MSA101/MVE187 2021 Lecture 7
Examples
Gibbs sampling
Hierarchical models
Slice sampling
Tips and tricks

Petter Mostad

Chalmers University

September 20, 2021

Review: The Metropolis-Hastings algorithm

Given a probability density f that we want to simulate from. Construct a *proposal function* $q(y \mid x)$ which for every x gives a probability density for a proposed new value y. The algorithm starts with a choice of an initial value $x^{(0)}$ for x, and then simulates each $x^{(t)}$ based on $x^{(t-1)}$. Specifically, given $x^{(t)}$,

- Simulate a new value y according to $q(y \mid x^{(t)})$.
- Compute the acceptance probability

$$\rho(x^{(t)}, y) = \min\left(\frac{f(y)q(x^{(t)} \mid y)}{f(x^{(t)})q(y \mid x^{(t)})}, 1\right).$$

- Set

$$x^{(t+1)} = \begin{cases} y & \text{with probability } \rho(x^{(t)}, y) \\ x^{(t)} & \text{with probability } 1 - \rho(x^{(t)}, y) \end{cases}$$

- We looked at random-walk proposals and independent proposals, and how to choose them. Some technical issues.

Overview of today

- A heart transplant example from Albert
- Using several different proposal functions
- Gibbs sampling
- More technical stuff and advice

Heart transplant example from Albert (chapter 7)

- For 94 hospitals that do heart transplant surgery, learn about the mortality rate λ_i at hospital i, $i = 1, \ldots, 94$.
- **Presentation break for illustration in R**
- A possible question: At a new exposure e, what is the chance of dying at hospital i?
- Another possible question: The probability that $\lambda_i < \lambda_j$ for hospitals i, j.
- Model: $y_i \mid \lambda_i \sim \text{Poisson}(e_i \lambda_i)$, but how to model $\lambda_1, \ldots, \lambda_{94}$?
- Three possibilities:
  - Equal: $\lambda = \lambda_1 = \cdots = \lambda_{94}$ drawn from a prior we specify.
  - Independent: $\lambda_1, \ldots, \lambda_{94}$ drawn indepedently from a prior we specify.
  - $\lambda_1, \ldots, \lambda_{94}$ drawn from a joint distribution: We learn about that distribution from data!
- In terms of estimates, we will get below

$$\frac{\sum_{j=1}^{94} y_j}{\sum_{j=1}^{94} e_j} \qquad \text{or} \qquad \frac{y_1}{e_1}, \ldots, \frac{y_{94}}{e_{94}} \qquad \text{or} \qquad w\frac{\sum_{j=1}^{94} y_j}{\sum_{j=1}^{94} e_j} + (1-w)\frac{y_i}{e_i}$$

Assuming equal rates

- If we use the prior $\pi(\lambda) \propto 1/\lambda$ and data from hospital 1 we get

$$\begin{aligned} \pi(\lambda \mid y_1) &\propto_\lambda \pi(y_1 \mid \lambda)\pi(\lambda) \propto_\lambda \text{Poisson}(y_1; e_i\lambda)/\lambda \propto_\lambda e^{e_1\lambda}\lambda^{y_1-1} \\ &\propto_\lambda \text{Gamma}(\lambda; y_1, e_1) \end{aligned}$$

- The posterior after considering all data becomes

$$\text{Gamma}\left(\sum_{j=1}^{94} y_i, \sum_{j_1}^{94} e_i\right) = \text{Gamma}(277, 294681) = \text{Gamma}(S_y, S_e).$$

- Note that the expected value becomes S_y/S_e.
- Computing with the Poisson-Gamma conjugacy, we get that the predictive distribution at new exposure e is

$$\begin{aligned} \pi(y) &= \frac{\text{Poisson}(y; \lambda e)\,\text{Gamma}(\lambda; S_y, S_e)}{\text{Gamma}(\lambda; S_y + h, S_e + e)} \\ &= \text{Negative-Binomial}\left(y; S_y, \frac{S_e}{S_e + e}\right). \end{aligned}$$

- **Presentation break for R computation**

Assuming rates are independent

- If we use the improper prior $\pi(\lambda_i) \propto_{\lambda_i} 1/\lambda_i$, then the posterior becomes improper for the hospitals where no deaths have occurred ($y_i = 0$). Problem!
- For other hospitals we get $\lambda_i \mid \text{data} \sim \text{Gamma}(y_i, e_i)$, with expectation y_i/e_i.
- We can use a proper prior, but where should the information come from to make this prior?
- Most reasonable to pool the information form different hospitals, but acknowledge that the λ_i may be different.

Using a hierarchical model

- We assume the λ_i are sampled from some distribution, AND we try to learn the parameters of this distribution from the data!
- We use the model

$$y_i \mid \lambda_i \sim \text{Poisson}(\lambda_i e_i) \text{ and } \lambda_i \sim \text{Gamma}\left(\alpha, \frac{\alpha}{\mu}\right),$$

$$\pi(\alpha) \propto \frac{1}{\alpha} \text{ and } \pi(\mu) \propto_{\mu} \frac{1}{\mu}$$

- Note: With this parametrization, the expectation of the Gamma distribution is μ and its standard deviation is $\mu/\sqrt{\alpha}$, so this parametrization can be easily interpreted.
- We now have a fully specified Bayesian model with 96 parameters $\mu, \alpha, \lambda_1, \lambda_2, \dots, \lambda_{94}$.
- The posterior distribution on α will tell us to what extent the λ_i are similar.

Computations for model 3

- The model above has $94 + 2$ unobserved variables. For more easy computation, note that the distribution of $y_1, \dots, y_{94}$, α, and μ is equivalent in the following marginalized model:

$$y_i \sim \text{Neg-Binomial}\left(\alpha, \frac{\alpha/\mu}{\alpha/\mu + e_i}\right), \; \pi(\alpha) \propto_\alpha \frac{1}{\alpha} \text{ and } \pi(\mu) \propto_\mu \frac{1}{\mu}$$

- As we now only have 2 unknown variables, we can do inference for μ and α for example with discretization or MCMC.
- If we then want the posterior density for some particular λ_j, note that

$$\lambda_j \mid \alpha, \mu, \text{data} \sim \text{Gamma}\left(\alpha + y_j, \frac{\alpha}{\mu} + e_j\right).$$

- Computations (in R) can now answer questions such as
  - What is the probability of no deaths in hospital 24 given a new exposure of 1000?
  - What is the probability that hospital 90 is really better than hospital 9, i.e., that $\lambda_{90} < \lambda_9$?

Computations in model 3

- For the posterior $\pi(\alpha, \mu \mid \text{data})$

$$\pi(\alpha, \mu \mid \text{data}) \propto_{\alpha,\mu} \frac{1}{\alpha\mu} \prod_{i=1}^{94} \text{Neg-Binomial}\left(y_i; \alpha, \frac{\alpha}{\alpha + \mu e_i}\right)$$
$$\propto_{\alpha,\mu} \frac{1}{\alpha\mu} \prod_{i=1}^{94} \frac{\Gamma(y_i + \alpha)}{\Gamma(\alpha)} \left(\frac{\alpha}{\alpha + \mu e_i}\right)^{\alpha} \left(\frac{\mu e_i}{\alpha + \mu e_i}\right)^{y_i}.$$

- To make the posterior more symmetrical, improve numerical properties, and avoid problems that α and μ can only have positive values, we do the *reparametrization* $\theta_1 = \log(\alpha)$ and $\theta_2 = \log(\mu)$, i.e., $\alpha = e^{\theta_1}$ and $\mu = e^{\theta_2}$.

More on computations for model 3

- With the reparametrization above, the posterior for $\pi(\theta_1, \theta_2 \mid \text{data})$ is proportional to

$$\prod_{i=1}^{94} \frac{\Gamma(y_i + e^{\theta_1})}{\Gamma(e^{\theta_1})} \left(\frac{e^{\theta_1}}{e^{\theta_1} + e^{\theta_2} e_i} \right)^{e^{\theta_1}} \left(\frac{e^{\theta_2} e_i}{e^{\theta_1} + e^{\theta_2} e_i} \right)^{y_i}.$$

- Remember that for numerical reasons we prefer to compute the logged posterior (or use the R function `dnbinom`):

$$\begin{aligned} L(\theta_1, \theta_2) &= \sum_{1=1}^{94} \log(\Gamma(y_i + e^{\theta_1})) - \log(\Gamma(e^{\theta_1})) + \theta_1 e^{\theta_1} + y\theta_2 \log(e_i) \\ & \quad - y_i e^{\theta_1} \log(e^{\theta_1} + e^{\theta_2} e_i) \end{aligned}$$

- When discretizing this in 2D, it's good to know for approximately what values you expect it to be large: Note that $\sum_i y_i / \sum_i e_i \approx 0.001$ so $\mu \approx 0.001$ and $\theta_2 \approx -7$. Furthermore, from our interpretation of α, we see that it is probably greater than 1, so $\theta_1 > 0$. See R computations.
- **Presentation break for computations in R**

Switching between several proposal functions

- We presented the Metropolis Hastings algorithm as using only *one* proposal density.
- Actually
  - you may use a whole menu of propsal functions, and
  - you may switch between them in a systematic or random way,

  as long as the resulting Markov chain in the end becomes ergodic.
- For some "difficult" posterior densities, you might usually use a small-step random walk, but occasionally use a large-step proposal, tailored to jump between separate "islands" of high posterior density.
- A very popular possibility: Using proposal densities that fix all but one (or all but some) of the variables.
- You need to cycle through different proposal functions so that all variables have a chance to be updated.
- When computing the acceptance probability

  $$\rho(x^{(t)}, y) = \min\left(\frac{f(y)q(x^{(t)} \mid y)}{f(x^{(t)})q(y \mid x^{(t)})}, 1\right).$$

  usually many factors cancel, so there are computational advantages.
- In Albert, this is called "Metropolis within Gibbs".

Gibbs sampling

- If $(x_1, x_2, \ldots, x_n)$ is the variable vector, imagine that you cycle through proposal functions $j = 1, \ldots, n$, where proposal j only changes x_j, leaving all other variables unchanged.
- Assume in fact proposal j simulates a new proposed value x_j^* from

$$\pi(x_j \mid x_1, \ldots, x_{j-1}, x_{j+1}, \ldots, x_n),$$

  the conditional distribution of x_j given all the other variables.
- The acceptance probability in the MH algorithm is computed with

$$\begin{aligned} & \frac{\pi(x^*)q(x \mid x^*)}{\pi(x)q(x^* \mid x)} \\ = \quad & \frac{\pi(x_1, \ldots, x_j^*, \ldots, x_n)\pi(x_j \mid x_1, \ldots, x_{j-1}, x_{j+1}, \ldots, x_n)}{\pi(x_1, \ldots, x_j, \ldots, , x_n)\pi(x_j^* \mid x_1, \ldots, x_{j-1}, x_{j+1}, \ldots, x_n)} \\ = \quad & \frac{\pi(x_1, \ldots, x_{j-1}, x_{j+1}, \ldots, x_n)}{\pi(x_1, \ldots, x_{j-1}, x_{j+1}, \ldots, x_n)} = 1 \end{aligned}$$

  So accept always!
- This algorithm is called *Gibbs sampling*.

Gibbs sampling: Examples

- Example: Simulate from a bivariate normal distribution. The conditional distributions are normal, formulas are given in a previous lecture. **Presentation break for example in R**
- Example: Data $y_1, y_2, \ldots, y_n$ are from a Normal(μ, τ^{-1}) distribution, with independent priors $\mu \sim$ Normal$(0, 1)$ and $\tau \sim$ Gamma$(3, 4)$.
  - When τ is fixed we get
$$\mu \mid \tau, \text{data} \sim \text{Normal}\left(\frac{n\bar{y}\tau}{n\tau + 1}, \frac{1}{n\tau + 1}\right).$$
  - When μ is fixed we get
$$\tau \mid \mu, \text{data} \sim \text{Gamma}\left(3 + \frac{n}{2}, 4 + \frac{1}{2}\sum_{i=1}^{n}(y_i - \mu)^2\right).$$
- When τ is fixed, the formula above is a result of the formula for the posterior in the Normal-Normal conjugacy with fixed precision.
- When μ is fixed, the formula above is a result of the formula for the posterior in the Normal-Gamma conjugacy with fixed expectation.
- **Presentation break for R computation**

Gibbs sampling: Summary

- For many models it is easy to implement and program.
- In particular, in hierarchical models Gibbs sampling is sometimes quite easy to find the formulas for (i.e., the conditional densities to simulate from).
- No need to bother with acceptance probabilities!
- However, the convergence may be too slow for practical use if
    - the variables are highly correlated in the posterior, or
    - separate regions of high posterior density cannot easily be reached by changing one variable at a time.
- You may use blocked Gibbs sampling: Updating a subset of the variables sampling from their conditional distribution given the remaining variables.

Hierarchiclal models

- Sometimes, observed data have dependencies that can best be described using a hierarchy.
- The heart transplant data is an example.
- Example: Test results for students may depend on the class they are in, the school they attend, and the country they live in.
- A statistical model for the data should then contain a random variable for each "source of infuence"; they would depend on each other in a hierarchy, which can be drawn as an upside-down tree, or more generally as a network.
- When making computations, the tree structure can be very useful, for example to find conditional distributions for Gibbs sampling.

A hierarchical example

Data $x_1, \ldots, x_8$ and $y_1, \ldots, y_6$ are organized into groups, and we want to predict a value z_1 in a third group. We assume a model

$$\begin{aligned}
x_1, \ldots, x_8 &\sim \text{Normal}(\mu_1, \tau_1^{-1}) \\
y_1, \ldots, y_6 &\sim \text{Normal}(\mu_2, \tau_1^{-1}) \\
z_1 &\sim \text{Normal}(\mu_3, \tau_1^{-1}) \\
\mu_1, \mu_2, \mu_3 &\sim \text{Normal}(10, \tau_0^{-1}) \\
\tau_0 &\sim \text{Gamma}(1, 4) \\
\tau_1 &\sim \text{Gamma}(7, 3)
\end{aligned}$$

- **Presentation break for drawing**
- We can make predictions for z_1 given data $x_1, \ldots, x_8$ and $y_1, \ldots, y_6$ by simulating with Gibbs sampling from the model where the data is fixed and the remaining variables $\mu_1, \mu_2, \mu_3, \tau_0, \tau_1, z_1$ are simulated.
- Note: The exact form for the conditional distributions of each of these variables can be found using conjugacy.

Conditional distributions for the example

The conditional distributions become (prove yourself!)

$$
\begin{aligned}
\mu_1 \mid x_1, \ldots, x_8, \tau_1, \tau_0 &\sim \text{Normal}\left(\frac{10\tau_0 + 8\overline{x}\tau_1}{\tau_0 + 8\tau_1}, \frac{1}{\tau_0 + 8\tau_1}\right) \\
\mu_2 \mid y_1, \ldots, y_6, \tau_1, \tau_0 &\sim \text{Normal}\left(\frac{10\tau_0 + 6\overline{y}\tau_1}{\tau_0 + 6\tau_1}, \frac{1}{\tau_0 + 6\tau_1}\right) \\
\mu_3 \mid z_1, \tau_1, \tau_0 &\sim \text{Normal}\left(\frac{10\tau_0 + z_1\tau_1}{\tau_0 + \tau_1}, \frac{1}{\tau_0 + \tau_1}\right) \\
\tau_0 \mid \mu_1, \mu_2, \mu_3 &\sim \text{Gamma}\left(1 + \frac{3}{2}, 4 + \frac{1}{2}\sum_{i=1}^{3}(\mu_i - 10)^2\right) \\
\tau_1 \mid \mu_1, \mu_2, \mu_3, x_1 \ldots x_8, y_1 \ldots y_6, z_1 &\sim \text{Gamma}\left(7 + \frac{15}{2}, 3 + \frac{1}{2}\sum_{i=1}^{8}(x_i - \mu_1)^2\right. \\
&\left. + \frac{1}{2}\sum_{i=1}^{6}(y_i - \mu_2)^2 + \frac{1}{2}(z_1 - \mu_3)^2\right) \\
z_1 \mid \mu_3, \tau_1 &\sim \text{Normal}(\mu_3, \tau_1^{-1})
\end{aligned}
$$

Presentation break for computations in R

Hierarchical models

- In most hierarchical models, there are (at least some) conditional distributions that do not have nice analytic forms.
- Using the posterior density over all the variables and removing factors that do not involve the variable we want to simulate, we still get a function proportional to its conditional density.
- We may update this variable using another type of Metropolis Hastings proposal (like random walk).
- Note: It may often be better to work with the logged posterior density: Then one may remove additive terms not involving the variable one wants to simulate over.

The slice sampler

- Idea: Do Gibbs sampling from "the area under the density curve". **Presentation break for drawing**
- More formally, given density $f_x(x)$, simulate from the joint density

$$f(x, u) = I(0 < u < f_x(x))$$

- Works even if the density f_x is known only up to a constant.
- The challenge is to simulate x uniformly on $\{x : u < f_x(x)\}$. This is most easily done if for example f_x is a decreasing function, so that it is invertible.
- Example: Simulate from the density $\pi(x) = \frac{1}{2}\exp\left(-\sqrt{x}\right)$. We iterate between the following steps:
  - Given an x value, simulate $u \sim \text{Uniform}\left(0, \frac{1}{2}\exp\left(-\sqrt{x}\right)\right)$.
  - Given a u value simulate $x \sim \text{Uniform}\left(0, (\log(2u))^2\right)$: Note that $u = \frac{1}{2}\exp\left(-\sqrt{x}\right)$ if and only if $x = (\log(2u))^2$ and that $\pi(x)$ is decreasing as a function of x.

Generalization to more dimensions

- The theory can easily be extended to more dimensions: When we want to simulate from the density

  $$f(x) = \prod_{i=1}^{n} g_i(x)$$

  we can define the joint density

  $$h(x, u_1, \dots, u_n) = \prod_{i=1}^{n} I\left(0 < u_i < g_i(x)\right)$$

- We see that the marginal density for x is $f(x)$.
- We simulate from the joint density using Gibbs sampling. This is very easy for the variables $u_1, \dots, u_n$.
- The conditional distribution of x given $u_1, \dots, u_n$ is the uniform distribution on the set

  $$\cap_{i=1}^{n}\{x : u_i < g_i(x)\}.$$

  If it is easy to compute this set, slice sampling works well. One example: If all the $g_i(x)$ functions are decreasing and invertible.

Example: The Challenger disaster

- The goal is to compute the probability that a space shuttle “o-ring” fails at a specific temperature. (An o-ring failing because of cold weather was the cause of the Challenger space shuttle disaster).
- Data $(x_1, y_1), \dots, (x_n, y_n)$ where x_i denotes the temperature (in Farenheit) and y_i is 1 if there is a failure, 0 otherwise. **Presentation break for illustration in R**.
- We use a logistic regression model:

$$y_i \sim \text{Bernoulli}(p(x_i)) \qquad p(x_i) = \frac{\exp(a + bx_i)}{1 + \exp(a + bx_i)}.$$

- The posterior becomes (using flat priors on a and b)

$$\begin{aligned} \pi(a, b \mid \text{data}) &\propto \prod_{i=1}^{n} \left(\frac{\exp(a + bx_i)}{1 + \exp(a + bx_i)} \right)^{y_i} \left(\frac{1}{1 + \exp(a + bx_i)} \right)^{1 - y_i} \\ &= \prod_{i=1}^{n} \frac{\exp(a + bx_i)^{y_i}}{1 + \exp(a + bx_i)} \end{aligned}$$

Example continued

- Simulate from posterior for parameters (a, b) using slice sampling:
  - For $i = 1, \ldots, n$, simulate $u_i \sim \text{Uniform}\left[0, \frac{\exp(a+bx_i)^{y_i}}{1+\exp(a+bx_i)}\right]$.
  - Simulate (a, b) uniformly on set satisfying, for all i, $u_i < \frac{\exp(a+bx_i)^{y_i}}{1+\exp(a+bx_i)}$.
- Corresponds to $a + bx_i > \log(u_i/(1-u_i))$ for i with $y_i = 1$, and $a + bx_i < \log((1-u_i)/u_i)$ for i with $y_i = 0$.
- To simulate (a, b) uniformly on this set, we first simulate a with

$$a \sim \text{Uniform}\left[\max_{y_i=1}\left(\log\frac{u_i}{1-u_i} - bx_i\right), min_{y_i=0}\left(\log\frac{1-u_i}{u_i} - bx_i\right)\right]$$

- Then for b, we need to be more careful, simulating b uniformly in the interval of numbers
  - Greater than $\left(\log\frac{u_i}{1-u_i} - a\right)/x_i$ for i with $y_i = 1$ and $x_i > 0$.
  - Smaller than $\left(\log\frac{u_i}{1-u_i} - a\right)/x_i$ for i with $y_i = 1$ and $x_i < 0$.
  - Smaller than $\left(\log\frac{1-u_i}{u_i} - a\right)/x_i$ for i with $y_i = 0$ and $x_i > 0$.
  - Greater than $\left(\log\frac{1-u_i}{u_i} - a\right)/x_i$ for i with $y_i = 0$ and $x_i < 0$.

Example continued

- This is actually Example 7.11 in RC, but the book contains some errors:
  - Confusion beween (a, b) and (α, β)
  - Second and fourth formulas on page 220 are wrong.
  - No need to use a prior for a and b to get this to work; use centering instead.
- Note that a and b are highly correlated in the posterior if we implement the code directly. Much improved convergence and accuracy is obtained by *centering* the data: Subtracting the average value from the temperature values, performing the analysis, and then adding back the average value.
- **Presentation break for computation in R**.

Summing up some tips and tricks

- Usually a good idea to compute with the logarithm of the posterior, instead of the posterior itself.
- Reparametrize all variables so that they are defined on the real line (if it is possible and convenient).
- Make sure your code avoids underflow and overflow numerical problems. Make sure a function computing (logged) posterior density will always return sensible answers for any values that might be proposed.
- Reparametrize the model, if possible and convenient, so that parameters are as uncorrelated as possible in the posterior. Otherwise, you may try out a random walk with correlated proposals.
- Do a normal approximation if convenient: A mode is nice to know, and the variances, and the covariance matrix, may be helpful for deciding step lengths in your MCMC! (Rule of thumb, two times standard deviation, does not always work).
- If available, use some classical anlaysis to find reasonable starting values for your parameters.
- Vary the starting point of the Markov chain! (Propose from prior?)
- For more complex models, tailored proposals may be necessary!

Checking convergence

- We know the results from MCMC will be correct in the limit when the sample size $\to \infty$.
- Only in very special cases (e.g. using "coupling") do we know how big the sample size needs to be to get a certain accuracy.
- In practice "checking convergence" means checking for signs of non-convergence or slow convergence (slow "mixing"):
    - Monitor variable values and cumulative averages.
    - Check autocorrelations for variables.
    - Check acceptance rates (but higher is not always better, unless you are using independent proposals!)
    - Use multiple starting points for the MCMC chain!
    - Use multiple parallell chains, and compare variace within chains with variance between chains! (Special tests have been developed).
- An important ingredient is to *understand* your model and your posterior, so that you can guess what might cause convergence problems, and check for such problems.