

Uncertainty Analysis for Computationally Expensive Models

David Ruppert

Dept. of Statistical Science and School of Operations Research and Information Engineering, Cornell University

July 30, 2012

- **Calibration:** estimate parameters in a model
- **Uncertainty analysis:** confidence or credible region, etc.
- Bayesian modeling and MCMC are particularly suitable for the calibration and uncertainty analysis
- A standard implementation requires the evaluation of a model (simulator) at each MCMC iteration
 - but often the model is computationally expensive
- A computationally feasible approach uses an emulator (interpolant) in place of the simulator

- The emulator must be developed using a relatively small number of simulator evaluations
- These evaluations should be concentrated in the high posterior density region (HPDR)
 - The HPDR could be less than 1% of the parameter space
 - the location and shape of the HPDR is not known in advance
- Evaluations that are close to each other in the parameter space are wasteful

- Our algorithm iterates between
 - using the current emulator to select new points for running the model
 - updating the emulator using the new evaluations
- Except for a paper of Rasmussen, we are not aware of other work where the emulator is built on a small and a priori unknown set

SOARS = Statistical and Optimization Analysis using Response Surfaces

- SOARS has 4 steps and iterates between the final 3 steps
 - 1 locate the posterior mode using global optimization
 - 2 explore the region around the mode to learn the size and shape of the HPDR using GRIMA (Grow the (design) Region and IMprove the Approximation) (Bliznyuk et al., 2012)
 - 3 construct a Radial Basin Function (RBF) emulator (response surface) of the log posterior
 - 4 run MCMC using the emulator

- $\mathbf{Y}_i = (Y_{i,1} \dots, Y_{i,d})^\top$, $i = 1, \dots, n$, is a multivariate time series
- $\mathbf{f}_i(\boldsymbol{\beta}) = (f_{i,1}(\boldsymbol{\beta}), \dots, f_{i,d}(\boldsymbol{\beta}))^\top$ is the simulator output for time i
- $\boldsymbol{\beta}$ is the vector of unknown parameters in the simulator.
- In the absence noise we expect that

$$\mathbf{Y}_i = \mathbf{f}_i(\boldsymbol{\beta})$$

- Noise can be modeled using standard techniques such as
 - transformations
 - variance functions
 - time series models

Example: Town Brook watershed

- Town Brook is in the Cannonsville watershed, part of the NYC water supply
- Town Brook is a small watershed so works well as a case study
 - MCMC using the exact posterior is feasible, although it takes over a week
 - therefore, SOARS can be compared with an exact implementation

Town Brook watershed: data and simulator

- $\mathbf{Y}_i = (Y_{i,1}, Y_{i,2})^T = (\text{flow, concentration of phosphorus})$ on i th day
- $f_i(\boldsymbol{\beta})$ is output from SWAT2005 (Soil and Water Assessment Tool, 2005 version)
 - SWAT takes seconds to run on the Town Brook watershed
 - SWAT will take minutes or hours on larger watersheds
 - $\boldsymbol{\beta}$ is vector of eight parameters in the SWAT model

SOARS Step 1: Optimization

- θ is the parameter vector containing β and parameters in the noise model
- $\pi(\theta|\mathbf{Y})$ is the unnormalized posterior = likelihood \times prior density
- The goal is to find, characterize, and construct the emulator on a HPDR
 - e.g., a $1 - \alpha$ credible region
- The HPDR is located by using a global maximizer to find the posterior mode
 - high accuracy is not important
 - we only need to get into $C_R(\alpha)$, not find the mode

- After optimization, but before GRIMA, evaluate the log-likelihood on a Latin hypercube centered at the (approximate) mode
- GRIMA produces a nested sequence $\mathcal{D}_0, \mathcal{D}_1, \dots$ of sets of evaluation points
- \mathcal{D}_0 is the set of evaluation points from optimization plus those from the Latin hypercube
 - except “outliers” (outside the HPDR) are excluded

SOARS Step 2: GRIMA, continued

- Given the current set \mathcal{D}_i , let \mathcal{C} be the set of parameter values whose distance from \mathcal{D}_i is exactly r .
 - r is a tuning parameter that varies during GRIMA
- Let $\tilde{\ell}_i$ be the emulator of the log-posterior on \mathcal{D}_i .
- The candidate for the next evaluation point is the point in \mathcal{C} where $\tilde{\ell}_i$ is maximized.
- Because this point is exactly at distance r from \mathcal{D}_i , it is neither
 - redundant (too close to the current evaluation points) nor
 - well outside the HPDR (too far from them)

- GRIMA allows r to increase initially so that the entire HPDR is covered quickly
- Eventually r decreases so that the set of evaluation points becomes dense

SOARS Step 3: RBF interpolation

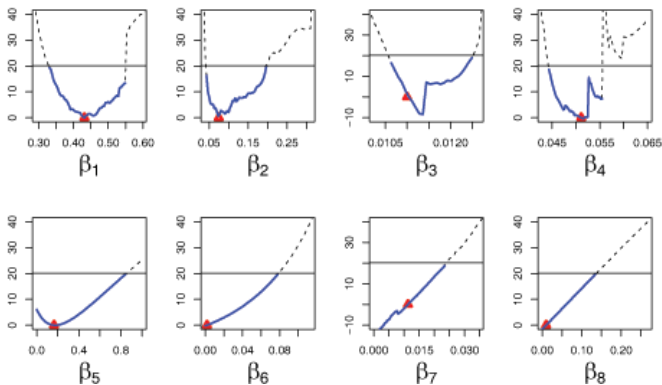
- the RBF response surface is updated repeatedly
 - Bliynyuk et al. (2012) have an efficient algorithm for updating
- RBF interpolation is sensitive to the parametrization and is improved by sphering

- MCMC using the emulator is run **after** GRIMA terminates to estimate the posterior
- MCMC is also used **during** GRIMA to decide when to terminate
 - termination occurs when the total variation norms between successive estimates of the univariate log posterior densities are small
 - norms estimated by importance sampling
- MCMC is also used **during** GRIMA to estimate the HPDR

- $h(\mathbf{Y}_i, \boldsymbol{\lambda}) = h\{\mathbf{f}_i(\boldsymbol{\beta}), \boldsymbol{\lambda}\} + \boldsymbol{\varepsilon}_i$ (transform-both-sides)
 - $h(\mathbf{y}, \boldsymbol{\lambda}) = \{h(y_1, \lambda_1) \cdots h(y_d, \lambda_d)\}^T$
 - $h(y, \lambda) = (1 - \Delta)h_{BC}(y, \lambda) + \Delta \log(y)$
 - $h_{BC}(y, \lambda)$ is the Box-Cox family
 - therefore, $\boldsymbol{\varepsilon}_i$ can be (multivariate) Gaussian
- $\boldsymbol{\varepsilon}_i = \boldsymbol{\Phi}\boldsymbol{\varepsilon}_{i-1} + \mathbf{u}_i$ (vector AR(1))
 - \mathbf{u}_i is Gaussian white noise with covariance matrix $\boldsymbol{\Sigma}_u$
- noise parameters are $\boldsymbol{\lambda}$, $\boldsymbol{\phi}$, and $\boldsymbol{\Sigma}_u$

- Optimization was done with DSS, a global optimizer
- 1900 function evaluations were used
 - problem: SWAT output is nonsmooth with many local maxima and 8 parameters
- For a more computationally expensive simulator, one would need to parallelize the optimization

Town Brook Profile Plots



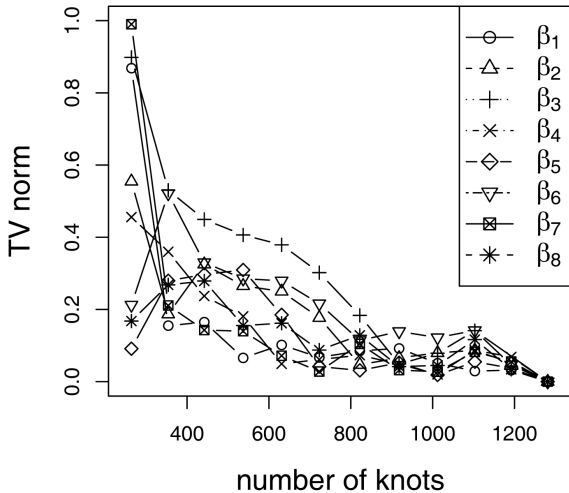
These plots use the exact unnormalized posterior

Parameters are varied one at a time about

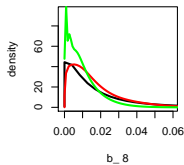
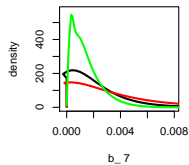
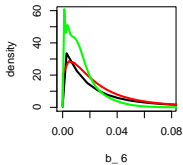
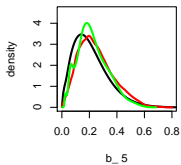
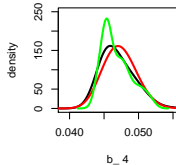
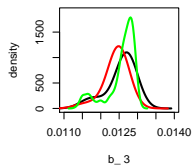
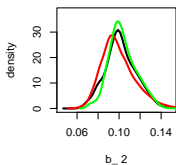
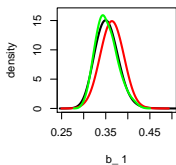
DDS termination point at ▲

- We did 500 evaluations prior to GRIMA with a Latin hypercube design
- GRIMA used a total of 1017 function evaluations
- A total of 3,517 expensive evaluations were used for
 - optimization
 - the Latin hypercube sample, and
 - GRIMA

Town Brook: stopping GRIMA



Town Brook: Accuracy of the Emulator



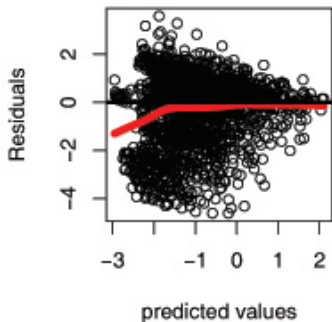
Black: 60,000 MCMC iterations with exact posterior

Red: SOARS (3517 exact plus 60,000 iterations with emulator)

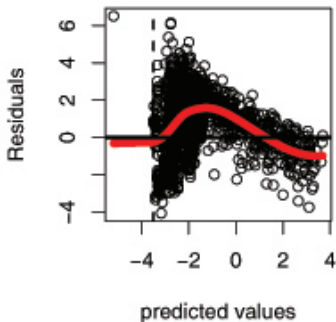
Green: 3500 MCMC iterations with exact posterior

Town Brook: Model Adequacy

(a) Flow



(b) TDP



- SOARS, especially the GRIMA algorithm, can handle the nonsmoothness of SWAT output
- Given a budget for the expensive evaluations, SOARS outperforms standard MCMC
 - it is better to use the expensive evaluations to build the emulator rather than for MCMC itself
- Uncertainty analysis and calibration took about 3,500 evaluations
 - calibration alone took 1,900 evaluations and was not particularly accurate
 - the calibration was improved during the uncertainty analysis

- RBF interpolation suffers from the curse of dimensionality
- The nonsmoothness of SWAT output makes optimization difficult
- Thousands of expensive function evaluations are necessary with an 8-parameter SWAT model
- Parallelization is necessary for larger problems (e.g., more parameters or larger Watershed)

- Christine Shoemaker, Professor, Civil and Environmental Engineering, Cornell University
- Nikolay Bliznyuk, Assistant Professor, Statistics, University of Florida
- Yingxing Li, Assistant Professor, Xiamen University
- Yilun Wang, Associate Professor, University of Electronic Science and Technology of China