A Parametric Bootstrap Method for Uncertainty Quantification in Ill-Posed, Inverse Problems with Poisson Noise

Johnathan M. Bardsley\textsuperscript{a,*}, Heikki Haario\textsuperscript{b,c}, and Antti Solonen\textsuperscript{b,c}

\textsuperscript{a}Dept. of Mathematical Sciences, Univ. of Montana, Missoula, Montana, USA;
\textsuperscript{b}Dept. of Mathematics and Physics, Lappeenranta Univ. of Technology, Lappeenranta, Finland;
\textsuperscript{c}Finnish Meteorological Institute, Helsinki, Finland.

(Received 00 Month 20XX; final version received 00 Month 20XX)

For large-scale, ill-posed inverse problems, regularization is required in order to stabilize the inversion process. In the Bayesian setting, regularization corresponds to the choice of the prior probability density function, which incorporates both prior knowledge and uncertainty about the unknown. The assessment of uncertainty in the Bayesian setting must therefore take into account both observation and prior model uncertainties. The classical parametric bootstrap is a Monte Carlo method that estimates parameter uncertainty by repeatedly resampling observations and computing corresponding parameter estimates. It has the advantage that each sample is computed by solving an optimization problem, which for many large-scale problems is significantly more efficient than implementing a Markov chain Monte Carlo method. However, the classical bootstrap samples do not reflect the prior model uncertainties. In this paper, we extend the classical parametric bootstrap to the Bayesian setting, so that it takes into account both observation and prior uncertainties. We motivate our approach by showing that sampling from the posterior density function in the linear, Gaussian case can be viewed as a parametric bootstrap method, and then extend the approach in a natural way to the Poisson noise case. The resulting bootstrap sample density is not equal to the posterior density function, however, the resulting samples provide valuable information about parameter uncertainty in instances where implementing a Markov chain Monte Carlo method is too difficult or computationally intensive. Several numerical experiments are included to justify the method.

Keywords: inverse problems, bootstrap, uncertainty quantification

AMS Subject Classification: 15A29, 65F22, 65C05, 65C60, 94A08

1. Introduction

In this manuscript, we focus on linear inverse problems of first kind Fredholm type:

\[ b(s) = Ax(s) \overset{\text{def}}{=} \int_{\Omega} a(s,t)x(t)dt, \]  

where \( s,t \in \Omega \subset \mathbb{R}^d \) is the computational domain; \( b(s) \) is a function that represents the observed image; \( a(s,t) \) is the kernel; and \( x(t) \) is the unknown that we wish to estimate.

In practice, images are only measured at discrete points, and hence one works with a numerically discretized version of (1). Moreover, the observations contain random errors that must be modeled statistically. Assuming independent and identically distributed Gaussian measurement noise yields

\[ b = Ax + e, \]

\textsuperscript{*}Corresponding author. Email: bardsleyj@mso.umt.edu
where \( x \in \mathbb{R}^n \) and \( b \in \mathbb{R}^m \) are discretizations of \( x \) and \( b \) in (1); \( A \) is the real valued \( m \times n \) matrix that arises when approximating the integration operation in (1) with a quadrature rule; and \( e \sim \mathcal{N}(0, \lambda^{-1}I) \), where the inverse-variance parameter \( \lambda \) is known as the precision.

Noise model (2) will figure prominently in the discussion that follows, however the new results of the paper will relate to the Poisson noise model, which is used in both astronomical and medical imaging [6, 10]:

\[
b = \text{Poiss}(Ax + g),
\]  

(3)

where \( g \) is the \( m \times 1 \) vector of background counts and is assumed known.

For the above two probability models, the probability density functions \( p(b|x) \) have the form

\[
p(b|x) \propto \exp \left( -\frac{\lambda}{2} \|Ax - b\|^2 \right),
\]  

(4)

for noise model (2), and

\[
p(b|x) \propto \exp \left( -\sum_{j=1}^{n^2} \left\{ ([Ax]_j + g_j) + b_j \ln([Ax]_j + g_j) \right\} \right),
\]  

(5)

for noise model (3). In both cases, the maximum likelihood estimators (i.e. the maximizers of \( p(b|x) \) in (4) and (5) with respect to \( x \)) are unstable with respect to the noise in the data \( b \). Such instability is a characteristic of ill-posed inverse problems and is due to the fact that for models of the form (1), \( b \) and \( x \) live in infinite dimensional function spaces, and \( A \) is a compact operator between these two spaces [12]. The standard technique for overcoming this instability is regularization [12].

We motivate regularization from a Bayesian point of view. Specifically, we assume a Gaussian prior of the form

\[
p(x|\delta) \propto \delta^{n/2} \exp \left( -\frac{\delta}{2} x^T L x \right),
\]  

(6)

where \( \delta \) is the prior precision. Such a prior arises, for example, if we model \( x \) using a Gaussian Markov random field [3]. In this case, statistical assumptions are made about the values of the unknown elements \( x_i \) of \( x \) based the values of neighboring elements. The matrix \( L \) is determined by the neighborhood relationships that are assumed, while \( \delta \) controls the strength of the relationship between \( x_i \) and its neighborhoods. Thus the prior incorporates both prior knowledge and uncertainty.

With the prior in hand, Bayes' Law gives us the posterior density function, which we write in least squares form:

\[
p(x|b, \lambda, \delta) \propto p(b|x, \lambda)p(x|\delta)
\]

\[
\propto \exp \left( -\frac{\lambda}{2} \|Ax - b\|^2 - \frac{\delta}{2} x^T L x \right)
\]

\[
= \exp \left( -\frac{1}{2} \|\hat{A}x - \hat{b}\|^2 \right),
\]

2
where
\[ \hat{b} = \begin{bmatrix} \lambda^{1/2} b \\ 0 \end{bmatrix}_{(n+m) \times 1} \quad \text{and} \quad \hat{A} = \begin{bmatrix} \lambda^{1/2} A \\ \delta^{1/2} L^{1/2} \end{bmatrix}_{(n+m) \times n}. \]

Samples from \( p(x|b, \lambda, \delta) \) can be computed by solving the optimization problem
\[
x = \arg\min_x \| \hat{A} \hat{x} - (\hat{b} + \hat{e}) \|_2^2, \quad \hat{e} \sim N(0, I_{(n+m) \times 1}). \tag{7}
\]

This optimization-based approach for sampling from \( p(x|b, \lambda, \delta) \) is also used in [2]. To see that (7) yields samples from \( p(x|b, \lambda, \delta) \), note that the solution of the normal equations for (7) is given by
\[
x = (\hat{A}^T \hat{A})^{-1} \hat{A}^T (\hat{b} + \hat{e}) \\
= (\lambda A^T A + \delta L)^{-1} \lambda A^T b + w, \quad w \sim N(0, (\lambda A^T A + \delta L)^{-1}), \tag{8}
\]
which has probability density \( p(x|b, \lambda, \delta) \).

The key observation regarding (7) is that it is very similar to the parametric bootstrap method [7] applied to the model \( \hat{b} = \hat{A} \hat{x} + \hat{e} \), i.e. to
\[
\begin{bmatrix} \lambda^{1/2} b \\ 0 \end{bmatrix} = \begin{bmatrix} \lambda^{1/2} A \\ \delta^{1/2} L^{1/2} \end{bmatrix} \hat{x} + \hat{e}, \quad \hat{e} \sim N(0, I_{(n+m) \times 1}).
\]

However, there are two main differences. First, in classical parametric bootstrap [7] we would compute an estimator \( \bar{x} \), and then compute the bootstrap samples by repeatedly solving
\[
x = \arg\min_x \| \hat{A} \hat{x} - (\hat{A} \bar{x} + \hat{e}) \|_2^2, \quad \hat{e} \sim N(0, I_{(n+m) \times 1}),
\]
though if \( \bar{x} = (\lambda A^T A + \delta L)^{-1} \lambda A^T b \) is the estimator, one can show that this also has sample density \( p(x|b, \delta) \). Classical bootstrapping is discussed in the inverse problems setting in [1], however the authors only consider examples with a small number of parameters that don’t require regularization (i.e. are not ill-posed), for which the classical bootstrap is ideally suited.

The second difference is that when resampling \( \hat{b} \), we draw from both the observation and prior equations. For classical bootstrap [7], only the observations are resampled, even in the case of ridge regression [11]. However, as we discussed above, the prior incorporates both prior knowledge and uncertainty, and hence, if we only resample the observations, the resulting sample density will not reflect the prior uncertainty. This is supported by the results in Figure 1, where classical bootstrap (resampling only the observations) and our implementation of bootstrapping are compared in the Gaussian noise case. Note that the uncertainty bounds for the classical bootstrap are significantly less than those obtained by sampling from the posterior density function.

Just as the classical parametric bootstrap can be implemented with a Poisson noise model, we extend (7) to the Poisson noise case. The resulting bootstrap sample density will not be equal to posterior density function; indeed, in future work, we hope to derive the form of this density. In this paper, we use the bootstrap samples directly, as they nonetheless provide useful information about the uncertainties in \( x \) in a very efficient manner, by taking advantage of a state of the art numerical optimization method. For the
large-scale problems of interest to us, the cost savings will be significant when compared to applying a Markov chain Monte Carlo method to the same problem.

The remainder of the paper is organized as follows. In Section 2, we present the parametric bootstrap in the Poisson noise case. We then embed it within a hierarchical model, which allows us to estimate $\delta$ as well. We test the resulting bootstrap method on examples from image deblurring and computed tomography in Section 3, where we also numerically compare, for a small-scale example, the bootstrap samples with those obtained from a converged Metropolis-Hastings method for sampling from the posterior density function $p(x|b, \delta)$. Finally, we end with conclusions in Section 4.

2. The Parametric Bootstrap for Inverse Problems with Poisson Data

From here on out, we assume that the data $b$ arises from a Poisson distribution, which is the case in both astronomical imaging and positron emission tomography [6, 10]. Specifically, we assume noise model (3). As above, we assume a Gaussian Markov Random field prior prior $p(x|\delta)$ of the form (6). The corresponding augmented noise model then has the form

$$\begin{bmatrix} b \\ 0 \end{bmatrix} = \begin{bmatrix} \text{Poiss}(Ax + g) \\ \delta^{1/2}L^{1/2}x + e \end{bmatrix}, \quad e \sim N(0, I_n),$$

which has likelihood (also the posterior density) function

$$p(x|b, \delta) \propto \exp \left( -\sum_{i=1}^{n} \{ [Ax]_i + g_i - b_i \ln([Ax]_i + g_i) \} + \frac{\delta}{2} x^T L x \right). \quad (9)$$

Mimicking (7), we compute a bootstrap sample by first resampling data via

$$\begin{bmatrix} \tilde{b} \\ \tilde{e} \end{bmatrix} \sim \begin{bmatrix} \text{Poiss}(b) \\ N(0, \delta^{-1}I) \end{bmatrix}, \quad (10)$$

and then by solving the optimization problem

$$x = \arg\min_{x \geq 0} \left\{ \sum_{i=1}^{n} \{ [Ax]_i + g_i - \tilde{b}_i \ln([Ax]_i + g_i) \} + \frac{\delta}{2} \|L^{1/2}x - \tilde{e}\|^2 \right\}. \quad (11)$$
June 22, 2013 Inverse Problems in Science and Engineering PoissSamp

Staying closer to the classical bootstrap, we could instead first compute an estimator \( \bar{x} \) and then resample instead from

\[
\begin{bmatrix}
\tilde{b} \\
\tilde{e}
\end{bmatrix}
\sim
\begin{bmatrix}
\text{Pois}(A\bar{x} + g) \\
\mathcal{N}(L^{1/2}\bar{x}, \delta^{-1}I)
\end{bmatrix}.
\]

Both approaches work well in practice, but we use (10), since it more closely mimics the Gaussian case (7).

The primary computational challenge in this approach is that (11) is a nontrivial optimization problem; note, in particular, the presence of the nonnegativity constraint. Fortunately, we have an efficient and convergent optimization method for solving this problem, which allows us to compute the bootstrap samples very efficiently. The algorithm is presented in [5] and its convergence is proved for problems of the type (11) in [4]. In all implementations, we use a stopping tolerance of \(10^{-6}\) for the projected gradient norm; see [5] for more details.

Finally, we emphasize that the sample density defined by (11) is not the posterior density (9). To obtain such samples, some other algorithm, such as a Markov chain Monte Carlo (MCMC) method, must be used. However, for the large-scale imaging problems of interest to us, MCMC methods are very computationally intensive to implement. In order to explore the similarity between bootstrap and MCMC samples, in Section 3.1 we present a numerical comparison between the bootstrap samples and samples obtained from a Metropolis-Hastings algorithm. The results suggest that the two densities are nearly identical for that example, but more work must be done before any concrete statements can be made.

### 2.1 Estimating \( \delta \) Using a Hierarchical Model

The prior precision (or regularization) parameter \( \delta \) can also be estimated using bootstrapping. This requires that we assume a probability model for \( \delta \). Following [2], we choose a Gamma distribution

\[
p(\delta) \propto \delta^{\alpha - 1} \exp(-\beta \delta) .
\]

For the results below, we choose \( \alpha = 1 \) and \( \beta = 10^{-4} \), so that \( \delta \) has mean \( \alpha/\beta = 10^4 \) and variance \( \alpha/\beta^2 = 10^8 \), making \( p(\delta) \) quite uninformative, i.e. approximately constant over the possible values of \( \delta \). Given (5), (6) and (13), by Bayes’ law the posterior density function has the form

\[
p(x, \delta | b) \propto p(b | x) p(x | \delta) p(\delta)
\]

\[
\propto \delta^{n/2 + \alpha - 1} \exp\left(-\sum_{i=1}^{n} \left\{ |Ax|_i + g_i - \tilde{b}_i \ln(|Ax|_i + g_i) \right\} - \frac{\delta}{2} x^T L x - \beta \delta \right).
\]

We can then compute a bootstrap sample by resampling as in (10) or (12), and then by solving

\[
(x, \delta) = \arg \min_{x \geq 0, \delta} \left\{ \sum_{i=1}^{n} \left\{ |Ax|_i + g_i - \tilde{b}_i \ln(|Ax|_i + g_i) \right\} + \delta \left( \frac{1}{2} \|L^{1/2}x - \tilde{e}\|^2 + \beta \right) - \left( \frac{n}{2} + \alpha - 1 \right) \ln \delta \right\} ,
\]

5
which can be done simply using coordinate-wise descent as follows.

**An Iterative Method for Solving (15):**

1. Compute \( \tilde{b} \) and \( \tilde{e} \) as in (10) or (12). Choose initial guess \( \delta_0 \), and set \( k = 1 \).
2. Define \( x_k = \arg \max_x p(x|b, \delta_{k-1}) \), i.e., solve (11) with \( \delta = \delta_{k-1} \).
3. Define \( \delta_k = \arg \max_x p(\delta|x_k) = \left( \frac{n^2 + \alpha - 1}{2} \right) \frac{1}{\sqrt{\pi \gamma^2}} \). \( \delta_k \).
4. If stopping tolerances are met, let \( (x_k, \delta_k) \) be your bootstrap sample. Otherwise set \( k = k + 1 \) and return to step 2.

In practice, we’ve found that a few iterations are sufficient provided \( \delta_0 \) is near the minimizer, which is the case when you are computing multiple bootstrap samples and take \( \delta \) to be the mean of the previously sampled values of \( \delta \). This is what we do in our implementations. However, if you already have a reliable estimate for \( \delta \), obtained for example using a method from [4], you could fix \( \delta \) and only use (11), or choose a more informative probability density \( p(\delta) \).

3. Numerical Experiments

We consider three different examples in which data is accurately modeled using a Poisson distribution: image deblurring in one and two dimensions, and positron emission tomography.

3.1 One-Dimensional Image Deblurring

We begin with a one-dimensional example (from [12]). We also used this example in the Introduction and will use a smaller scale version to compare the bootstrap sample density with samples from \( p(x|b, \delta) \) defined by (9) obtained using a Markov chain Monte Carlo method.

The Fredholm integral equation in this case is of the form

\[
b(s) = \int_0^1 A(s - s') x(s') ds',
\]

with a Gaussian convolution kernel \( A(s) = \exp(-s^2/(2\gamma^2))/\sqrt{\pi \gamma^2}, \gamma > 0 \). Then \( A \) has the form

\[
[A]_{ij} = h \exp\left(-((i - j)h)^2/(2\gamma^2)\right)/\sqrt{\pi \gamma^2}, \quad 1 \leq i, j \leq n,
\]

where \( h = 1/n \) with \( n \) the number of grid points in \( [0, 1] \). We use \( n = 80 \), and the resulting \( A \) has full column rank with condition number on the order of \( 10^{16} \), resulting in a severely ill-conditioned problem. The true image is given by the solid line in Figure 2, and the data \( b \) is generated using (3) with background \( g = 10 \cdot 1 \). The scale of the true image was chosen so that the signal-to-noise ratio (SNR) is approximately 25.

We compute 1000 bootstrap samples, with four iterations of the coordinate-wise descent method of the previous subsection for each bootstrap sample. The pixel-wise mean of the samples of \( x \), together with pixel-wise 95% quantiles, are plotted on the upper-right in Figure 2, while on the lower-left is the histogram of the computed values for \( \delta \). Note that the samples for \( x \) are all nonnegative, since they were computed from (11). And finally, we perform the bootstrap sampling in the case that \( \delta \) is fixed to be \( 2.1 \times 10^{-5} \), which is the mean of the bootstrap samples of \( \delta \) computed in the previous example.
Figure 2. In the upper-left, is the one-dimensional true image and blurred noisy data. On the upper-right is
a plot of the mean of the bootstrap samples of $x$ together with 95% credibility bands, with $\delta$ estimated using
bootstrapping. On the lower-left is a histogram of the bootstrap samples of $\delta$ in that case. And finally, on the
lower-right, are the results for the case when the regularization parameter is fixed to be $\delta = 2.1 \times 10^{-5}$.

3.1.1 A Comparison with MCMC samples from $p(x|b, \delta)$

In the case of a linear model and Gaussian noise, we showed in Section 1 that the boot-
strap samples are samples from $p(x|b, \delta)$. However, when the data is Poisson distributed
and the Poisson likelihood is used instead, this is no longer the case. Nonetheless, it
is of interest to compare the results from the bootstrap method with those from a long
MCMC chain with the Metropolis-Hastings (MH) sampler, which we know yields samples
from the $p(x|b, \delta)$. For this purpose, we use the Delayed Rejection Adaptive Metropolis
algorithm (DRAM, [8]), which is an adaptive variant of the MH algorithm.

As a test case, we use a simplified version of the one dimensional deblurring problem
above, where the dimension is low enough for the DRAM sampler to converge reasonably
fast. The true image and data are given on the left in Figure 3; note that the true signal
is positive everywhere so that the non-negativity constraints do not play a role. We take
1000 samples using our bootstrap method and 100000 samples using DRAM. In both
cases, the median and 95% credibility bounds are computed, and they are compared in
Figure 3, showing that the results obtained with the two samplers are very similar. Since
we know that DRAM correctly samples from $p(x|b, \delta)$, this suggests that the bootstrap
sample density well-approximates $p(x|b, \delta)$ in this case.

Finally, we note that a single MH sample only requires the evaluation of $p(x|b, \delta)$ and
the computation of a sample from the proposal. In contrast, a single sample obtained
using our bootstrap method will typically be much more computationally expensive,
since the optimization algorithm will require many function, gradient, and even Hessian
evaluations. Moreover, we have to assume that the problem is well-posed enough for
the optimizer to numerically work; it is in this case. On the other hand, the bootstrap
method yields independent sample, while for MH an independent sample requires many
samples from the proposal. We might claim that traditional MH type samplers with a good proposal will be more effective for relatively small dimensional, nonlinear problems, but as the dimension of the parameter space increases and good MH proposals get harder, or even impossible, to come by, our parametric bootstrap method may become preferable.

3.2 Two-Dimensional Image deblurring

In the two dimensional delburring case, the model has the form

\[ b(s, t) = \int_0^1 \int_0^1 a(s - s', t - t')x(s', t')ds' dt'. \]

For our tests, we choose a Gaussian convolution kernel \( a \), and discretize using mid-point quadrature on an \( 128 \times 128 \) uniform computational grid over \([0,1] \times [0,1] \). Periodic boundary conditions for the image are assumed, so that \( A \) is an \( 128^2 \times 128^2 \) block circulant with circulant blocks matrix, and hence is diagonalizable by the two-dimensional discrete Fourier transform (DFT) \([12]\). Finally, the data \( b \) is generated using (3) with background \( g = 10 \cdot 1 \). The scale of the true image was chosen so that the signal-to-noise ratio (SNR) is approximately 21. The true image and data are plotted in Figure 4.

We compute 1000 bootstrap samples. The pixel-wise mean of the bootstrap samples for \( x \) is plotted on the upper-left in Figure 5, and the pixel-wise standard deviation is plotted in the upper-right. A histogram for the bootstrap samples for \( \delta \) is plotted on the lower-left, and finally, in the lower-right, is a plot of the MAP solution for \( \delta \) taken to be
3.3 Positron Emission Tomography

In positron emission tomography (PET), a radioactive tracer element is injected into the body, which exhibits radioactive decay, resulting in photon emission. The emitted photon pairs that reach the detector are recorded, as is the line \( L(\omega, y) \) along which they traveled. Here \( \omega \) corresponds to the angle the line \( L(\omega, y) \) makes with the horizontal axis, and \( y \) corresponds to the minimum distance from the origin (usually the center of the spatial domain) to \( L(\omega, y) \). PET data \( b(\omega, y) \) is obtained from the photon counts corresponding to each line \( L(\omega, y) \). After some processing of the data, the model relating the tracer density \( x \) to \( b \) has Fredholm form (1). Specifically, if we parameterize each line as \( L(\omega, y) = \{ z(s) \mid 0 \leq s \leq S \} \), we have (see [9] for details)

\[
b(\omega, y) = \int_{L(\omega, y)} x(z(s)) ds.
\]  

The integral on the right is the Radon transform.

After discretization, (17) can be written as a system of linear equations of the form \( b = Ax \). The discretization occurs both in the spatial domain, where \( x \) is defined, as well as in the Radon transform \( ((\omega, y)) \) domain, where the data \( b \) is defined. We use a uniform \( n \times n \) spatial grid, and a uniform grid for the transform domain with \( n \) angles and \( n \) detectors. In our experiments, \( n = 100 \) so that \( A \) has size \( 10000 \times 10000 \).

Since the data \( b \) consists of photon counts, a Poisson noise model of the form (3) is used [10]. We use the Shepp-Logan phantom generated using MATLAB’s `phantom`
Figure 6. The upper-left plot is of the true image. The upper-right plot is of the pixel-wise mean of the bootstrap samples. On the lower-left is the pixel-wise standard deviation. On the lower-right is a histogram for the bootstrap samples of $\delta$.

function for our true tracer density. We take $\mu = 0$, which is standard for PET numerical experiments [10]. The true tracer density $x$ is then scaled so that SNR=28. Both the data, which was generated using (3) with $g = 10 \cdot 1$, and the true tracer density in the PET case are shown in the upper-left in Figure 6.

We compute 1000 bootstrap samples. The pixel-wise mean of the bootstrap samples for $x$ is plotted on the upper-right in Figure 5, and the pixel-wise standard deviation is plotted in the lower-left. Finally, a histogram of the bootstrap samples of $\delta$ is plotted on the lower-right.

4. Conclusions

We have presented a parametric bootstrap sampling scheme for assessing uncertainty in inverse problems with Poisson distributed data. The approach is motivated from the connection between posterior sampling in the Gaussian case and bootstrap sampling of both the observation and prior equations. We focus on large-scale inverse problems that require regularization, and assume that the prior is a probability model for the unknown that incorporates both prior knowledge and uncertainty. When bootstrapping is implemented in such cases, it therefore makes intuitive sense to resample from both the prior and the observation equations, which is what we do. This deviates from the classical bootstrap where only the observations are resampled. In the case that $\delta$ also needs to be estimated, we assume a probability model for the regularization parameter $\delta$ and estimate $\delta$ using a coordinate descent method.

Our bootstrap method has the benefit that most of the computational effort is performed by the optimization scheme, and hence for large-scale cases, such as the two-
dimensional imaging examples considered here, it will be both easier to implement and more efficient than a Markov chain Monte Carlo (MCMC) method. The down side is that the bootstrap sample density is not the posterior density. More work needs to be done to determine what the bootstrap density is. Nonetheless, we do show in a single example that the bootstrap density well-approximates the posterior.

With the bootstrap samples in hand, the reconstructed image is taken to be the mean of the $x$ samples, and uncertainty is quantified via the pixel-wise standard deviation image. Moreover, the samples of the regularization parameter $\delta$ yield a sample density, from which statistics can be computed. Numerical experiments are performed on three synthetic examples with Poisson noise: image deblurring in one and two dimensions, and positron emission tomography. In all cases, the method works well, and samples can be computed in minutes on a laptop computer. Thus for large-scale examples in which a convergent MCMC method is not available, and the data is Poisson distributed, our bootstrap method seems a viable alternative.

Acknowledgements

This work was supported by a U.S. Department of Energy Nevada National Security Site Directed Research Grant LO-3-13, by the Academy of Finland (134937, and the Finnish Center of Excellence in Inverse Problems, 250215), as well as by funding from Lappeenranta University of Technology, Lappeenranta, Finland.

References