Calibrating Environmental Engineering Models

David Ruppert

Cornell University

November 4, 2006

What this talk is about

Project Team

- Christine Shoemaker, co-PI, Professor of Civil and Environmental Engineering
  - PhD in applied math
  - works in applied optimization
- David Ruppert, co-PI
- Nikolai Bliznyouk, PhD student in Operations Research
- Christine's students and post-docs
  - Rommel Regis
  - Stefan Wild
  - Pradeep Mugunthan

Work of Chris Shoemaker and her students

Christine Shoemaker's research:

- global optimization
- optimization of computationally expensive functions
- methods for calibration and uncertainty analysis
- example: remediation of a US-DOD site
  - contaminated with chlorinated ethenes in the soil and groundwater

Calibrating Environment Models

- Typical problem in environmental engineering:
  - remediation of comtaminated groundwater
- Modeled as a system of partial differential equations describing
  - movement of pollutants
  - chemical reactions
- Coefficients are unknown
- Calibration means estimation of parameters in model
  - this is an inverse problem
  - also a nonlinear regression problem

Why is Calibration Difficult?

- Likelihood may be multimodal
- Non-Gaussian data
- Spatial and temporal correlations
- Model is computationally expensive
  - May take minutes or even hours to evaluate the model for one set of parameter values

Deterministic component of model

- ith observation is

$$Y_i = (Y_{i,1}, \ldots, Y_{i,d})^T$$

- in absence of noise:

$$Y_{i,j} = f_j(X_i, \boldsymbol{\beta})$$

  - $f_j(\cdot)$ comes from scientific theory
  - X_i is a covariate vector
  - $\boldsymbol{\beta}$ contains the parameters of interest

- noise is modeled empirically

Components of the noise model

We modeled the noise via:

- data transformation
- spatial-temporal correlation model

Purpose of data transformation

We used transformations to:

- normalize the response distribution
- stabilize the variance

Transform-both-sides model

- The transform-both-sides model is

$$h\left\{ Y_{i,j}, \lambda_j \right\} = h\left\{ f_j(X_i, \boldsymbol{\beta}), \lambda_j \right\} + \epsilon_{i,j},$$

Transform-both-sides model

- The transform-both-sides model is

$$h\{Y_{i,j}, \lambda_j\} = h\{f_j(X_i, \boldsymbol{\beta}), \lambda_j\} + \epsilon_{i,j},$$

- equivalently

$$Y_{i,j} = h^{-1}\left[h\{f_j(X_i, \boldsymbol{\beta}), \lambda_j\} + \epsilon_{i,j}, \lambda_j\right]$$

Transform-both-sides model

- The transform-both-sides model is

$$h\left\{Y_{i,j}, \lambda_j\right\} = h\left\{f_j(X_i, \boldsymbol{\beta}), \lambda_j\right\} + \epsilon_{i,j},$$

- equivalently

$$Y_{i,j} = h^{-1}\left[h\left\{f_j(X_i, \boldsymbol{\beta}), \lambda_j\right\} + \epsilon_{i,j}, \lambda_j\right]$$

- transforms both sides of the equation giving deterministic model

Transform-both-sides model

- The transform-both-sides model is

$$h\left\{Y_{i,j}, \lambda_j\right\} = h\left\{f_j(X_i, \boldsymbol{\beta}), \lambda_j\right\} + \epsilon_{i,j},$$

- equivalently

$$Y_{i,j} = h^{-1}\left[h\left\{f_j(X_i, \boldsymbol{\beta}), \lambda_j\right\} + \epsilon_{i,j}, \lambda_j\right]$$

- transforms both sides of the equation giving deterministic model
- preserves the theoretical model

Transform-both-sides model

- The transform-both-sides model is

$$h\{Y_{i,j}, \lambda_j\} = h\{f_j(X_i, \boldsymbol{\beta}), \lambda_j\} + \epsilon_{i,j},$$

- equivalently

$$Y_{i,j} = h^{-1}\left[h\{f_j(X_i, \boldsymbol{\beta}), \lambda_j\} + \epsilon_{i,j}, \lambda_j\right]$$

- transforms both sides of the equation giving deterministic model
- preserves the theoretical model
- $\{h(\cdot, \lambda) : \lambda \in \Lambda\}$ is some transformation family

Transform-both-sides examples

- the identity transformation gives the usual nonlinear regression model

Transform-both-sides examples

- the identity transformation gives the usual nonlinear regression model
  - additive Gaussian errors

Transform-both-sides examples

- the identity transformation gives the usual nonlinear regression model
  - additive Gaussian errors
- if we use the log transformation then

$$Y_{i,j} = \exp\left[\log\{f_j(X_i, \boldsymbol{\beta})\} + \epsilon_{i,j}\right] = f_j(X_i, \boldsymbol{\beta}) \exp(\epsilon_{i,j})$$

Transform-both-sides examples

- the identity transformation gives the usual nonlinear regression model
  - additive Gaussian errors
- if we use the log transformation then

$$Y_{i,j} = \exp\left[\log\{f_j(X_i, \boldsymbol{\beta})\} + \epsilon_{i,j}\right] = f_j(X_i, \boldsymbol{\beta}) \exp(\epsilon_{i,j})$$

  - multiplicative, lognormal errors

Transform-both-sides examples

- the identity transformation gives the usual nonlinear regression model
  - additive Gaussian errors
- if we use the log transformation then

$$Y_{i,j} = \exp\left[\log\{f_j(X_i, \boldsymbol{\beta})\} + \epsilon_{i,j}\right] = f_j(X_i, \boldsymbol{\beta}) \exp(\epsilon_{i,j})$$

  - multiplicative, lognormal errors
- if we use the square root transformation

$$Y_{i,j} = \left[\sqrt{f_j(X_i, \boldsymbol{\beta})} + \epsilon_{i,j}\right]^2$$

Transform-both-sides examples

- the identity transformation gives the usual nonlinear regression model
  - additive Gaussian errors
- if we use the log transformation then

$$Y_{i,j} = \exp\left[\log\{f_j(X_i,\boldsymbol{\beta})\} + \epsilon_{i,j}\right] = f_j(X_i,\boldsymbol{\beta})\exp(\epsilon_{i,j})$$

  - multiplicative, lognormal errors
- if we use the square root transformation

$$Y_{i,j} = \left[\sqrt{f_j(X_i,\boldsymbol{\beta})} + \epsilon_{i,j}\right]^2$$

  - notice a problem?

The Box-Cox family

- the most common transformation family is due to Box and Cox (1964):

$$
\begin{aligned}
h(y, \lambda) &= \frac{y^\lambda - 1}{\lambda} \text{ if } \lambda \neq 0 \\
&= \log(y) \text{ if } \lambda = 0
\end{aligned}
$$

The Box-Cox family

- the most common transformation family is due to Box and Cox (1964):

$$
\begin{aligned}
h(y, \lambda) &= \frac{y^\lambda - 1}{\lambda} \text{ if } \lambda \neq 0 \\
&= \log(y) \text{ if } \lambda = 0
\end{aligned}
$$

- technical problem:
  - does not map $(0, \infty)$ onto $(-\infty, \infty)$, except for $\lambda = 0$
  - so transformed response has a truncated normal distribution
  - this makes Bayesian inference more complex

COIL transformation family

- COnvex combination of Identity and Log (COIL) family:

$$h_C(y, \lambda) = \lambda y + (1 - \lambda) \log(y), \quad 0 \leq \lambda \leq 1.$$

COIL transformation family

- COnvex combination of Identity and Log (COIL) family:

$$h_C(y, \lambda) = \lambda y + (1 - \lambda)\log(y), \quad 0 \leq \lambda \leq 1.$$

- We restrict λ to $[0, 1)$, since $h_C(\cdot, 1)$ does not map $(0, \infty)$ to $(-\infty, \infty)$

COIL transformation family

- COnvex combination of Identity and Log (COIL) family:

$$h_C(y, \lambda) = \lambda y + (1 - \lambda) \log(y), \quad 0 \leq \lambda \leq 1.$$

- We restrict λ to $[0, 1)$, since $h_C(\cdot, 1)$ does not map $(0, \infty)$ to $(-\infty, \infty)$
- COIL can approximate Box-Cox:

COIL transformation family

- COnvex combination of Identity and Log (COIL) family:

$$h_C(y, \lambda) = \lambda y + (1 - \lambda)\log(y), \quad 0 \le \lambda \le 1.$$

- We restrict λ to $[0, 1)$, since $h_C(\cdot, 1)$ does not map $(0, \infty)$ to $(-\infty, \infty)$
- COIL can approximate Box-Cox:
  - For each $\lambda \in [0, 1)$ there are constants $\lambda' \in [0, 1)$ and $a, b \in \mathbb{R}$ such that

$$h_{BC}(y, \lambda) \approx a + b\, h_C(y, \lambda')$$

  for a wide range of y values (verified empirically)

COIL transformation family

- COnvex combination of Identity and Log (COIL) family:

$$h_C(y, \lambda) = \lambda y + (1-\lambda)\log(y), \quad 0 \leq \lambda \leq 1.$$

- We restrict λ to $[0, 1)$, since $h_C(\cdot, 1)$ does not map $(0, \infty)$ to $(-\infty, \infty)$
- COIL can approximate Box-Cox:
  - For each $\lambda \in [0, 1)$ there are constants $\lambda' \in [0, 1)$ and $a, b \in \mathbb{R}$ such that

$$h_{BC}(y, \lambda) \approx a + b\, h_C(y, \lambda')$$

    for a wide range of y values (verified empirically)
- The inverse $h_C^{-1}(\cdot, \lambda)$ does not have a closed form

COIL transformation family

- COnvex combination of Identity and Log (COIL) family:

$$h_C(y, \lambda) = \lambda y + (1 - \lambda) \log(y), \quad 0 \leq \lambda \leq 1.$$

- We restrict λ to $[0, 1)$, since $h_C(\cdot, 1)$ does not map $(0, \infty)$ to $(-\infty, \infty)$
- COIL can approximate Box-Cox:
  - For each $\lambda \in [0, 1)$ there are constants $\lambda' \in [0, 1)$ and $a, b \in \mathbb{R}$ such that

$$h_{BC}(y, \lambda) \approx a + b\, h_C(y, \lambda')$$

    for a wide range of y values (verified empirically)
- The inverse $h_C^{-1}(\cdot, \lambda)$ does not have a closed form
  - evaluate by interpolation (fast)

Multivariate transformations

- Define

$$\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_d)^T$$

- and

$$h(y, \boldsymbol{\lambda}) = \{h(y_1, \lambda_1), \ldots, h(y_d, \lambda_d)\}^T$$

Separable correlation model

- Define the noise vectors:
  - $\epsilon_i = (\epsilon_{i,1}, \ldots, \epsilon_{i,d})^T = h\{Y_i, \boldsymbol{\lambda}\} - h\{f(X_i, \boldsymbol{\beta}), \boldsymbol{\lambda}\}$
  - $\epsilon_{\bullet,j} = (\epsilon_{1,j}, \ldots, \epsilon_{n,j})^T$
  - $\boldsymbol{\epsilon} = (\epsilon_1^T, \ldots, \epsilon_n^T)^T$
- $\text{cov}(\epsilon_{i,j}, \epsilon_{i',j'}) = \boldsymbol{C}_{j,j'} \cdot \rho_{ST}(X_i, X_{i'}; \boldsymbol{\gamma})$
  - $\boldsymbol{C}$ is a $d \times d$ covariance matrix for ϵ_i
  - $\rho_{ST}(X_i, X_{i'}; \boldsymbol{\gamma})$ is a space-time correlation function parameterized by $\boldsymbol{\gamma}$
- $\text{Var}\{\boldsymbol{\epsilon}\} = \boldsymbol{\Sigma}(\boldsymbol{\theta}) = \boldsymbol{S}(\boldsymbol{\gamma}) \otimes \boldsymbol{C}$
  - $\boldsymbol{\theta} = (\boldsymbol{\gamma}, \boldsymbol{C})$
  - $\boldsymbol{S}_{i,i'}(\boldsymbol{\gamma}) = \rho_{ST}(X_i, X_{i'}; \boldsymbol{\gamma})$

TBS Likelihood

- Our statistical model is $h\{\boldsymbol{Y},\boldsymbol{\lambda}\} \sim MVN\left[h\{\boldsymbol{f}(\boldsymbol{\beta}),\boldsymbol{\lambda}\},\boldsymbol{\Sigma}(\boldsymbol{\theta})\right]$
- Likelihood is

$$[\boldsymbol{Y}|\boldsymbol{\beta},\boldsymbol{\lambda},\boldsymbol{\theta}] =$$

$$\frac{\exp\left[-0.5\,\|h(\boldsymbol{Y},\boldsymbol{\lambda}) - h\{\boldsymbol{f}(\boldsymbol{\beta}),\boldsymbol{\lambda}\}\|^2_{\boldsymbol{\Sigma}(\boldsymbol{\theta})^{-1}}\right]}{(2\pi)^{nd/2}|\boldsymbol{\Sigma}(\boldsymbol{\theta})|^{1/2}} \cdot |J_h(\boldsymbol{Y},\boldsymbol{\lambda})|$$

  - $|J_h(\boldsymbol{Y},\boldsymbol{\lambda})|$ is the Jacobian
  - $\boldsymbol{\Sigma}(\boldsymbol{\theta})$ is the covariance matrix

Overview of Methodology

- Goal:

Overview of Methodology

- Goal:
  - Approximate the posterior density accurately with as few expensive likelihood evaluations as possible

Overview of Methodology

- Goal:
  - Approximate the posterior density accurately with as few expensive likelihood evaluations as possible
- There are four steps:

Overview of Methodology

- Goal:
  - Approximate the posterior density accurately with as few expensive likelihood evaluations as possible
- There are four steps:
  1. Locate the region(s) of high posterior density

Overview of Methodology

- Goal:
  - Approximate the posterior density accurately with as few expensive likelihood evaluations as possible
- There are four steps:
  1. Locate the region(s) of high posterior density
  2. Find an "experimental design" that covers the region of high posterior density

Overview of Methodology

- Goal:
  - Approximate the posterior density accurately with as few expensive likelihood evaluations as possible
- There are four steps:
  1. Locate the region(s) of high posterior density
  2. Find an "experimental design" that covers the region of high posterior density
     - the likelihood is evaluated on this design

Overview of Methodology

- Goal:
  - Approximate the posterior density accurately with as few expensive likelihood evaluations as possible
- There are four steps:
  1. Locate the region(s) of high posterior density
  2. Find an "experimental design" that covers the region of high posterior density
     - the likelihood is evaluated on this design
  3. Use function evaluations from Steps 1 and 2 to approximate the posterior

Overview of Methodology

- Goal:
  - Approximate the posterior density accurately with as few expensive likelihood evaluations as possible
- There are four steps:
  1. Locate the region(s) of high posterior density
  2. Find an "experimental design" that covers the region of high posterior density
     - the likelihood is evaluated on this design
  3. Use function evaluations from Steps 1 and 2 to approximate the posterior
  4. MCMC and standard Bayesian analysis using the approximate posterior density

Removing nuisance parameters

- The posterior density is

$$[\boldsymbol{\beta}, \boldsymbol{\lambda}, \boldsymbol{\theta} | \, \boldsymbol{Y}] = \frac{[\boldsymbol{\beta}, \boldsymbol{\lambda}, \boldsymbol{\theta}, \, \boldsymbol{Y}]}{\int [\boldsymbol{\beta}, \boldsymbol{\lambda}, \boldsymbol{\theta}, \, \boldsymbol{Y}] \, d\boldsymbol{\beta} \, d\boldsymbol{\lambda} \, d\boldsymbol{\theta}},$$

  - where $[\boldsymbol{\beta}, \boldsymbol{\lambda}, \boldsymbol{\theta}, \, \boldsymbol{Y}] = [\, \boldsymbol{Y} | \boldsymbol{\beta}, \boldsymbol{\lambda}, \boldsymbol{\theta}] \cdot [\boldsymbol{\beta}, \boldsymbol{\lambda}, \boldsymbol{\theta}]$

Removing nuisance parameters

- The posterior density is

$$[\boldsymbol{\beta}, \boldsymbol{\lambda}, \boldsymbol{\theta} | \, \boldsymbol{Y}] = \frac{[\boldsymbol{\beta}, \boldsymbol{\lambda}, \boldsymbol{\theta}, \, \boldsymbol{Y}]}{\int [\boldsymbol{\beta}, \boldsymbol{\lambda}, \boldsymbol{\theta}, \, \boldsymbol{Y}] \, d\boldsymbol{\beta} \, d\boldsymbol{\lambda} \, d\boldsymbol{\theta}},$$

  - where $[\boldsymbol{\beta}, \boldsymbol{\lambda}, \boldsymbol{\theta}, \, \boldsymbol{Y}] = [\, \boldsymbol{Y} | \boldsymbol{\beta}, \boldsymbol{\lambda}, \boldsymbol{\theta}] \cdot [\boldsymbol{\beta}, \boldsymbol{\lambda}, \boldsymbol{\theta}]$
- Interest focuses on

$$[\boldsymbol{\beta} | \, \boldsymbol{Y}] = \int [\boldsymbol{\beta}, \boldsymbol{\lambda}, \boldsymbol{\theta} | \, \boldsymbol{Y}] \, d\boldsymbol{\lambda} \, d\boldsymbol{\theta}$$

Removing nuisance parameters - four methods

- Exact:

$$[\boldsymbol{\beta}|\, \boldsymbol{Y}] = \int [\boldsymbol{\beta}, \boldsymbol{\lambda}, \boldsymbol{\theta}|\, \boldsymbol{Y}]\, d\boldsymbol{\lambda}\, d\boldsymbol{\theta}$$

- Profile posterior:

$$\pi_{\max}(\boldsymbol{\beta},\, \boldsymbol{Y}) = \sup_{\boldsymbol{\zeta}} [\boldsymbol{\beta}, \boldsymbol{\zeta},\, \boldsymbol{Y}] = [\boldsymbol{\beta}, \widehat{\boldsymbol{\zeta}}(\boldsymbol{\beta}),\, \boldsymbol{Y}]$$

  - $\widehat{\boldsymbol{\zeta}}(\boldsymbol{\beta})$ maximizes $[\boldsymbol{\beta}, \boldsymbol{\zeta},\, \boldsymbol{Y}]$ with respect to $\boldsymbol{\zeta}$
- Laplace approximation:
  - multiplies the profile posterior by a correction factor
- Pseudo-posterior:

$$[\boldsymbol{\beta}, \widehat{\boldsymbol{\zeta}}(\widehat{\boldsymbol{\beta}}),\, \boldsymbol{Y}]$$

  - $\{\widehat{\boldsymbol{\beta}}, \widehat{\boldsymbol{\zeta}}(\widehat{\boldsymbol{\beta}})\}$ is the MAP = joint mode of posterior

Finding posterior mode using Condor

- When locating the posterior mode we want:

Finding posterior mode using Condor

- When locating the posterior mode we want:
  1. As few expensive function evaluations as possible

Finding posterior mode using Condor

- When locating the posterior mode we want:
  1. As few expensive function evaluations as possible
  2. A small percentage of “wasted evaluations”

Finding posterior mode using Condor

- When locating the posterior mode we want:
  1. As few expensive function evaluations as possible
  2. A small percentage of "wasted evaluations"
     a) few evaluation locations in region of very low posterior probability

Finding posterior mode using Condor

- When locating the posterior mode we want:
  1. As few expensive function evaluations as possible
  2. A small percentage of "wasted evaluations"
     a) few evaluation locations in region of very low posterior probability
     b) few evaluation locations that are very close together

Finding posterior mode using Condor

- When locating the posterior mode we want:
  1. As few expensive function evaluations as possible
  2. A small percentage of "wasted evaluations"
     a) few evaluation locations in region of very low posterior probability
     b) few evaluation locations that are very close together
  3. Getting very close to the mode is not a goal

Finding posterior mode using Condor

- When locating the posterior mode we want:
  1. As few expensive function evaluations as possible
  2. A small percentage of "wasted evaluations"
     a) few evaluation locations in region of very low posterior probability
     b) few evaluation locations that are very close together
  3. Getting very close to the mode is not a goal
- All good optimization techniques achieve 1

Finding posterior mode using Condor

- When locating the posterior mode we want:
  1. As few expensive function evaluations as possible
  2. A small percentage of "wasted evaluations"
     a) few evaluation locations in region of very low posterior probability
     b) few evaluation locations that are very close together
  3. Getting very close to the mode is not a goal
- All good optimization techniques achieve 1
- Optimization methods based on numerical derivatives violate 2 b)

Finding posterior mode using Condor

- When locating the posterior mode we want:
    1. As few expensive function evaluations as possible
    2. A small percentage of "wasted evaluations"
        a) few evaluation locations in region of very low posterior probability
        b) few evaluation locations that are very close together
    3. Getting very close to the mode is not a goal
- All good optimization techniques achieve 1
- Optimization methods based on numerical derivatives violate 2 b)
    - MATLAB's `fmincon` exhibited this problem

Finding posterior mode using Condor

- When locating the posterior mode we want:
  1. As few expensive function evaluations as possible
  2. A small percentage of "wasted evaluations"
     a) few evaluation locations in region of very low posterior probability
     b) few evaluation locations that are very close together
  3. Getting very close to the mode is not a goal
- All good optimization techniques achieve 1
- Optimization methods based on numerical derivatives violate 2 b)
  - MATLAB's `fmincon` exhibited this problem
- `CONDOR` uses sequential quadratic programming

Finding posterior mode using Condor

- When locating the posterior mode we want:
  1. As few expensive function evaluations as possible
  2. A small percentage of "wasted evaluations"
     a) few evaluation locations in region of very low posterior probability
     b) few evaluation locations that are very close together
  3. Getting very close to the mode is not a goal
- All good optimization techniques achieve 1
- Optimization methods based on numerical derivatives violate 2 b)
  - MATLAB's `fmincon` exhibited this problem
- `CONDOR` uses sequential quadratic programming
  - worked well in our empirical tests

Further function evaluations needed

- Goal:
  - approximate posterior on $C_R(\alpha) = \{\boldsymbol{\beta} : [\boldsymbol{\beta}, \boldsymbol{Y}] > \kappa(\alpha)\}$
- Function evaluations in optimization stage insufficient to approximate posterior accurately

Constructing the experimental design

1. Normal approximation to posterior

Constructing the experimental design

1. Normal approximation to posterior
   - requires a small number of additional function evaluations

Constructing the experimental design

1. Normal approximation to posterior
   - requires a small number of additional function evaluations
2.

$$\widehat{C}_R(\alpha) = \left\{ \boldsymbol{\beta} : (\boldsymbol{\beta} - \widehat{\boldsymbol{\beta}})^T \left[\widehat{\boldsymbol{I}}^{\boldsymbol{\beta}\boldsymbol{\beta}} \right]^{-1} (\boldsymbol{\beta} - \widehat{\boldsymbol{\beta}}) \leq \chi^2_{p,1-\alpha} \right\}$$

Constructing the experimental design

1. Normal approximation to posterior
   - requires a small number of additional function evaluations
2. $$\widehat{C}_R(\alpha) = \left\{ \boldsymbol{\beta} : (\boldsymbol{\beta} - \widehat{\boldsymbol{\beta}})^T \left[\widehat{\boldsymbol{I}}^{\boldsymbol{\beta\beta}} \right]^{-1} (\boldsymbol{\beta} - \widehat{\boldsymbol{\beta}}) \leq \chi^2_{p,1-\alpha} \right\}$$
3. Space-filling design on $\widehat{C}_R(\alpha)$

Constructing the experimental design

1. Normal approximation to posterior
   - requires a small number of additional function evaluations
2. $$\widehat{C}_R(\alpha) = \left\{ \boldsymbol{\beta} : (\boldsymbol{\beta} - \widehat{\boldsymbol{\beta}})^T \left[\widehat{\boldsymbol{I}}^{\boldsymbol{\beta}\boldsymbol{\beta}} \right]^{-1} (\boldsymbol{\beta} - \widehat{\boldsymbol{\beta}}) \leq \chi^2_{p,1-\alpha} \right\}$$
3. Space-filling design on $\widehat{C}_R(\alpha)$
4. Remove points not in $\widehat{C}_R(\alpha')$ for $\alpha' < \alpha$

Constructing the experimental design

1. Normal approximation to posterior
   - requires a small number of additional function evaluations
2. $$\widehat{C}_R(\alpha) = \left\{ \boldsymbol{\beta} : (\boldsymbol{\beta} - \widehat{\boldsymbol{\beta}})^T \left[\widehat{\boldsymbol{I}}^{\boldsymbol{\beta}\boldsymbol{\beta}} \right]^{-1} (\boldsymbol{\beta} - \widehat{\boldsymbol{\beta}}) \le \chi^2_{p,1-\alpha} \right\}$$
3. Space-filling design on $\widehat{C}_R(\alpha)$
4. Remove points not in $\widehat{C}_R(\alpha')$ for $\alpha' < \alpha$
   - E.g., $\alpha = 0.1$ and $\alpha' = 0.01$

Radial basis functions

- $\pi(\cdot, \boldsymbol{Y})$ denotes one of the approximations to $[\boldsymbol{\beta}, \boldsymbol{Y}]$

Radial basis functions

- $\pi(\cdot, \boldsymbol{Y})$ denotes one of the approximations to $[\boldsymbol{\beta}, \boldsymbol{Y}]$
- $l(\cdot) = \log\{\pi(\cdot, \boldsymbol{Y})\}$ is interpolated at $\mathcal{B}_D = \{\boldsymbol{\beta}^{(1)}, \ldots, \boldsymbol{\beta}^{(N)}\}$ by

$$\widetilde{l}(\boldsymbol{\beta}) = \sum_{i=1}^{N} a_i \phi(\|\boldsymbol{\beta} - \boldsymbol{\beta}^{(i)}\|_2) + q(\boldsymbol{\beta})$$

where

- $a_1, \ldots, a_N \in \mathbb{R}$
- ϕ is a radial basis function
    - we used $\phi(r) = r^3$
- $q \in \Pi_m^p$ (the space of polynomials in $\mathbb{R}^p$ of degree $\leq m$
- $\boldsymbol{\beta} \in \mathbb{R}^p$

Autoregressive Metropolis-Hastings algorithm

- draw MCMC sample from $\widetilde{\pi}(\cdot, \boldsymbol{Y}) = \exp\{\widetilde{l}(\cdot)\}$

Autoregressive Metropolis-Hastings algorithm

- draw MCMC sample from $\widetilde{\pi}(\cdot, \boldsymbol{Y}) = \exp\{\widetilde{l}(\cdot)\}$
  - restrict sample to $\widehat{C}_R(\alpha')$

Autoregressive Metropolis-Hastings algorithm

- draw MCMC sample from $\widetilde{\pi}(\cdot, \boldsymbol{Y}) = \exp\{\widetilde{l}(\cdot)\}$
  - restrict sample to $\widehat{C}_R(\alpha')$
- Metropolis-Hastings candidate:
  $\boldsymbol{\beta}^c = \boldsymbol{\mu} + \boldsymbol{\rho}(\boldsymbol{\beta}^{(t)} - \boldsymbol{\mu}) + \boldsymbol{e}_t$

Autoregressive Metropolis-Hastings algorithm

- draw MCMC sample from $\widetilde{\pi}(\cdot, \boldsymbol{Y}) = \exp\{\widetilde{l}(\cdot)\}$
  - restrict sample to $\widehat{C}_R(\alpha')$
- Metropolis-Hastings candidate: $\boldsymbol{\beta}^c = \boldsymbol{\mu} + \boldsymbol{\rho}(\boldsymbol{\beta}^{(t)} - \boldsymbol{\mu}) + \boldsymbol{e}_t$
  - $\boldsymbol{\mu}$ = location parameter

Autoregressive Metropolis-Hastings algorithm

- draw MCMC sample from $\widetilde{\pi}(\cdot, \boldsymbol{Y}) = \exp\{\widetilde{l}(\cdot)\}$
  - restrict sample to $\widehat{C}_R(\alpha')$
- Metropolis-Hastings candidate:
  $\boldsymbol{\beta}^c = \boldsymbol{\mu} + \boldsymbol{\rho}(\boldsymbol{\beta}^{(t)} - \boldsymbol{\mu}) + \boldsymbol{e}_t$
  - $\boldsymbol{\mu}$ = location parameter
  - $\boldsymbol{\rho}$ = autoregressive parameter (matrix)

Autoregressive Metropolis-Hastings algorithm

- draw MCMC sample from $\widetilde{\pi}(\cdot, \boldsymbol{Y}) = \exp\{\widetilde{l}(\cdot)\}$
  - restrict sample to $\widehat{C}_R(\alpha')$
- Metropolis-Hastings candidate:
  $\boldsymbol{\beta}^c = \boldsymbol{\mu} + \boldsymbol{\rho}(\boldsymbol{\beta}^{(t)} - \boldsymbol{\mu}) + \boldsymbol{e}_t$
  - $\boldsymbol{\mu} =$ location parameter
  - $\boldsymbol{\rho} =$ autoregressive parameter (matrix)
    - $\rho = 0 \rightarrow$ independence MH

Autoregressive Metropolis-Hastings algorithm

- draw MCMC sample from $\widetilde{\pi}(\cdot, \boldsymbol{Y}) = \exp\{\widetilde{l}(\cdot)\}$
  - restrict sample to $\widehat{C}_R(\alpha')$
- Metropolis-Hastings candidate:
  $\boldsymbol{\beta}^c = \boldsymbol{\mu} + \boldsymbol{\rho}(\boldsymbol{\beta}^{(t)} - \boldsymbol{\mu}) + \boldsymbol{e}_t$
  - $\boldsymbol{\mu} =$ location parameter
  - $\boldsymbol{\rho} =$ autoregressive parameter (matrix)
    - $\rho = 0 \rightarrow$ independence MH
    - $\rho = 1 \rightarrow$ random-walk MH

Autoregressive Metropolis-Hastings algorithm

- draw MCMC sample from $\widetilde{\pi}(\cdot, \boldsymbol{Y}) = \exp\{\widetilde{l}(\cdot)\}$
  - restrict sample to $\widehat{C}_R(\alpha')$
- Metropolis-Hastings candidate:
  $\boldsymbol{\beta}^c = \boldsymbol{\mu} + \boldsymbol{\rho}(\boldsymbol{\beta}^{(t)} - \boldsymbol{\mu}) + \boldsymbol{e}_t$
  - $\boldsymbol{\mu} =$ location parameter
  - $\boldsymbol{\rho} =$ autoregressive parameter (matrix)
    - $\rho = 0 \rightarrow$ independence MH
    - $\rho = 1 \rightarrow$ random-walk MH
  - $\boldsymbol{e}_t$'s are *i.i.d.* from density g

Autoregressive Metropolis-Hastings algorithm

- draw MCMC sample from $\widetilde{\pi}(\cdot, \boldsymbol{Y}) = \exp\{\widetilde{l}(\cdot)\}$
  - restrict sample to $\widehat{C}_R(\alpha')$
- Metropolis-Hastings candidate:
  $\boldsymbol{\beta}^c = \boldsymbol{\mu} + \boldsymbol{\rho}(\boldsymbol{\beta}^{(t)} - \boldsymbol{\mu}) + \boldsymbol{e}_t$
  - $\boldsymbol{\mu} =$ location parameter
  - $\boldsymbol{\rho} =$ autoregressive parameter (matrix)
    - $\rho = 0 \rightarrow$ independence MH
    - $\rho = 1 \rightarrow$ random-walk MH
  - $\boldsymbol{e}_t$'s are *i.i.d.* from density g
- if the candidate is accepted, then $\boldsymbol{\beta}^{(t+1)} = \boldsymbol{\beta}^c$

Autoregressive Metropolis-Hastings algorithm

- draw MCMC sample from $\widetilde{\pi}(\cdot, \boldsymbol{Y}) = \exp\{\widetilde{l}(\cdot)\}$
  - restrict sample to $\widehat{C}_R(\alpha')$
- Metropolis-Hastings candidate:
  $\boldsymbol{\beta}^c = \boldsymbol{\mu} + \boldsymbol{\rho}(\boldsymbol{\beta}^{(t)} - \boldsymbol{\mu}) + \boldsymbol{e}_t$
  - $\boldsymbol{\mu} =$ location parameter
  - $\boldsymbol{\rho} =$ autoregressive parameter (matrix)
    - $\rho = 0 \rightarrow$ independence MH
    - $\rho = 1 \rightarrow$ random-walk MH
  - $\boldsymbol{e}_t$'s are *i.i.d.* from density g
- if the candidate is accepted, then $\boldsymbol{\beta}^{(t+1)} = \boldsymbol{\beta}^c$
- otherwise, $\boldsymbol{\beta}^{(t+1)} = \boldsymbol{\beta}^{(t)}$

Applications in Environmental Engineering

- few statisticians are working on environmental engineering problems

Applications in Environmental Engineering

- few statisticians are working on environmental engineering problems
- environmental engineers typically use ad hoc and inefficient statistical methods

Applications in Environmental Engineering

- few statisticians are working on environmental engineering problems
- environmental engineers typically use ad hoc and inefficient statistical methods
- modern statistical techniques such as variance functions, transformations, spatial-temporal models potentially offer substantial improvements

GLUE

- GLUE = Generalized Likelihood Uncertainty Estimation

GLUE

- GLUE = Generalized Likelihood Uncertainty Estimation
- widely used

GLUE

- GLUE = Generalized Likelihood Uncertainty Estimation
- widely used
- apparently considered state-of-the-art by many environmental engineers

GLUE

- GLUE = Generalized Likelihood Uncertainty Estimation
- widely used
- apparently considered state-of-the-art by many environmental engineers
- replaces the likelihood function of iid normal errors with an arbitrary objective function

GLUE

- GLUE = Generalized Likelihood Uncertainty Estimation
- widely used
- apparently considered state-of-the-art by many environmental engineers
- replaces the likelihood function of iid normal errors with an arbitrary objective function
- shows no appreciation of maximum likelihood as a general method

GLUE

- GLUE = Generalized Likelihood Uncertainty Estimation
- widely used
- apparently considered state-of-the-art by many environmental engineers
- replaces the likelihood function of iid normal errors with an arbitrary objective function
- shows no appreciation of maximum likelihood as a general method
- objective function is not based on the data-generating probability model

Synthetic data example: Chemical spill

- To test algorithm:

Synthetic data example: Chemical spill

- To test algorithm:
  - use computationally inexpensive function

Synthetic data example: Chemical spill

- To test algorithm:
  - use computationally inexpensive function
  - then approximate and exact result can be compared

Synthetic data example: Chemical spill

- To test algorithm:
  - use computationally inexpensive function
  - then approximate and exact result can be compared
- chemical accident caused spill at two locations on a long channel

Synthetic data example: Chemical spill

- To test algorithm:
  - use computationally inexpensive function
  - then approximate and exact result can be compared
- chemical accident caused spill at two locations on a long channel
  - mass M spill at location 0 at time 0

Synthetic data example: Chemical spill

- To test algorithm:
  - use computationally inexpensive function
  - then approximate and exact result can be compared
- chemical accident caused spill at two locations on a long channel
  - mass M spill at location 0 at time 0
  - mass M spill at location L and time τ

Synthetic data example: Chemical spill

- To test algorithm:
  - use computationally inexpensive function
  - then approximate and exact result can be compared
- chemical accident caused spill at two locations on a long channel
  - mass M spill at location 0 at time 0
  - mass M spill at location L and time τ
- diffusion coefficient is d

Synthetic data example: Chemical spill

- To test algorithm:
  - use computationally inexpensive function
  - then approximate and exact result can be compared
- chemical accident caused spill at two locations on a long channel
  - mass M spill at location 0 at time 0
  - mass M spill at location L and time τ
- diffusion coefficient is d
- parameter vector is $\boldsymbol{\beta} = (m, d, l, \tau)^T$

Synthetic data example: Chemical spill

- To test algorithm:
  - use computationally inexpensive function
  - then approximate and exact result can be compared
- chemical accident caused spill at two locations on a long channel
  - mass M spill at location 0 at time 0
  - mass M spill at location L and time τ
- diffusion coefficient is d
- parameter vector is $\boldsymbol{\beta} = (m, d, l, \tau)^T$
- want estimate of average concentration at end of channel

Synthetic data example: Chemical spill

- To test algorithm:
  - use computationally inexpensive function
  - then approximate and exact result can be compared
- chemical accident caused spill at two locations on a long channel
  - mass M spill at location 0 at time 0
  - mass M spill at location L and time τ
- diffusion coefficient is d
- parameter vector is $\boldsymbol{\beta} = (m, d, l, \tau)^T$
- want estimate of average concentration at end of channel
- l is of special interest

Synthetic data example: Chemical spill

- To test algorithm:
  - use computationally inexpensive function
  - then approximate and exact result can be compared
- chemical accident caused spill at two locations on a long channel
  - mass M spill at location 0 at time 0
  - mass M spill at location L and time τ
- diffusion coefficient is d
- parameter vector is $\boldsymbol{\beta} = (m, d, l, \tau)^T$
- want estimate of average concentration at end of channel
- l is of special interest
- need assessments of uncertainty as well

Chemical spill model

- Model is:

$$
\begin{aligned}
C(s,t;M,D,L,\tau) &= \frac{M}{\sqrt{4\pi D t}} \exp\left[\frac{-s^2}{4Dt}\right] \\
&+ \frac{M}{\sqrt{4\pi D(t-\tau)}} \exp\left[\frac{-(s-L)^2}{4D(t-\tau)}\right] \cdot \mathbb{I}(\tau < t)
\end{aligned}
$$

Details of simulation

- assume data is collected at spatial location 0 (0.5) 2.5 and times 0.3 (0.3) 60 (5 time 200 observations)
- assume that a major goal is to estimate average concentration of time interval [40, 140] at the end of the channel ($s = 3$), specifically

$$F(\boldsymbol{\beta}) = \sum_{i=0}^{20} f\{(3, 40 + 5i), \boldsymbol{\beta}\}$$

  - requires additional function evaluations (but not much more computation)

Details, continued

- $\lambda = 0.333$ in COIL family
- one chemical species
- σ can be integrated out of the posterior analytically

Posterior densities: components of $\boldsymbol{\beta}$

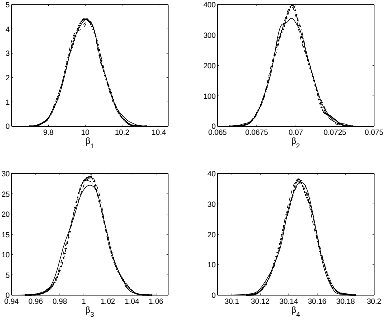

Figure: Kernel estimates of the posterior densities of $\boldsymbol{\beta}_i$'s with the exact joint posterior (solid line) and RBF approximations to joint posterior (dashed line), pseudoposterior (dashed-dotted line), profile posterior with and without Laplace correction (dotted and large dotted lines, respectively).

Posterior densities: $F(\beta)$

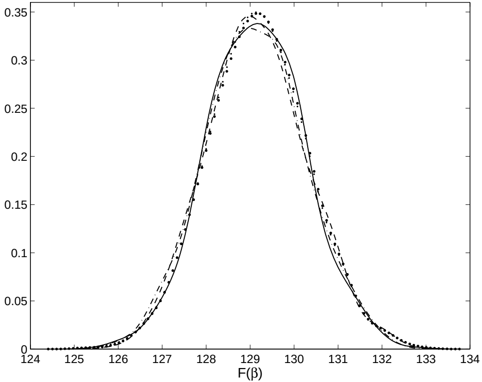

Figure: Kernel smoothed density estimates for the posterior of $F(\beta)$.

Results of a Monte Carlo experiment

	true	MC mean		ratio of C.I. lengths		
		exact	RBF	size .9	size .95	size .99
$\beta_1 = M$	10	10.0057 (.0866)	10.0061 (.0893)	.9969 (.0602)	.9961 (.0624)	.9844 (.0738)
$\beta_2 = D$	.07	.07008 (.00097)	.07008 (.00101)	.9910 (.0592)	.9888 (.0612)	.9687 (.0673)
$\beta_3 = L$	1	1.0005 (.0136)	1.0005 (.0134)	.9671 (.0785)	.9662 (.0765)	.9604 (.0750)
$\beta_4 = \tau$	30.16	30.1610 (.0096)	30.1610 (.0096)	.9786 (.0779)	.9709 (.0818)	.9403 (.0835)
$F(\beta)$	128.998	129.063 (1.087)	129.067 (1.100)	.9959 (.062)	.9937 (.0628)	.9841 (.0695)

Results of a Monte Carlo experiment

Table: Observed coverage probabilities.

	size .9 cred. int.		size .95 cred. int.		size .99 cred. int.	
	exact	RBF	exact	RBF	exact	RBF
β_1	.905 (.009)	.904 (.009)	.950 (.007)	.944 (.007)	.986 (.004)	.990 (.003)
β_2	.908 (.009)	.903 (.009)	.954 (.007)	.951 (.007)	.991 (.003)	.987 (.004)
β_3	.916 (.009)	.899 (.010)	.953 (.007)	.954 (.007)	.989 (.003)	.988 (.003)
β_4	.904 (.009)	.909 (.009)	.947 (.007)	.945 (.007)	.988 (.003)	.987 (.004)
$F(\beta)$	.904 (.009)	.902 (.009)	.947 (.007)	.937 (.008)	.994 (.002)	.980 (.004)

What have we achieved?

In this research we have:

- applied modern statistical tools to calibration of environmental engineering models, e.g.,

What have we achieved?

In this research we have:

- applied modern statistical tools to calibration of environmental engineering models, e.g.,
  - transform-both-side

What have we achieved?

In this research we have:

- applied modern statistical tools to calibration of environmental engineering models, e.g.,
  - transform-both-side
  - spatial-temporal correlation models

What have we achieved?

In this research we have:

- applied modern statistical tools to calibration of environmental engineering models, e.g.,
  - transform-both-side
  - spatial-temporal correlation models
- implemented a Bayesian method of uncertainty analysis

What have we achieved?

In this research we have:

- applied modern statistical tools to calibration of environmental engineering models, e.g.,
  - transform-both-side
  - spatial-temporal correlation models
- implemented a Bayesian method of uncertainty analysis
- substantially reduced the number of evaluations of the computationally expensive environmental model

Race Brook, Nov 5, 2005