Project Title

Characterizing Uncertainty in the Inputs of Simulation Code.

Executive Summary

We propose to work on the problem of calibrating the parameters of computer code used for simulation of physical phenomena. We will explore statistical methods based on a Bayesian approach implemented with Sampling Importance Resampling (SIR). This provides a fast approximation to the posterior distribution of the computer model parameters in situations where the code has been evaluated over a dense grid of points. The particular application that we will consider corresponds to a physical problem where the compression of a gas is measured as a function of a given shock. This is a challenging calibration problem since both input and output consist of fucntions of time. The project will be carried out by the PIs Herbert Lee and Bruno Sansó and a graduate student under their joint supervision from AMS at UCSC, in collaboration with David Higdon, from Statistical Sciences, CCS-6 at LANL, and Charles Nakhleh, a physicist in X-2 at LANL.

This problem is important to scientists at Los Alamos as the study of sudden forces on materials is critical to much of the design work at the Lab. This particular simulation is the precursor to a full physical experiment that Los Alamos hopes to run in the near future. Developing methodology for solving the inference problem for the computer model is necessary both to help determine aspects of the design of the physical experiment, as well as for the analysis of the data that would be produced from the experiment. Thus success in this proposed collaboration has clear potential for deeper future collaboration as the experiment progresses.

Project Description

Computer code based on mathematical models is frequently used for the description of physical phenomena. In situations where data are difficult or expensive to obtain, computer models are used as proxies for direct observations. Usually the code is highly complex and computationally demanding and depends on a number of inputs that need to be tuned in order for the code to produce realistic simulations. This is done by comparing the computer model output to observational data and trying to find the combinations of input parameters that produce the best possible match. In most realistic applications this is an ill posed problem that may not have a unique solution. An additional problem is that the process of calibrating the parameters usually requires the evaluation of the computer code over a dense grid of input values. This may be impossible to do due to time and computational constraints. A possible solution is to build a statistical equivalent model (SEM) that provides a fast to evaluate approximation to the computer model (Sacks et al., 1989). Unfortunately this strategy introduces a source of error that needs to be accounted for. Another source of uncertainty is the measurement error that may be present in the observations.

In summary, effective methods for the calibration of computer model parameters need to tackle three problems: non-uniqueness of the solutions, slow and expensive evaluations of the code and uncertainty induced by noise observations. Recent developments in the literature (Kennedy and O'Hagan, 2001; Higdon et al., 2003; Lee et al., 2007) consider Bayesian methods that provide the ability to express prior knowledge on the computer model parameters using probability distributions. Such prior distributions can be used to constrain the parameter space and provide information about the most likely values, thus solving the identificability problem. On the other hand, hierarchical statistical models can be used to create effective SEMs that incorporate prior and measurement error uncertainties and propagate them to produce probabilistic inference on the input parameters. The solution provided by Bayesian methods to the calibration problem consists of a probability distribution for the computer model parameter. This is used to determine how likely a certain range of values is.

The Statistical Sciences group at LANL has recently given us access to Matlab computer code that simulates an isentropic compression experiment. Here the hydrodynamic system evolution is modeled following the methods on Kuropatenko viscosity detailed by Caraman et al. (1998). A force applied along one boundary is propogated through the medium, and the resulting effect on the system is given by the code. The goal is to infer the unknown and unobservable inputs using the observable outputs and the properties of the medium. In this setting, both the inputs and outputs take functional forms, which is more complicated and computationally intensive than the traditional settings in the literature. Hence new methodology is needed to analyze such experiments.

As just mentioned, the additional difficulty in this experiment is that both the parameter that controls the computer simulator and the output obtained from it, are infinitely dimensional. Thus the key to a successful calibration of the model is to obtain a parsimonious representation of the input and output functions. Since input shocks have a clearly defined structure, we will explore parametric represention of those curves based on two or three parameters. We propose the use of process convolutions to describe the output, since these provide parsimonious yet flexible representations of Gaussian processes.

The setting of our computer experiment is that there is some true process, $\zeta(\boldsymbol{\theta}, t)$, that in the real world characterizes the functional relationship between sets of inputs $\{\boldsymbol{\theta}, t\}$ and $y(t)$. $\zeta(\boldsymbol{\theta}, t)$ is simulated with a computer code producing $\eta(\boldsymbol{\theta}, t)$. There is a single data point consisting of a set of observations in time $y(t)$. These correspond to the unknown functional input $\boldsymbol{\theta}$. So we assume that the computer simulator produces an unbiased approximation to the true functional output and the observations $y(t) = \eta(\boldsymbol{\theta}, t) + \varepsilon(t)$.

The posterior density (ignoring for now the many possible nuisance parameters) is then of the form

$$P(\boldsymbol{\theta}|y, \eta) \propto \mathcal{L}(\boldsymbol{\theta}|y, \eta)\pi(\boldsymbol{\theta}),$$

where $\mathcal{L}$ denotes the likelihood and $\pi(\boldsymbol{\theta})$ the denotes the initial distribution on $\boldsymbol{\theta}$ encompassign our prior knowledge on $\boldsymbol{\theta}$. Thus, for Gaussian error, we have $\mathcal{L}(\boldsymbol{\theta}|y, \eta) \propto \exp\left(-\frac{1}{2\sigma^2}\sum_t \varepsilon(t)^2\right)$. Often it is not possible to find an analytic solution for the posterior of $\boldsymbol{\theta}$. Common strategies for inference center on Markov Chain Monte Carlo methods. If η is

easy to evaluate, this is straightforward. If η is expensive to run, then we need a SEM based on a limited number of runs to calculate the likelihood. When the number of simulation runs is large, building a SEM can be computational very expensive. In this proposal we will focus on using SIR to avoid building a SEM for a large number of computer model simulations.

Sampling Importance Resampling

While the traditional approach to fully Bayesian modeling for inverse problems involves Markov chain Monte Carlo methods, such methods can be prohibitively expensive computationally, particularly when large numbers of model runs are available. Instead, we turn to Sampling Importance Resampling (SIR). SIR is a fast method for sampling from a distribution (Rubin, 1988). Given a target distribution F with density $f(\theta)$, is possible to obtain a sample of F by sampling n points, say $\theta_1, \ldots, \theta_n$, from a distribution with density g, calculate the weights $w_i = f(\theta_i)/g(\theta_i)$ and resample the θ_i with replacement using such weights. In our setting the target distribution has a density that is proportional to $\mathcal{L}(\boldsymbol{\theta}|y, \eta)\pi(\boldsymbol{\theta})$.

The inverse likelihood is inexpensive to evaluate at any input location where the simulator has already been run. Suppose that the inputs for our bank of computer output were sampled independently from some distribution defined by the density g. Then we can resample from the posterior after designating the sampling weights,

$$w(\theta_i) = \frac{f(\theta_i)}{g(\theta_i)} = \frac{\pi(\theta_i)\mathcal{L}(y|\eta(x,\theta_i))}{g(\theta_i)} \quad .$$

Thus application of the SIR algorithm is straightforward and we are able to obtain a discrete approximation to the inverse problem posterior, without having to re-run the computer simulator. Alternatively, we can use these inverse importance weights in a Monte Carlo integration for any point estimation. In the presence of nuisance parameters, we can sample from the hyperprior at some subset of the resampling iterations, and couple these values with the sample from $g(\theta)$ for resampling.

In order to calculate these weights, we need to know $g(\theta_i)$ at each $\theta_i \in \mathrm{S}$. Often, with computer models where the input configuration has been decided by the user, we will have no knowledge (at least no usable knowledge) about the nature of g. In fact, it may seem odd to assume that the sampling was random at all. However, the role of g in the weights is to counter the effect of the original sampling on any posterior estimate, and this remains the case whether or not we believe that g truly describes the sampler's intent. In the case where the variables θ_i are discrete with a manageable support, we can compute the empirical probability function to estimate the $g(\theta_i)$ marginals. When this is not possible, we will use a Kernel Density Estimate (KDE). For Normal kernels, this generally describes estimates of the sort

$$\hat{g}(\theta) = \tfrac{1}{n}\textstyle\sum_j \mathrm{N}(\theta|m_j, Vh^2)$$

V is an estimate of the variance of $g(t)$, h is a smoothing parameter or bandwidth, and m_j is a location dependent upon θ_j. The version which we use below, with shrinkage for the

individual means, is described in West (1993).

$$\begin{aligned} \mathbf{m}_j &= \theta_j\sqrt{1-h^2} + \bar{\theta}(1-\sqrt{1-h^2}) \\ h &= \left(\frac{4}{n(1+2p)}\right)^{\frac{1}{1+4p}}, \quad p = \dim(\theta) \end{aligned}$$

The literature on KDE methods is vast, and the best choice will be application specific. See the books by Bowman & Azzalini (1997) and Simonoff (1996) for examples.

Personnel

PIs: Herbert Lee (UCSC) and Bruno Sansó (UCSC) in collaboration with David Higdon (LANL) and Charles Nakhleh (LANL). One AMS graduate student.

References

Bowman, A. W. and Azzalini, A. (1997). *Applied Smoothing Techniques for Data Analysis*. Oxford University Press.

Caraman, Shashkov, and Whalen (1998). Formulations of artificial viscosity for multi-dimensional shock wave computations. *Journal of Computational Physics*, 144:70–97.

Higdon, D., Lee, H., and Holloman, C. (2003). Markov chain Monte Carlo-based approaches for inference in computationally intensive inverse problems. *Bayesian Statistics*, 7.

Kennedy, M. and O'Hagan, A. (2001). Bayesian calibration of computer models. *Journal of the Royal Statistical Society, Series B Statistical Methodology*, 63:425–464.

Lee, H., Sansó, B., Zhou, W., and Higdon, D. (2007). Inference for a proton accelerator using convolution models. *Journal of the American Statistical Association*. to appear.

Rubin, D. (1988). Using the SIR algorithm to simulate posterior distributions by data augmentation. In Bernardo, J., DeGroot, M., and Lindley, D. amd Smith, A., editors, *Bayesian statistics 3*. Oxford University Press Inc.

Sacks, J., Welch, W., Mitchell, T., and Wynn, H. (1989). Design and analysis of computer experiments. *Statistical Science*, 4:409–435.

Simonoff, J. S. (1996). *Smoothing Methods in Statistics*. Springer-Verlag.

West, M. (1993). Approximating posterior distributions by mixtures. *Journal of the Royal Statistical Society, Series B, Methodological*, 55:409–422.