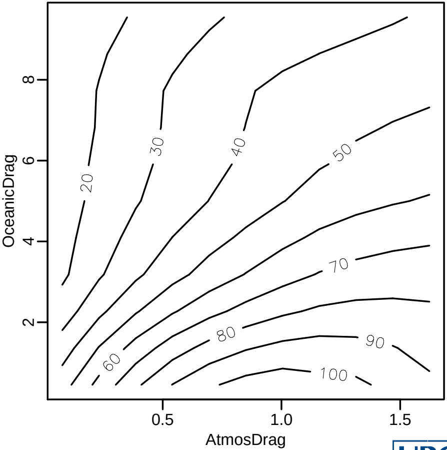
- 13 input variables:
  SnowAlbedo, Shortwave, MinLead, OceanicHeat, Snowfall, Longwave, AtmosDrag, SensHeat, LatentHeat, LogIceStr, OceanicDrag, OpenAlbedo, IceAlbedo
- 4 output variables:
  IceMass, IceArea, IceVelocity, RangeOfArea
- 157 good runs of the code (out of 191 attempts)

Arctic Sea Ice: Experimental Design (First 3 Input Variables)



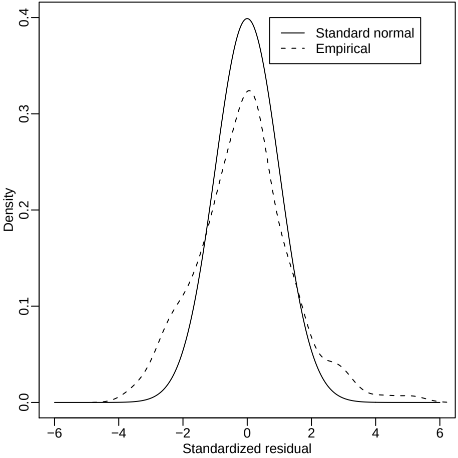
Arctic Sea Ice: Sensitivity Analysis for Ice Velocity





Arctic Sea Ice: The Inference Problem Illustrated

Standardized Residuals $= (y(\mathbf{x}) - \hat{y}(\mathbf{x}))/se(\hat{y}(\mathbf{x}))$ from 10-fold cross-validation (RangeOfArea):



Computer Codes in General

Computer code $y(\mathbf{x})$

- $\mathbf{x} = (x_1, \ldots, x_d)$ is a vector of d input variables
- y is a (scalar) output variable

Statistical Approximating Model

Approximation / prediction / emulation of $y(\mathbf{x})$ is the "engine" of analysis of computer experiments:

- To replace the computer model in future with a fast surrogate
- Sensitivity analysis
- Visualization
- Optimization
- Code calibration and assessment

Random Function Model

The unknown function $y(\mathbf{x})$ of vector-valued $\mathbf{x}$ is treated as if it were a *realization of a random function* (RF) or *Gaussian Stochastic Process* (GaSP), $Z(\mathbf{x})$:

- $Z(\mathbf{x})$ has a Gaussian (normal) distribution at any $\mathbf{x}$.

Random Function Model

The unknown function $y(\mathbf{x})$ of vector-valued $\mathbf{x}$ is treated as if it were a *realization of a random function* (RF) or *Gaussian Stochastic Process* (GaSP), $Z(\mathbf{x})$:

- $Z(\mathbf{x})$ has a Gaussian (normal) distribution at any $\mathbf{x}$.
- $Z(\mathbf{x})$ has mean β (can be generalized to regression terms).

Random Function Model

The unknown function $y(\mathbf{x})$ of vector-valued $\mathbf{x}$ is treated as if it were a *realization of a random function* (RF) or *Gaussian Stochastic Process* (GaSP), $Z(\mathbf{x})$:

- $Z(\mathbf{x})$ has a Gaussian (normal) distribution at any $\mathbf{x}$.
- $Z(\mathbf{x})$ has mean β (can be generalized to regression terms).
- $\mathrm{Var}(Z(\mathbf{x})) = \sigma_Z^2$, i.e., is constant.

Random Function Model

The unknown function $y(\mathbf{x})$ of vector-valued $\mathbf{x}$ is treated as if it were a *realization of a random function* (RF) or *Gaussian Stochastic Process* (GaSP), $Z(\mathbf{x})$:

- $Z(\mathbf{x})$ has a Gaussian (normal) distribution at any $\mathbf{x}$.
- $Z(\mathbf{x})$ has mean β (can be generalized to regression terms).
- $\mathrm{Var}(Z(\mathbf{x})) = \sigma_Z^2$, i.e., is constant.
- At two input vectors, $\mathbf{x}$ and $\mathbf{x}'$, the corresponding Z's are correlated,

$$\mathrm{Corr}(Z(\mathbf{x}), Z(\mathbf{x}')) \equiv R(\mathbf{x}, \mathbf{x}'),$$

  where R is a correlation function known up to various parameters.

  $\mathrm{Corr}(Z(\mathbf{x}), Z(\mathbf{x}'))$ is key to the random function model.

Power-Exponential Correlation

In one dimension: $R(x, x') = e^{-\theta|x-x'|^p}$.

- $\theta \geq 0$ controls the activity of the function.
  - Larger θ gives smaller correlation, i.e., $Z(\mathbf{x})$ and $Z(\mathbf{x}')$ are less related and the function is more complex.
- $p \in [1, 2]$ affects the smoothness of the random function.
  - $p = 2$ (Gaussian or squared exponential correlation function) gives very smooth realizations (good for approximating functions with many derivatives).
  - $p = 1$ gives much rougher realizations (good for continuous but non-differentiable functions).

Much used in practice.

Statistical Approximator/Predictor/Emulator (Kriging)

n runs, input vectors $\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(n)}$, and output values $\mathbf{y}$.
Approximator of $y(\mathbf{x})$ at any $\mathbf{x}$:

$$\hat{y}(\mathbf{x}) = \hat{\beta} + \mathbf{r}(\mathbf{x})^T(\mathbf{x})\mathbf{R}^{-1}(\mathbf{y} - \mathbf{1}\hat{\beta}),$$

where

- $\hat{\beta} = \mathbf{1}^T\mathbf{R}^{-1}\mathbf{y}/\mathbf{1}^T\mathbf{R}^{-1}\mathbf{1}$
- $\mathbf{1}$ is an $n \times 1$ vector of 1's
- $\mathbf{R}$ is an $n \times n$ correlation matrix, with element i, j given by $R(\mathbf{x}^{(i)}, \mathbf{x}^{(j)})$
- $\mathbf{r}(\mathbf{x})$ is an $n \times 1$ vector of correlations with element i given by $R(\mathbf{x}^{(i)}, \mathbf{x})$

Standard Error of the Approximator

Standard error of $\hat{y}(\mathbf{x})$:

$$\mathrm{se}(\hat{y}(\mathbf{x})) = \sigma^2 \left[1 - \mathbf{r}(\mathbf{x})'\mathbf{R}^{-1}\mathbf{r}(\mathbf{x}) + \frac{\left(1 - \mathbf{1}'\mathbf{R}^{-1}\mathbf{r}(\mathbf{x})\right)^2}{\mathbf{1}'\mathbf{R}^{-1}\mathbf{1}} \right],$$

where $\mathbf{r}(\mathbf{x})$, $\mathbf{R}$, and $\mathbf{1}$ were defined on the previous slide.
We have to estimate the correlation parameters

$$\theta_1, \ldots, \theta_d,\ p_1, \ldots, p_d,$$

e.g., by maximum likelihood.
THE PROBLEM: If we just PLUG-IN the estimates of the correlation parameters, the above $\mathrm{se}(\hat{y}(\mathbf{x}))$ ignores this source of uncertainty and is too small.

Bayesian Formulation

In principle, Bayes can deal with all sources of uncertainty.
Let $\boldsymbol{\theta}$ denote all the correlation parameters.
Posterior for $\boldsymbol{\theta}$ given $\mathbf{y}$ from Bayes rule:

$$p(\boldsymbol{\theta} \mid \mathbf{y}) \propto \pi(\boldsymbol{\theta}) L(\mathbf{y} \mid \boldsymbol{\theta})$$

- $\pi(\boldsymbol{\theta})$ is the prior for $\boldsymbol{\theta}$
- $L(\mathbf{y} \mid \boldsymbol{\theta})$ is the multivariate normal likelihood arising from the RF model.

The predictive distribution for $y(\mathbf{x})$ is

$$p(y(\mathbf{x} \mid \mathbf{y})) = \int p(y(\mathbf{x}) \mid \mathbf{y}, \boldsymbol{\theta}) p(\boldsymbol{\theta} \mid \mathbf{y}) \, d\boldsymbol{\theta}.$$

Objections

I want to be a Bayesian BUT...

1. I'll have to choose a prior, i.e., $\pi(\boldsymbol{\theta})$.

Objections

I want to be a Bayesian BUT...

1. I'll have to choose a prior, i.e., $\pi(\boldsymbol{\theta})$.
2. I'll have to run MCMC.

Relax. You won't have to!

FBI: Main Results

1. Results:
    - Even the (computationally) cheapest Bayesian implementation beats the plug-in method in terms of validity of uncertainty measures.
2. Methods:
    - Plug-in (Kriging)
    - Fast Bayesian Implementation (FBI)
        - Fast to implement
        - Fast to execute

FBI Algorithm

Only implemented for the Gaussian correlation function, $R(x, x') = e^{-\theta|x-x'|^2}$.

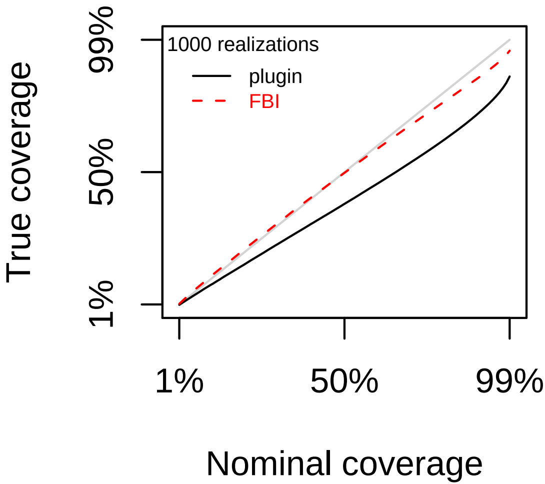
1. Use the $\theta_j^* = \log \theta_j$ transformation.
2. Maximize the log likelihood to get the MLE of $\boldsymbol{\theta}^*$ and the HESSIAN at the MLE.
3. Sample $\boldsymbol{\theta}^*$ from $N(\mathrm{MLE}, -\mathrm{HESSIAN}^{-1})$ to get:
    - $\hat{y}_{\mathrm{FBI}} = \mathrm{E}_{\boldsymbol{\theta}^*}(\hat{y}_{\mathrm{Plug-in}})$
    - $\mathrm{Var}(\hat{y}_{\mathrm{FBI}}) = \mathrm{Var}_{\boldsymbol{\theta}^*}(\hat{y}_{\mathrm{Plug-in}}) + \mathrm{E}_{\boldsymbol{\theta}^*}(\mathrm{Var}(\hat{y}_{\mathrm{Plug-in}}))$
4. Sampling is straightforward Monte Carlo or Quasi Monte Carlo (100 samples)! Not MCMC.

Simulations

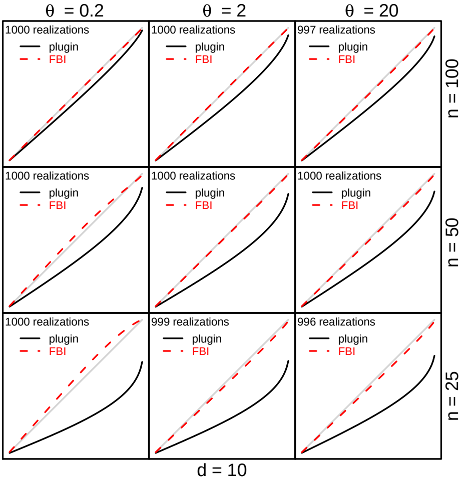
- Generate $y(\mathbf{x})$ at N values of $\mathbf{x} \in [0,1]^d$ as a realization of a Gaussian Random Function, $Z(\mathbf{x})$, with Gaussian correlation function.
- Choose n of the $\mathbf{x}$ points as an experimental design, and use the corresponding realized $y(\mathbf{x}) = Z(\mathbf{x})$ values to fit the RF model.
- Predict at the remaining $N - n$ test points.
- Compute actual coverage probabilities of the predictions (should equal the nominal coverage probability).

As $y(\mathbf{x})$ is sampled from the assumed statistical model, uncertainty estimates should be EXACT, leading to correct coverage probabilities.

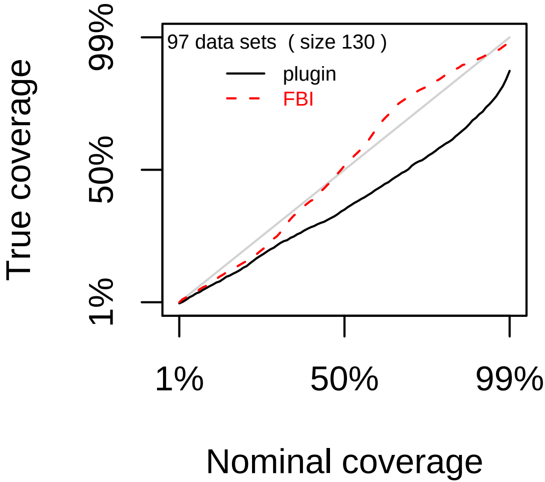
Simulation Results for $d = 1$, $n = 10$, and $\theta = 2$



Simulation Results for 10-dimensional $\mathbf{x}$



Arctic Sea Ice: $d = 13$ and $n = 130$



Summary

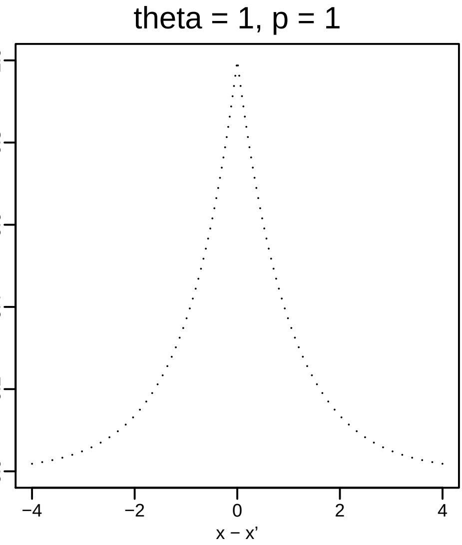
1. Methods:
   - Plug-in
   - Fast Bayesian Implementation (FBI) (extends the thesis of Karuri, 2005)
     - Fast to implement
     - Fast to execute
2. Results:
   - Even the (computationally) cheapest Bayesian implementation beats the plug-in method (consistent with Currin et al. 1991)
   - FBI has many approximations but IT WORKS!

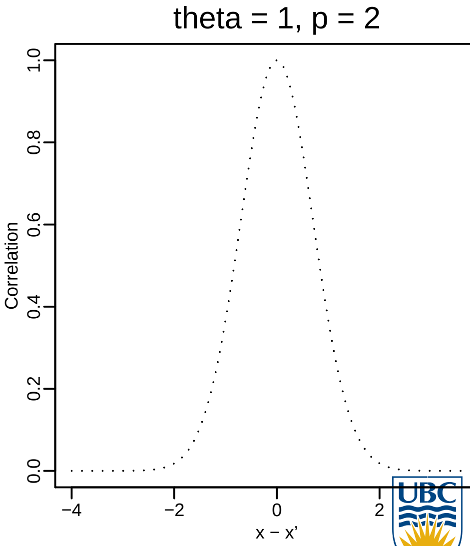
The End—Thank You

References on previous page:

- Currin, C., Mitchell, T., Morris, M., and Ylvisaker, D. (1991). "Bayesian Prediction of Deterministic Functions With Applications to the Design and Analysis of Computer Experiments." *Journal of the American Statistical Association*, 86, 953-963.
- Karuri, S. (2005). "Integration in Computer Experiments and Bayesian Analysis." Ph.D Thesis, University of Waterloo.

Power-Exponential Correlation





Correlation in Several Dimensions

With d-dimensional $\mathbf{x}$, i.e.,

$$\mathbf{x} = (x_1, \ldots, x_d),$$

we simply use a product correlation rule, i.e.,

$$R(\mathbf{x}, \mathbf{x}') = \prod_j R_j(x_j, x_j') = \prod_j e^{-\theta_j |x_j - x_j'|^{p_j}}.$$