Bayesian Estimation using MCMC

Kevin M. Esterling (UC – Riverside)

Presentation to GradQuant

October 29, 2015

What is Bayesian inference? And why is it useful?

You are probably already familiar with frequentist inference ...

Philosophy of Frequentist Inference

Probability is a property of the external/natural world

- Probability is an inherent property of a coin or dice or population, etc. but cannot ever be observed
- Inferring this property requires repeated observation; e.g., 1,000 coin flips
- Reporting results is awkward; can't report probability of a heads but instead the probability that an interval will cover this property

What is Bayesian inference? And why is it useful? (cont.)

Philosophy of Bayesian Inference

Bayesian inference posits that probability is best conceived of as a *subjective belief*

- The goal of research is to change **beliefs** about properties of the world; Bayesian analysis is a way to inform your audience how they rationally should change their beliefs after observing data
- If you agree with the subjective view, cool things happen:
  - A single observation can be quite meaningful
  - Reporting results is intuitive; e.g., probability of a heads
  - Optional: we hold subjective beliefs about probabilities/distributions before we observe data, so that information should not be discarded

What is Bayesian inference? And why is it useful? (still cont.)

Practical reasons to be Bayesian, given computational methods:

- Can approximate frequentist results
- But Bayesian methods are much more flexible and/or computationally faster
- Easier to state uncertainty about arbitrary functions of parameters
- Natural way to multiply impute missing data (additional parameters to estimate)

Bayes rule illustration: testing for a disease

You run a test on a patient and you get an ALERT! Here is what we know about the test procedure.

- Incidence of the disease in a population, $p(D) = 0.02$ **("Prior")**
- Probability of the test giving an alert given the presence of the disease, $p(A|D) = 0.95$ **("Likelihood")**
- Probability of the test giving an alert in the absence of the disease (false positive), $p(A|D^c) = 0.03$
  - D = "The patient has the disease"
  - A = "The test gave an alert"

Bayes rule: testing for a disease (cont.)

If test is positive (Alert occurs) use Bayes' Rule:

$$
\begin{aligned}
p(D|A) &= \frac{\textit{prior} \times \textit{likelihood}}{\textit{constant}} \\
&= \frac{p(D)p(A|D)}{p(D)p(A|D)+p(D^c)p(A|D^c)} \\
&= \frac{0.02 \times 0.95}{0.02 \times 0.95 + 0.98 \times 0.03} \qquad = \quad 0.38
\end{aligned}
$$

$\Rightarrow$ subjective probability that patient has the disease, given only one observation (!)

- Patient went from subjective probability of 0.02 to 0.38 of having the disease
- But, just because test is positive does not mean the patient certainly has the disease

Doing Bayesian statistical analysis

- General form of Bayes rule for statistical modeling:
  $p(\theta|y) \propto p(\theta)p(y|\theta)$
  - In words, the posterior density (belief after seeing the data) is proportional to the prior density (belief before seeing the data) times the likelihood of observing the data given those prior beliefs
  - We can drop the normalizing constant that makes the posterior a true probability density
- The old fashioned way was to derive the posterior *analytically*, often using "conjugate" priors
  - A conjugate prior for a likelihood yields a posterior in the same form as the prior
  - Example, conjugate prior for a binomial distribution is the beta distribution; or normal-normal ...

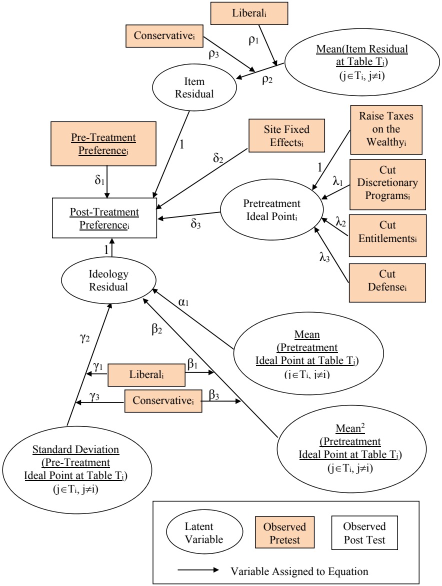
Old fashioned Bayesian statistics

Analytical solution for conjugate normal prior

Let $y_i \overset{iid}{\sim} N(\mu, \sigma^2), i = 1, \ldots, n$, with σ^2 known, and $\mathbf{y} = (y_1, \ldots, y_n)'$. If $\mu \sim N(\mu_0, \sigma_0^2)$ is the prior density for μ, the μ has posterior density,

$$\mu|\mathbf{y} \quad \sim \quad N\left(\frac{\mu_0\sigma_0^{-2} + \bar{y}\frac{n}{\sigma^2}}{\sigma_0^{-2} + \frac{n}{\sigma^2}}, \left(\sigma_0^{-2} + \frac{n}{\sigma^2}\right)^{-1}\right)$$

- Define "Precision" as $\dfrac{1}{\text{variance}} = \tau$
- Results:
    - *Posterior mean* is a precision-weighted average of the prior and data
    - *Posterior precision* is the sum of the prior precision and the data precision
    - Note: Bayesian and MLE converge with diffuse priors and/or lots of data


Liberal_i
Conservative_i
ρ1
ρ3
ρ2
Mean(Item Residual at Table T_i) (j∈T_i, j≠i)
Item Residual
Pre-Treatment Preference_i
Site Fixed Effects_i
Raise Taxes on the Wealthy_i
Cut Discretionary Programs_i
Cut Entitlements_i
Cut Defense_i
δ1
δ2
δ3
1
λ1
λ2
λ3
Post-Treatment Preference_i
Pretreatment Ideal Point_i
Ideology Residual
α1
β2
γ2
γ1
β1
γ3
β3
Liberal_i
Conservative_i
Mean (Pretreatment Ideal Point at Table T_i) (j∈T_i, j≠i)
Mean² (Pretreatment Ideal Point at Table T_i) (j∈T_i, j≠i)
Standard Deviation (Pre-Treatment Ideal Point at Table T_i) (j∈T_i, j≠i)
Latent Variable
Observed Pretest
Observed Post Test
Variable Assigned to Equation

Bayesian computational methods allow you to solve complex models

Recapping from earlier:

- Bayesian methods are much more flexible and/or computationally faster
- Easier to state uncertainty about arbitrary functions of parameters
- Natural way to multiply impute missing data (additional parameters to estimate)

and remember, with diffuse priors and/or large datasets Bayesian methods give similar results as MLE

New-fangled Bayesian statistics: MCMC

"Bayesian estimation using Gibbs sampling" (`WinBUGS`, `OpenBUGS`, `JAGS`, `Stan`, etc.)

- Gibbs sampling: sample an estimate from a candidate posterior distribution for each parameter, conditional on the current estimate of all other parameters
- MCMC = "Markov Chain Monte Carlo" = run the Gibbs sampler repeatedly until the parameters estimates converge to the posterior distribution (proof: stochastic process must converge under minimal conditions if the model is identified, so can start process from an arbitrary point in the posterior space)

New-fangled Bayesian statistics: MCMC (cont.)

Result is a simulated posterior distribution: computational approximation of the posterior

- The vector of draws post-convergence for each parameter is the marginal posterior distribution
- Use tools to assess convergence and conduct analysis/graphing
- For example, trivial to create sampling distributions of functions of parameters

New-fangled Bayesian statistics: MCMC (still cont.)

MCMC procedure

1. Specify model (likelihood and priors) with `WinBUGS` code
2. Load data and compile model
3. Provide initial values for parameters, latent variables, missing data
4. Run model for an initial "burn-in" period until MCMC converges on the posterior distribution
5. Save a sample of draws for parameters of interest
6. Summarize marginal distributions, plots, statistical tests

Model of the mean

Likelihood:

$$\left.\text{mass}_i \sim \phi(\mu, \tau)\ \right\} 1 \le i \le \text{n.obs} \qquad \text{IID assumption}$$

Priors:

$$\begin{aligned}
\mu &\sim \text{dunif}(0, 5000) && \text{Flat postive prior (“informative” prior!)} \\
\tau &= \frac{1}{\sigma^2} && \text{Precision is inverse of variance} \\
\sigma^2 &= \sigma \times \sigma && \text{Define variance in model} \\
\sigma &\sim \text{dunif}(0, 100) && \text{Flat positive prior}
\end{aligned}$$

Let's run this model in `OpenBUGS` **. . .**

Model of the mean

Likelihood:

$$\text{mass}_i \sim \phi(\mu, \tau) \quad \Big\} \, 1 \leq i \leq \text{n.obs}$$ IID assumption

Priors:

$\mu \sim \text{dunif}(0, 5000)$ Flat postive prior ("informative" prior!)

$\tau = \frac{1}{\sigma^2}$ Precision is inverse of variance

$\sigma^2 = \sigma \times \sigma$ Define variance in model

$\sigma \sim \text{dunif}(0, 100)$ Flat positive prior

Let's run this model in `OpenBUGS` **...**

Model of the mean

Likelihood:

$$\left.\text{mass}_i \sim \phi(\mu, \tau)\right\} 1 \le i \le \text{n.obs} \qquad \text{IID assumption}$$

Priors:

$\mu \sim \text{dunif}(0, 5000)$ Flat postive prior ("informative" prior!)

$\tau = \frac{1}{\sigma^2}$ Precision is inverse of variance

$\sigma^2 = \sigma \times \sigma$ Define variance in model

$\sigma \sim \text{dunif}(0, 100)$ Flat positive prior

Let's run this model in `OpenBUGS` **...**

Model of the mean

Likelihood:

$$\text{mass}_i \sim \phi(\mu, \tau) \quad \Big\} \, 1 \leq i \leq \text{n.obs} \qquad \text{IID assumption}$$

Priors:

$\mu \sim \text{dunif}(0, 5000)$ Flat postive prior ("informative" prior!)

$\tau = \frac{1}{\sigma^2}$ Precision is inverse of variance

$\sigma^2 = \sigma \times \sigma$ Define variance in model

$\sigma \sim \text{dunif}(0, 100)$ Flat positive prior

Let's run this model in `OpenBUGS` **...**

Model of the mean

Likelihood:

$$\text{mass}_i \sim \phi(\mu,\tau) \quad \Big\} 1 \leq i \leq \text{n.obs}$$ IID assumption

Priors:

$\mu \sim \text{dunif}(0, 5000)$ Flat postive prior ("informative" prior!)

$\tau = \frac{1}{\sigma^2}$ Precision is inverse of variance

$\sigma^2 = \sigma \times \sigma$ Define variance in model

$\sigma \sim \text{dunif}(0, 100)$ Flat positive prior

Let's run this model in `OpenBUGS` **...**

Model of the mean

Likelihood:

$$\text{mass}_i \sim \phi(\mu, \tau) \quad \Big\} \, 1 \leq i \leq \text{n.obs} \qquad \text{IID assumption}$$

Priors:

$$
\begin{array}{lll}
\mu & \sim & \text{dunif}(0, 5000) \qquad \text{Flat postive prior (“informative” prior!)} \\
\tau & = & \frac{1}{\sigma^2} \qquad \text{Precision is inverse of variance} \\
\sigma^2 & = & \sigma \times \sigma \qquad \text{Define variance in model} \\
\sigma & \sim & \text{dunif}(0, 100) \qquad \text{Flat positive prior}
\end{array}
$$

Let's run this model in `OpenBUGS` **...**

Model of the mean

Likelihood:

$$\text{mass}_i \sim \phi(\mu, \tau) \quad \Big\} 1 \leq i \leq \text{n.obs} \qquad \text{IID assumption}$$

Priors:

$$\begin{array}{lll}
\mu & \sim & \text{dunif}(0, 5000) \qquad \text{Flat postive prior (“informative” prior!)} \\
\tau & = & \frac{1}{\sigma^2} \qquad \text{Precision is inverse of variance} \\
\sigma^2 & = & \sigma \times \sigma \qquad \text{Define variance in model} \\
\sigma & \sim & \text{dunif}(0, 100) \qquad \text{Flat positive prior}
\end{array}$$

Let's run this model in `OpenBUGS` . . .

Common problems and some advice

- Always run multiple chains (usually three) in order to test convergence using BGR diagnostic
  - BGR diagnostic assesses within-to-between chain variance (assumes overdispersed/random initial values)
  - Consider both mathematical and empirical identification (just because you can write it down does not mean you should estimate it)
  - Best to start with simple model and build up complexity
- Assess burnin period, mixing carefully
- Be sure there are no missing data on RHS
- Read the manual; and Gelman and Hill (2006) is a great resource for multilevel modeling
- Learn scripting language

Using OpenBUGS with R

In practice, you want to store your data and analyze/graph results within R (or Stata or SAS etc.)

- Once you know how to use OpenBUGS you can read documentation to these R packages:
    - R2OpenBUGS, BRugs = Interact with OpenBUGS within R
    - CODA = Suite of tools to assess convergence and describe results
    - BRugs installs/loads all three
- Prepare data and inits text files to read directly into OpenBUGS
- Read OpenBUGS output as MCMC object for use in CODA and presenting results
- Call OpenBUGS from R for automating Bayesian analysis