Bayesian Statistical Inference for Black Box Models

Dr Christopher Drovandi

School of Mathematical Sciences
ARC Centre of Excellence for Mathematical and Statistical Frontiers (ACEMS)
Queensland University of Technology

Key ACEMS Members: CI Robert Kohn, CI Scott Sisson, CI Tony Pettitt, CI Kevin Burrage, AI Matias Quiroz, AI David Gunawan, AI Minh Ngoc Tran, AI Brodie Lawson, AI Pamela Burrage, David Nott + Others + Students/Post-docs

Bayesian Statistics

In Bayesian statistics we are interested in sampling from the posterior:

$$p(\theta|y) \propto p(y|\theta)p(\theta),$$

where $p(y|\theta)$ is the likelihood and $p(\theta)$ is the prior.

But how do we 'obtain' the posterior?

Bayesian Inference for Black Box Models

When the model has some known structure, eg hierarchical, it is often possible to construct a Bayesian algorithm that takes advantage of the structure such as component-wise MCMC (eg Gibbs sampling).

If the derivatives of $\log p(\theta|y)$ can be easily obtained then sophisticated sampling methods such as Hamiltonian Monte Carlo can be applied.

But... What if likelihood is a complex piece of computer code, ie black box model?

Intractable Models

Different methods for different levels of intractability:

- Type I: Likelihood is available but moderately expensive to compute.
- Type II: Likelihood is available but very expensive to compute.
- Type III: Likelihood is intractable but can be estimated stochastically in a reasonable time.
- Type IV: Likelihood is intractable but model simulation is cheap.
- Type V: Likelihood is intractable and model simulation is expensive.

An additional complexity is when the posterior distribution is an irregular distribution.

Type I and III Intractability

Firstly we consider cases where likelihood is moderately expensive or likelihood can be estimated unbiasedly and this is also moderately expensive.

Markov Chain Monte Carlo

Construct ergodic Markov chain with invariant distribution $p(\theta|y)$

A common MCMC algorithm is Metropolis Hastings (MH) MCMC, where proposals θ^* are accepted with probability

$$min\left(1, \frac{p(y|\theta^*)p(\theta^*)q(\theta|\theta^*)}{p(y|\theta)p(\theta)q(\theta^*|\theta)}\right),$$

where $q(\cdot)$ is the proposal density.

Markov Chain Monte Carlo

Some Limitations:

- Difficult to automate and adapt the method
- Need to tune the proposal distribution for good performance.
- Convergence can be difficult to assess
- Can have difficulty exploring irregular posteriors (eg multi-modality)
- Standard MCMC is a serial algorithm

Sequential Monte Carlo

SMC methods can be a useful alternative to MCMC in some applications (particularly for expensive black box models)

Basic idea:

- Moving a population of N particles through a sequence of distributions (starting with one easy to sample from and finishing at the target posterior)
- Can introduce the effect of either the data (data annealing) or the likelihood (likelihood annealing) sequentially

Sequential Monte Carlo

In likelihood annealing, the power posteriors are defined by

$$p_t(\theta|y) \propto p(y|\theta)^{\gamma_t} p(\boldsymbol{\theta}),$$

where $0 = \gamma_0 < \gamma_t < \gamma_T = 1$ and $0 < t < 1$.

At each iteration, the following steps are applied

- reweighting
- resampling
- moving to avoid particle degeneration, for example by several runs of an MCMC kernel. We can make use of population of particles.

Advantages of SMC

Advantages

- Naturally adaptive
- Easily parallelisable
- More capable of dealing with multimodal or complex posterior distributions

Challenges for SMC Methods

- Better harnessing population of particles for efficient move step.
- Recycling of all particles/samples/proposals generated through whole algorithm.
- Can be inefficient when there is a big difference between prior and posterior.
- Take advantage of huge literature on advanced MCMC methods and make them work better in SMC.

Type IV Intractability

Now move onto case where likelihood function is essentially intractable but model simulation is cheap.

Likelihood-Free Methods

Here we are interested in models where the likelihood is intractable, but simulation of data x from the model is feasible.

Likelihood-free methods simulate data and compare x with y based on some data summary $S(\cdot)$.

Bayesian methods target $p(\theta|s_y) \propto p(s_y|\theta)p(\theta)$ where $s_y = S(y)$.

Approximate Bayesian Computation

Approximate Bayesian computation (ABC) is current state-of-the-art likelihood-free Bayesian method.

Approximates $p(s_y|\theta)$ via simulation. Simulate n independent datasets $s_{1:n} = (s_1, ..., s_n)$ and use non-parametric kernel density estimation (KDE) to estimate density $p(s_y|\theta)$.

Feed 'approximate' likelihood into MCMC or SMC algorithms. SMC algorithms particularly effective, sequence of distributions formed by reducing bandwidth parameter of KDE.

Likelihood-Free Challenges

Challenges with likelihood-free methods

- Requires significant tuning.
- Scales poorly with dimension of summary statistic.
- Doesn't handle large number of parameters very well.

Various methods have been developed to address challenges but they still remain...

Type II and V Intractability

Likelihood function is available but very expense, or likelihood is completely intractable and model simulation is very expensive.

Emulation

An emulator is a statistic model of a (typically scalar) function which can have several inputs.

The statistical model (eg Gaussian process) can be estimated from a small number of function evaluations.

In Bayesian applications the function might be:

- Log-likelihood function
- One or more outputs of simulation model
- Discrepancy function in ABC.

One the emulator is fitted to training sample it can be used to predict quickly the function at untested locations. Also provides uncertainty quantification of prediction.

Emulation

Step 1: Obtain initial training sample (eg space filling design).

Step 2: Fit emulator to training sample.

Step 3: Use the emulator to guide where to take more training samples.

Step 4: Once happy with emulator use it in inference. Or use it to discard large portion of useless parameter space.

Emulation Challenges

Challenges for Emulation approaches:

- Prediction cost scales poorly with number of training samples (hence number of parameters).
- Makes strong assumption on the underlying function (eg smoothness).

The End

Email: c.drovandi@qut.edu.au

Publication Page:
https://eprints.qut.edu.au/view/person/Drovandi,_Christopher.html

Twitter: @chris_drovandi