Bayesian computation

Eric Feigelson

2nd East Asian Workshops on Astrostatistics
Nanjing & Guiyang
July 2018

Outline

- Introduction
- Stochastic processes & Markov chains
- MCMC algorithms:
  - Properties
  - Variety
  - Visualization
  - Convergence & stopping rules
  - Convergence diagnostic graphics

R/Stan application: Supernova Ia cosmology

Sec 10.11 & code 10.26 in *Bayesian Models for Astrophysical Data Using R, JAGS Python, and Stan* by Hilbe, de Souza & Ishida (2017)

See file HdSI_SNIa.pdf

Introduction to Bayesian computing

While the concepts of inverse probability and Bayesian inference were introduced Simon Pierre Laplace two centuries ago, prior to ~1990, Bayesian inferential applications were largely restricted to simple problems.

Bayesian estimation requires considerable more computation than least squares estimation (system of linear equations) or maximum likelihood estimation (optimization of a single likelihood function) because it often requires examination of, and sometimes integration over, the full space of possible models. Modern astrophysical models can have dozens (or more) parameters, requiring mapping of the prior-weighted likelihood function in high dimensions.

Markov chain Monte Carlo (MCMC) methods can, with varying degrees of efficiency, map the posterior draw sequential samples from the parameter space where the likelihood and prior are evaluated. For simpler problems, the Laplacian approximation can be much more efficient. Integrated Nested Laplacian Approximation (INLA) can be effective for many situations in astronomy (arxiv:1802.06280).

Stochastic processes I

Consider measurements of the Doppler motion of a star orbited by a companion star or exoplanet. A collection of Doppler velocities is a function of *t* is an example of a *stochastic process*: for each observed time *t, X(t)* is a random variable. t can represent any fixed variable: e.g. time, space (1D, 3D, …), space-time, or a lattice parameter space. Astronomers encounter them as functions of fixed time-like variables such as:

Brightness B(RA,Dec) defines an image

Brightness B(wavelength) defines a spectrum

Brightness B(time) defines a light curve

Spectral index or radial velocity (RA,Dec) defines other images

Density $\rho(x,y,z,v_x,v_y,v_z)$ defines a fluid flow

Stochastic processes II

Random variables can be functions of discrete or continuous time-like variables (e.g. pixelated images or lightcurve with accurate timestamps)

The r.v.'s themselves can assume discrete or continuous values (e.g. photon arrivals or real-valued brightness).

The observations can be sequences of i.i.d. r.v.'s, or they can exhibit dependencies. These dependencies can arise either from the instrument (e.g. point spread function in an image) or from the underlying physical process (e.g. timescale for brightness variations in an accretion disk).

A stochastic process is ***stationary*** if $(X(t_1), X(t_2), \ldots X(t_k))$ and $(X(t_1+\delta), X(t_2+\delta), \ldots X(t_k+\delta))$ have the same joint distribution for all δ, $t_1, \ldots, t_k$ and $k \geq 1$.

Note that trends in the mean cause nonstationarity. The spatial structure of an image with stars and galaxies is nonstationary. The brightness variations of a variable star may or may not be stationary.

Markov chains

A stochastic process $\{X_n\}$ is called a *discrete time Markov chain* if the distribution of X_{n+1} given the past $X_n, \ldots, X_0$ depends only on the immediate past. Further suppose that the probability for transitions from states *i* to *j* are fixed, P_{ij}. Formally, this process is written:

$$P\{X_{n+1} = j \mid X_n=i, X_{n-1}=i_{n-1}, \ldots X_0=i_0\} = P(X_{n+1}=j|X_n=i) = P_{ij}$$

For all states $i=i_0$ to i_{n-1} and all times $n \geq 0$

This process is called a *Markov chain.* The values of X_n are called *states*. They need not be integers.

A simple Markov chain is the *random walk*, $P_{i,i+1} = p = 1-P_{i,i-1}$.

Markov Chains II

$\mathbf{P}_{ij}$ can be written as a matrix of one-step transition conditional probabilities from *i* to *j*. An initial state at time 0 needs to be specified to give unconditional probability distributions at time *n*. Note that state *i* may communicate with some states *j* but not with some other states *k*. States that communicate are in the same *class*

Constructing Markov Chains with random numbers to generate i.i.d. sequences of values with a specified p.d.f. provide a suite of algorithms for simulating complicated probability distributions. These are called ***Markov Chain Monte Carlo*** techniques.

Sometimes a Markov process is not directly visible, but some outcome from the chain (e.g. a signal when it visits state *i*) is available. These ***hidden Markov models*** are valuable for a variety of inference problems.

Markov chain Monte Carlo techniques

- **Gibbs sampler** Here the multivariate problem is simplified to a sequence of univariate function evaluations. Consider a 3-dimensional parameter space (θ_1, θ_2,θ_3). Starting at an initial location θ^0, make a random step, simulate $\theta^1{}_1$ | ($\theta^0{}_2$, $\theta^0{}_3$,**X**), $\theta^1{}_2$ | ($\theta^1{}_1$, $\theta^0{}_3$,**X)** and $\theta^1{}_3$ | ($\theta^1{}_1$, $\theta^1{}_2$,**X**), and calculate the posterior (prior-weighted likelihood) at the new location. Continue similar iterations to form a chain, and create multiple chains with different starting points and random steps. For high-dimensional problems, the sequence of variable updates can be varied, and the space can be blocked into subspaces that are updated sequentially.

- **Metropolis-Hastings algorithm** This procedure increases the efficiency of the chain's mapping of the posterior distribution by accepting the next step forward if it increases the posterior or satisfies some probability rule. Strategies for jumping around the parameter space avoid being trapped in small regions of the distribution. An early and common procedure is to combine the Gibbs and Metropolis strategies

Metropolis, Rosenbluth, Teller 1953, "Equation of State Calculations by Fast Computing Machines." *J. Chem. Phys.* MANIAC I computer

Convergence measures and stopping rules for MCMC simulations are very important. Unlike the EM Algorithm for MLE, *there are no theorems guaranteeing convergence* on a maximum in the posterior distribution. Millions of iterations may be needed for a single MCMC chain, and many chains may to needed to obtain reliable results.

Common stopping criteria include:

- chain standard deviation becomes small
- autocorrelation within the chains becomes small
- within-chain and between-chain variances approach equality (Gelman-Rubin diagnostic)

Dozens of MCMC-type methods have been developed in the past ~20 years, and many are implemented in ~100 R/CRAN packages.

Algorithms

MCMC algorithms in the LaplacesDemon CRAN package:

- Adaptive Directional Metropolis-within-Gibbs (ADMG)
- Adaptive Griddy-Gibbs (AGG)
- Adaptive Hamiltonian Monte Carlo (AHMC)
- Adaptive Metropolis (AM)
- Adaptive Metropolis-within-Gibbs (AMWG)
- Adaptive-Mixture Metropolis (AMM)
- Affine-Invariant Ensemble Sampler (AIES)
- Automated Factor Slice Sampler (AFSS)
- Componentwise Hit-And-Run Metropolis (CHARM)
- Delayed Rejection Adaptive Metropolis (DRAM)
- Delayed Rejection Metropolis (DRM)
- Differential Evolution Markov Chain (DEMC)
- Elliptical Slice Sampler (ESS)
- Gibbs Sampler (Gibbs)
- Griddy-Gibbs (GG)
- Hamiltonian Monte Carlo (HMC)
- Hamiltonian Monte Carlo with Dual-Averaging (HMCDA)
- Hit-And-Run Metropolis (HARM)
- Independence Metropolis (IM)
- Interchain Adaptation (INCA)
- … … …

Two variants with code developed by astrostatisticians have acquired sudden popularity in astronomy:

- **MultiNest** This method was designed for complex posterior distributions with many modes (peaks) or degeneracies in high dimensions by Feroz & Hobson (Mon Not R Astro Soc 2008-09). A clustered nested sampling procedure reduces the computations for calculating the Bayesian evidence and posteriors. Written in Fortran 90, it has wrappers for C, C++, R, Python and Matlab.

- **Affine-Invariant Ensemble Sampler** This method was designed for badly scaled parameter spaces and skewed posterior distributions by Goodman & Weare (Comm Appl Math Comp Sci 2010). Here $>2K$ chains are simultaneous run, each with k starting points. For each iteration, the walkers are assigned new positions based on a scaled distance to another randomly-selected chain. Foreman-Mackey, Hogg, Lang, Goodman (Pub Astro Soc Pacific 2013) introduced a public-domain parallelized Python implementation called *emcee* with enthusiastic usage by astronomers.

Descriptions of dozens of MCMC algorithms

https://web.archive.org/web/20150531112558/http://www.bayesian-inference.com:80/mcmc

Interactive visualization of several MCMC algorithms

https://chi-feng.github.io/mcmc-demo/app.html

Characteristics of MCMC algorithms

Chain properties: Non-Markovian (e.g. adaptive, new values not just based on last value), recurrent (retuens to a chosen state), periodic (cyclical);y recurrent, irreducible (all states accessible). A Markov chain with a stationary, aperiodic, irreducible distribution is called *ergodic* with advantageous properties (central limit theorem, convergence).

Proposal generation: multivariate proposal with all parameters, or proposal for individual parameters (slower)

Acceptance rate: ratio of accepted proposals to total iterations

Blockwise sampling: Model parameters are divided into groups of correlated variables that are sampled separately. Allows higher acceptance rates, and tuning algorithms for different blocks. However convergence may be inhibited by inter-block correlation.

Highest posterior density intervals

Metropolis-Hastings algorithm

Consider a function *y* of a time-like series $x^{(t)}$. We want to construct a sequence of y values that sample a target distribution. We start with a 'proposal' distribution *q* that is simpler, and wider, than the (often unknown) target distribution. At each iteration *t* of the chain, perform two operations:

Sample $y \sim q\left(y|x^{(t)}\right)$ with probability $\alpha(x^{(t)}, y) = \min\left\{1, \frac{\pi(y)q\left(x^{(t)}|y\right)}{\pi\left(x^{(t)}\right)q\left(y|x^{(t)}\right)}\right\}$

If accepted, assign *y* to be $x^{(t+1)}$. If reject, do nothing and try again.

If q(y|x)=π(y), then the samples are independent
If q(y|x_ = q(y), we have the independence sampler
If q(y|x) = q(|y-x|), the we have a Metropolis random-walk

Convergence measures and stopping rules for MCMC simulations are very important. Unlike the EM Algorithm, there are no theorems guaranteeing convergence on a maximum in the posterior distribution. Millions of iterations may be needed for a single MCMC chain, and many chains may to needed to obtain reliable results.

Common stopping criteria include:

- chain standard deviation becomes small
- autocorrelation within the chains becomes small
- within-chain and between-chain variances approach equality (Gelman-Rubin diagnostic)

Dozens of MCMC-type methods have been developed in the past ~20 years, and many are implemented in ~100 R/CRAN packages. Two variants developed by astrostatisticians have acquired sudden popularity in astronomy:

Stopping rules and convergence diagnostics

There is no theorem to establish
when a Markov chain has converged.
All convergence diagnostics and stopping rules
are suggestive only.

A simple measure of convergence is when the MCMC reaches a user-specified scatter level. However, due to the autocorrelation, the number of iterations N overestimates the effective sample size. There are various suggested corrections to the standard deviation involving the correlation coefficient or ACF:

$$s = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \overline{x})^2},$$

$$\mathrm{E}\left[s^2\right] = \sigma^2\left[1 - \frac{2}{n-1}\sum_{k=1}^{n-1}\left(1 - \frac{k}{n}\right)\rho_k\right]$$

$$f = \sqrt{\frac{1+\rho}{1-\rho}},$$

$$\mathrm{ESS} = \frac{n}{1 + 2\sum_{k=1}^{\infty}\rho(k)}$$

$$\mathrm{nse}\left(\bar{h}_N\right) \approx \sqrt{\frac{\sigma_h^2}{N}\left\{1 + 2\sum_{l=1}^{N-1}\rho_l(h)\right\}}$$

Geyer 1992

Gelman-Rubin diagnostic

Names: Potential scale reduction factor, G-R diagnostic, G-R shrink factor, R-hat

Concept: Convergence is reasonably achieved when chains have 'forgotten' starting values and recent outputs of different chains are indistinguishable. The variance of the chain ensemble is the sum of within-chain and between-chain variances for n iterations/chain. If the chains have not converged, **W** (mean of the variances within each chain) will be too small and **B** too large. The initial values for the chains must be overdispersed compared to the final posterior distribution (including possible multiple modes).

$$\hat{\sigma}^2 = \frac{(n-1)W}{n} + \frac{B}{n}$$

chain ensemble variance:

R diagnostic: Convergence is reasonable achieved when $1.0 < R \leq 1.1$ where

$$R = \sqrt{\frac{(d+3)\hat{V}}{(d+1)W}}$$

$$d = \frac{2 * \hat{V}^2}{\mathrm{Var}(\hat{V})}$$

$$\hat{V} = \hat{\sigma}^2 + \frac{B}{mn}$$

Gelman, A. & Rubin, D.B. (1992). Inference from iterative simulation using multiple sequences. *Statistical Science*, 7, p. 457–511

Brooks, S.P. & Gelman, A. (1998) General methods for monitoring convergence of interactive simulations*, J. Computational & Graphical Statistics*, 7 434-455

Convergence diagnostics graphics

PDF estimate plot: Univariate or bivariate kernel density estimator plots of posterior from MCMC chains. Note assumptions underlying bandwidth selection. Histograms for discrete valued posteriors.

Trace plots: Time series-like plot of values for each variable in a chain

Autocorrelation function plot: Plot of ACF for each variable in each chain; high autocorrelation shows slow mixing and slow convergence.

Cross-correlatkon plot: Tile plot of correlations between parameters

ROC curve and separation plot: For problems with binary response variable

Caterpillar plot: For high-dimensional problems, stacked boxplot showing HPD & quantiles for each variable

Cumulative quantile plot: Shows evolution of 50%, 99%, … quantiles for n iterations

Gelman-Rubin-Brooks plot: G-R diagnostic vs. n iterations. Important to see recent fluctuations, rather than just test for R~1.0.

Geweke-Brooks plot: Shows Z-score (measuring similarity of beginning & end of a Markov chain) as increasing fractions of the early chain are omitted.

MCMC convergence in R

Diagnostics:

- Gelman and Rubin
- Geweke
- Heidelberger and Welch
- Raftery and Lewis
- Brooks and Gelman multivariate shrink factors

CRAN packages:

- coda
- LaplacesDemon
- ggmcmc
- boa

Many astronomers are not conducting sufficient tests to insure MCMC convergence

Astronomical example of Bayesian computation: Supernova Type Ia cosmology

See R script running Stan code with Hamiltonian MCMC in files:

H-dS-I_SNIa.pdf

H-dS-I_SNIa.R

excerpted from

Bayesian Models for Astrophysical Data using R, Python, JAGS and Stan

by Joseph Hilbe, Rafael de Souza & Emille Ishida (2017)