MCMC: Intermediate Examples

version: 2020-03-12 · 11:16:30

Example: Noncentral t-Distribution

Definition: Let $z \sim \mathcal{N}(\mu, \sigma^2) \quad \amalg \quad x \sim \chi^2_{(\nu)}$. Then

$$y = \frac{z}{\sqrt{x/\nu}} + \eta$$

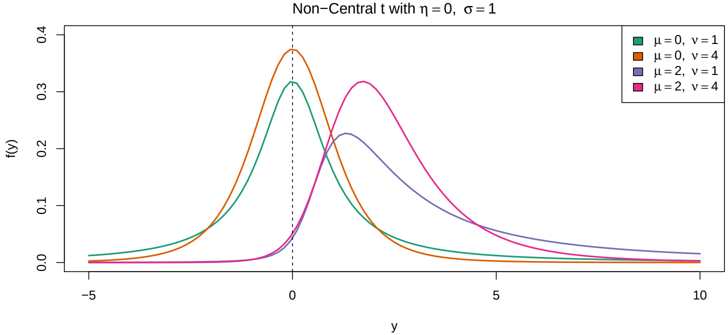
has a Noncentral Student-t distribution, denoted $y \sim t_{(\nu)}(\mu, \sigma, \eta)$.

Noncentral t-Distribution

Definition: Let $z \sim \mathcal{N}(\mu, \sigma^2) \quad \amalg \quad x \sim \chi^2_{(\nu)}$. Then

$$y = \frac{z}{\sqrt{x/\nu}} + \eta \sim t_{(\nu)}(\mu, \sigma, \eta).$$

Modeling: Allows very general specification of mean, variance, skewness and kurtosis.



Parameter Inference

- **Model:** $y_i \overset{\text{iid}}{\sim} t_{(\nu)}(\mu, \sigma, \eta)$
- **Loglikelihood:**

$$\ell(\mu, \sigma, \eta, \nu \mid \boldsymbol{y}) = \sum_{i=1}^{n} \texttt{dt(x = } y_i - \eta)/\sigma\texttt{, df = } \nu\texttt{, ncp = } \mu\texttt{, log = TRUE)} - n \log \sigma.$$

So for this problem we could get away with MLE, or

Approximate Bayesian Inference

1. **Unconstrain Parameters:**

$$\boldsymbol{\theta} = (\mu, \sigma, \eta, \nu) \quad \to \quad \boldsymbol{\psi} = (\mu, \lambda = \log \sigma, \eta, \omega = \log \nu).$$

(Approximation works much better on uncontrained scale.)

Approximate Bayesian Inference

1. **Unconstrain Parameters:**

$$\boldsymbol{\theta} = (\mu, \sigma, \eta, \nu) \quad \rightarrow \quad \boldsymbol{\psi} = (\mu, \lambda = \log \sigma, \eta, \omega = \log \nu).$$

   (Approximation works much better on uncontrained scale.)

2. **Posterior:** $p(\psi \mid \boldsymbol{y}) \propto \mathcal{L}(\psi \mid \boldsymbol{y}) \cdot \pi(\psi)$.

Approximate Bayesian Inference

1. **Unconstrain Parameters:**

$$\boldsymbol{\theta} = (\mu, \sigma, \eta, \nu) \quad \to \quad \boldsymbol{\psi} = (\mu, \lambda = \log \sigma, \eta, \omega = \log \nu).$$

   (Approximation works much better on uncontrained scale.)

2. **Posterior:** $p(\psi \mid \boldsymbol{y}) \propto \mathcal{L}(\psi \mid \boldsymbol{y}) \cdot \pi(\psi)$.

3. **Normal Approximation:** $\psi \mid \boldsymbol{y} \approx \mathcal{N}(\hat{\psi}, \hat{\boldsymbol{V}})$, where

$$\hat{\boldsymbol{\psi}} = \arg\max_{\boldsymbol{\psi}} \log p(\boldsymbol{\psi} \mid \boldsymbol{y}), \qquad \hat{\boldsymbol{V}} = -\left[\frac{\partial^2}{\partial \boldsymbol{\psi}^2} \log p(\hat{\boldsymbol{\psi}} \mid \boldsymbol{y})\right]^{-1}.$$

   (Also called the mode-quadrature approximation.)

Approximate Bayesian Inference

1. **Unconstrain Parameters:**

$$\boldsymbol{\theta} = (\mu, \sigma, \eta, \nu) \quad \to \quad \boldsymbol{\psi} = (\mu, \lambda = \log \sigma, \eta, \omega = \log \nu).$$

   (Approximation works much better on uncontrained scale.)

2. **Posterior:** $p(\psi \mid \boldsymbol{y}) \propto \mathcal{L}(\psi \mid \boldsymbol{y}) \cdot \pi(\psi)$.

3. **Normal Approximation:** $\psi \mid \boldsymbol{y} \approx \mathcal{N}(\hat{\psi}, \hat{\boldsymbol{V}})$, where

$$\hat{\boldsymbol{\psi}} = \arg\max_{\boldsymbol{\psi}} \log p(\boldsymbol{\psi} \mid \boldsymbol{y}), \qquad \hat{\boldsymbol{V}} = -\left[\frac{\partial^2}{\partial \boldsymbol{\psi}^2} \log p(\hat{\boldsymbol{\psi}} \mid \boldsymbol{y})\right]^{-1}.$$

   (Also called the mode-quadrature approximation.)

4. **Monte Carlo Sampling:**

   i. $\boldsymbol{\psi}^{(1)}, \ldots, \boldsymbol{\psi}^{(M)} \overset{\text{iid}}{\sim} \mathcal{N}(\hat{\boldsymbol{\psi}}, \hat{\boldsymbol{V}})$.

   ii. $\boldsymbol{\theta}^{(m)} = \left(\mu^{(m)}, \exp(\lambda^{(m)}), \eta^{(m)}, \exp(\omega^{(m)})\right)$.

Parameter Inference

- **Model:** $y_i \overset{\text{iid}}{\sim} t_{(\nu)}(\mu, \sigma, \eta)$
- **Loglikelihood:**

$$\ell(\mu, \sigma, \eta, \nu \mid \boldsymbol{y}) = \sum_{i=1}^{n} \texttt{dt(x = } y_i - \eta)/\sigma\texttt{, df = } \nu\texttt{, ncp = } \mu\texttt{, log = TRUE)} - n \log \sigma.$$

  So for this problem we could get away with MLE, or Approximate Bayesian Inference.

- **However:**
  - Don't have gradients for noncentral-t in **TMB**.
  - What if we had $y \mid \boldsymbol{x} \sim t_{(\nu)}(\mu, \sigma, \boldsymbol{x}'\boldsymbol{\beta})$?

Parameter Inference

- **Model:**

$$y_i \overset{\text{iid}}{\sim} t_{(\nu)}(\mu, \sigma, \eta) \qquad \Longleftrightarrow \qquad y_i = \frac{z_i}{\sqrt{x_i/\nu}} + \eta, \qquad \begin{array}{l} z_i \overset{\text{iid}}{\sim} \mathcal{N}(\mu, \sigma^2) \\ x_i \overset{\text{iid}}{\sim} \chi^2_{(\nu)} \end{array}$$

- **Observed Data:** $\boldsymbol{y}_{\text{obs}} = \boldsymbol{y} = (y_1, \ldots, y_n)$.
- **Missing Data:** $\boldsymbol{y}_{\text{miss}} = \boldsymbol{x} = (x_1, \ldots, x_n)$.
- **Complete Data:** $\boldsymbol{y}_{\text{comp}} = (\boldsymbol{y}, \boldsymbol{x})$, with

$$\begin{aligned} x_i &\overset{\text{iid}}{\sim} \chi^2_{(\nu)} \\ y_i \mid x_i &\overset{\text{ind}}{\sim} \mathcal{N}(\eta + \gamma/x_i^{1/2}, \tau^2/x_i), \end{aligned}$$

  where $\gamma = \mu\nu^{1/2}$ and $\tau = \sigma\nu^{1/2}$.

Parameter Inference

- **Model:** $y_i \overset{\text{iid}}{\sim} t_{(\nu)}(\mu, \sigma, \eta)$
- **Observed Data:** $\boldsymbol{y}_{\text{obs}} = \boldsymbol{y} = (y_1, \ldots, y_n)$.
- **Complete Data:** $\boldsymbol{y}_{\text{comp}} = (\boldsymbol{y}, \boldsymbol{x})$, with

$$
\begin{aligned}
x_i &\overset{\text{iid}}{\sim} \chi^2_{(\nu)} & \gamma &= \mu\nu^{1/2}, \\
y_i \mid x_i &\overset{\text{ind}}{\sim} \mathcal{N}(\eta + \gamma/x_i^{1/2}, \tau^2/x_i), & \tau &= \sigma\nu^{1/2}.
\end{aligned}
$$

- **Inference:** Let $\boldsymbol{\theta} = (\eta, \gamma, \tau^2, \nu)$.
  - **EM Algorithm:** This would require taking expectations of x, $x^{1/2}$, and $\log x$ with respect to

$$
\begin{aligned}
p(x \mid y, \boldsymbol{\theta}) &\propto \exp\left\{ -\frac{1}{2}\frac{(y - \eta - \gamma x^{-1/2})^2}{\tau^2/x} + \frac{1}{2}\log x + (\tfrac{\nu-2}{2})\log x - \frac{x}{2} \right\} \\
&\propto \exp\left\{ Ax + Bx^{1/2} + C\log x \right\},
\end{aligned}
$$

  a nonstandard distribution (don't even know its normalizing constant).

Parameter Inference

- **Model:** $y_i \overset{\text{iid}}{\sim} t_{(\nu)}(\mu, \sigma, \eta)$
- **Observed Data:** $\boldsymbol{y}_{\text{obs}} = \boldsymbol{y} = (y_1, \ldots, y_n)$.
- **Complete Data:** $\boldsymbol{y}_{\text{comp}} = (\boldsymbol{y}, \boldsymbol{x})$, with
$$x_i \overset{\text{iid}}{\sim} \chi^2_{(\nu)},$$
$$y_i \mid x_i \overset{\text{ind}}{\sim} \mathcal{N}(\eta + \gamma/x_i^{1/2}, \tau^2/x_i).$$
- **Inference:** Let $\boldsymbol{\theta} = (\eta, \gamma, \tau^2, \nu)$.
  - **EM Algorithm:** Requires expectations wrt $p(x \mid y, \boldsymbol{\theta}) \propto \exp\left\{Ax + Bx^{1/2} + C \log x\right\}$.
  - **Bayesian Data Augmentation:**

1. Implement an MCMC algorithm on the augmented posterior distribution
$$p(\boldsymbol{x}, \boldsymbol{\theta} \mid \boldsymbol{y}) \propto p(\boldsymbol{y}, \boldsymbol{x} \mid \boldsymbol{\theta}) \times \pi(\boldsymbol{\theta}).$$
2. If $(\boldsymbol{x}^{(1)}, \boldsymbol{\theta}^{(1)}), \ldots, (\boldsymbol{x}^{(M)}, \boldsymbol{\theta}^{(M)})$ is an MCMC sample from $p(\boldsymbol{x}, \boldsymbol{\theta} \mid \boldsymbol{y})$,

   then the stationary distribution of $\boldsymbol{\theta}^{(1)}, \ldots, \boldsymbol{\theta}^{(M)}$ is $p(\boldsymbol{\theta} \mid \boldsymbol{y}) = \int p(\boldsymbol{x}, \boldsymbol{\theta} \mid \boldsymbol{y})\, \mathrm{d}\boldsymbol{x}$.

   (Works for exactly the same reason that the histogram of each random variable in any MCMC converges to its marginal distribution.)

Bayesian Data Augmentation

- **Complete Data Likelihood:** Don't cancel out anything involving $\boldsymbol{\theta}$ or $\boldsymbol{x}$:

$$\begin{aligned}\ell(\boldsymbol{\theta} \mid \boldsymbol{x}, \boldsymbol{y}) &= \log p(\boldsymbol{y}, \boldsymbol{x} \mid \boldsymbol{\theta}) \\ &= -\frac{1}{2} \sum_{i=1}^{n} \left[\frac{(y_i - \eta - \gamma x_i^{-1/2})^2}{\tau^2 / x_i} - (\nu - 1) \log x_i + x_i \right] \\ &\quad - n \left[\frac{\tau^2 + \nu}{2} + \log \Gamma(\nu/2) \right].\end{aligned}$$

- **MCMC Algorithm:** A block Metropolis-within-Gibbs sampler with the following conditional updates:
  - **Update for $(\boldsymbol{\eta}, \boldsymbol{\gamma}, \boldsymbol{\tau})$:** Canceling everything that doesn't depend on $\beta = (\eta, \gamma)$ and τ, conditional likelihood $\ell(\beta, \tau \mid \nu, \boldsymbol{x}, \boldsymbol{y})$ is that of a regression-like model

$$y_i \overset{\text{ind}}{\sim} \mathcal{N}(\boldsymbol{u}_i' \beta, \tau^2 / x_i), \qquad \boldsymbol{u}_i = (1, 1/x_i^{1/2}).$$

Bayesian Data Augmentation

- **Complete Data Likelihood:**

$$\ell(\boldsymbol{\theta} \mid \boldsymbol{x}, \boldsymbol{y}) = -\frac{1}{2} \sum_{i=1}^{n} \left[\frac{(y_i - \eta - \gamma x_i^{-1/2})^2}{\tau^2 / x_i} - (\nu - 1) \log x_i + x_i \right] - n \left[\frac{\tau^2 + \nu}{2} + \log \Gamma\left(\frac{\nu}{2}\right) \right].$$

- **MCMC Algorithm:** A block Metropolis-within-Gibbs sampler with:
  - **Update for $(\boldsymbol{\eta}, \boldsymbol{\gamma}, \boldsymbol{\tau})$:** Canceling everything that doesn't depend on $\boldsymbol{\beta} = (\eta, \gamma)$ and τ, conditional likelihood $\ell(\boldsymbol{\beta}, \tau \mid \nu, \boldsymbol{x}, \boldsymbol{y})$ is that of a regression-like model

$$y_i \overset{\text{ind}}{\sim} \mathcal{N}(\boldsymbol{u}_i' \boldsymbol{\beta}, \tau^2 / x_i), \qquad \boldsymbol{u}_i = (1, 1/x_i^{1/2}).$$

  - **Conjugate Prior:** Multivariate Normal Inverse-Gamma (mNIX) distribution

$$(\boldsymbol{\beta}, \tau^2) \sim \text{mNIX}(\boldsymbol{\lambda}, \boldsymbol{\Sigma}, \alpha, \gamma) \qquad \iff \qquad \begin{aligned} \tau^2 &\sim \text{Inv-Gamma}(\alpha, \gamma) \\ \boldsymbol{\beta} \mid \tau^2 &\sim \mathcal{N}(\boldsymbol{\lambda}, \tau^2 \cdot \boldsymbol{\Sigma}). \end{aligned}$$

$\Longrightarrow$ Exact Gibbs update for $p(\boldsymbol{\beta}, \tau^2 \mid \nu, \boldsymbol{x}, \boldsymbol{y})$.

Bayesian Data Augmentation

- **Complete Data Likelihood:**

$$\ell(\boldsymbol{\theta} \mid \boldsymbol{x}, \boldsymbol{y}) = -\frac{1}{2} \sum_{i=1}^{n} \left[\frac{(y_i - \eta - \gamma x_i^{-1/2})^2}{\tau^2 / x_i} - (\nu - 1) \log x_i + x_i \right] - n \left[\frac{\tau^2 + \nu}{2} + \log \Gamma\left(\frac{\nu}{2}\right) \right].$$

- **MCMC Algorithm:** A block Metropolis-within-Gibbs sampler with:
  - **Update for ν:** Conditional likelihood is

$$\ell(\nu \mid \eta, \gamma, \tau, \boldsymbol{x}, \boldsymbol{y}) = -n \log \Gamma(\tfrac{1}{2}\nu) - \tfrac{1}{2}\nu \times \left(n \log(2) - \textstyle\sum_{i=1}^{n} \log x_i\right).$$

  - **Proposal Distribution:** Conditional likelihood only depends on $x_i \stackrel{\text{iid}}{\sim} \chi^2_{(\nu)}$ which is an Exponential Family $\implies \ell(\nu \mid \eta, \gamma, \tau, \boldsymbol{x}, \boldsymbol{y})$ is convex. Could do Newton-Raphson to obtain a mode-quadrature normal approximation, but easier to use a random walk proposal.
  - **Prior Distribution:** Use $\log \nu \sim \mathcal{N}(0, 2^2)$. Basically uninformative, since $\Pr(.005 < \nu < 170) \approx 99\%$ (recall that $t_{(\nu=1)} \sim$ Cauchy and $t_{(\nu \geq 30)} \approx \mathcal{N}(0,1)$). Think of this prior as regularizing inference (i.e., prevents ν from floating off to 0 or ∞).

Bayesian Data Augmentation

- **Complete Data Likelihood:**

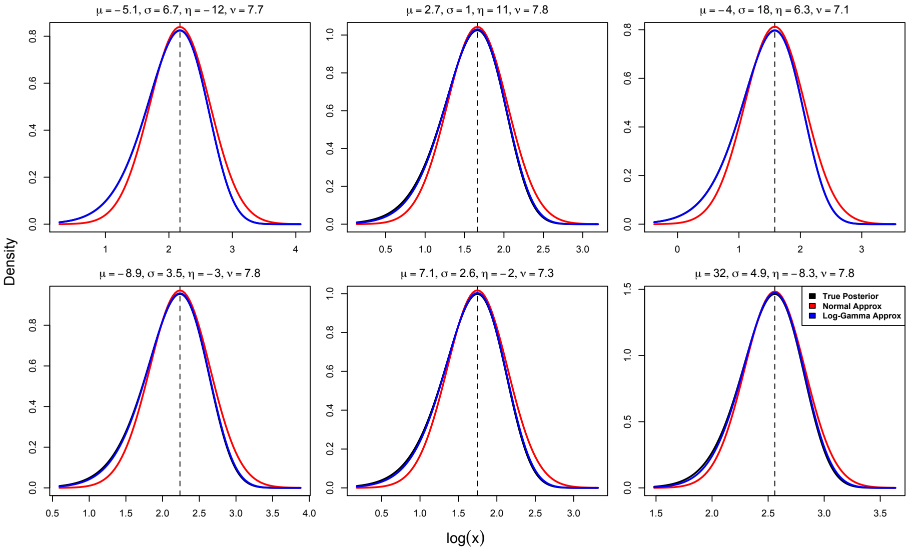
$$\ell(\boldsymbol{\theta} \mid \boldsymbol{x}, \boldsymbol{y}) = -\frac{1}{2}\sum_{i=1}^{n}\left[\frac{(y_i - \eta - \gamma x_i^{-1/2})^2}{\tau^2/x_i} - (\nu - 1)\log x_i + x_i\right] - n\left[\frac{\tau^2 + \nu}{2} + \log\Gamma\left(\frac{\nu}{2}\right)\right].$$

- **MCMC Algorithm:** A block Metropolis-within-Gibbs sampler with:
  - **Update for $\boldsymbol{x}$:** Conditional posterior is

$$p(\boldsymbol{x} \mid \boldsymbol{y}, \boldsymbol{\theta}) \propto \prod_{i=1}^{n} \exp\left\{A_i x_i + B_i x_i^{1/2} + C\log x_i\right\}.$$

  - **Proposal Distribution:**
    - Note that the x_i are conditionally independent given everything else $\implies$ exact Gibbs sampler produces IID samples.
    - Could do MWG, but this requires n tuning parameters (one for each x_i).
    - Note that mode of $Ax + Bx^{1/2} + C\log x$ has an analytic solution $\implies$ tuning-free MIID-within-Gibbs mode-quadrature proprosal.

Proposal Distribution for $p(x \mid y, \theta)$



MCMC Code Checking

- Much more difficult than checking that $\hat{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}} \ell(\boldsymbol{\theta} \mid \boldsymbol{y})$, since
  - MCMC is a random algorithm
  - Don't know much about $p(\boldsymbol{\theta} \mid \boldsymbol{y})$ – that's why we're doing MCMC in the first place!
- **Recommendation:** check code meticulously at <u>every</u> step.

  Whenever I skip a step, 99% of time there will be an error and then I don't know if it's in the last step or the one(s) I skipped. So I end up checking every step anyway, except now it takes longer.

Code Checking Strategies

1. Compare every *simplified* conditional likelihood $\ell(\theta_j \mid \boldsymbol{\theta}_{-j}, \boldsymbol{y})$ to the *unsimplified* likelihood $\log p(\boldsymbol{y} \mid \boldsymbol{\theta})$.

Difference between the two for any value of θ_j should be equal to a constant (possibly depending on $\boldsymbol{y}$ and $\boldsymbol{\theta}_{-j}$).

2. Compare every simplified posterior $p(\theta_j \mid \boldsymbol{\theta}_{-j}, \boldsymbol{y})$ to the unsimplified posterior $\mathcal{L}(\boldsymbol{\theta} \mid \boldsymbol{y}) \times \pi(\boldsymbol{\theta})$.

Same as for loglikelihoods, but now checking Jacobians, i.e., if prior is $\pi(\boldsymbol{\theta})$ but sampling is done on $\boldsymbol{\psi} = g(\boldsymbol{\theta})$, then $\pi(\boldsymbol{\psi}) = \pi\left(g^{-1}(\boldsymbol{\psi})\right) \left| \frac{\partial}{\partial \boldsymbol{\psi}} g^{-1}(\boldsymbol{\psi}) \right|$.

3. Compare *sampling* from $p(\theta_j \mid \boldsymbol{\theta}_{-j}, \boldsymbol{y})$ to analytic conditional.

To get analytic conditional, recall that $p(\theta_j \mid \boldsymbol{\theta}_{-j}, \boldsymbol{y}) \propto \mathcal{L}(\boldsymbol{\theta} \mid \boldsymbol{y}) \times \pi(\boldsymbol{\theta})$, to normalize evaluate 1-d integral numerically.

4. Compare sampling from $p(\boldsymbol{\theta} \mid \boldsymbol{y})$ for given MCMC to sample from same posterior with a different MCMC.

Both samplers should give same results.

Code Checking for Noncentral t

Notation: $\boldsymbol{\theta} = (\mu, \sigma, \eta, \nu)$, $\boldsymbol{\varphi} = (\eta, \gamma = \mu\nu^{1/2}, \tau^2 = \sigma^2\nu, \nu) = (\boldsymbol{\beta}, \tau^2, \nu)$.

1. Simplified vs unsimplified likelihoods:

$\ell(\eta, \gamma, \tau^2 \mid \nu, \boldsymbol{x}, \boldsymbol{y})$, $\ell(\nu \mid \eta, \gamma, \tau^2, \boldsymbol{x}, \boldsymbol{y})$, $\log p(\boldsymbol{x} \mid \boldsymbol{\varphi}, \boldsymbol{y})$ can each be checked against

$$p(\boldsymbol{y}, \boldsymbol{x} \mid \boldsymbol{\varphi}) = \underbrace{p(\boldsymbol{y} \mid \boldsymbol{x}, \eta, \gamma, \tau^2)}_{\overset{\text{ind}}{\sim} \mathcal{N}(\eta + \gamma \boldsymbol{x}^{-1/2}, \tau^2 \boldsymbol{x}^{-1})} \times \underbrace{p(\boldsymbol{x} \mid \nu)}_{\overset{\text{iid}}{\sim} \chi^2_{(\nu)}}$$

Code Checking for Noncentral t

Notation: $\boldsymbol{\theta} = (\mu, \sigma, \eta, \nu)$, $\boldsymbol{\varphi} = (\eta, \gamma = \mu\nu^{1/2}, \tau^2 = \sigma^2\nu, \nu) = (\boldsymbol{\beta}, \tau^2, \nu)$.

2. Conditional updates:

- $p(\nu \mid \ldots)$ and $p(x_i \mid \ldots)$ compare to analytic 1D posterior $\propto p(\boldsymbol{y}, \boldsymbol{x} \mid \boldsymbol{\varphi})\pi(\boldsymbol{\varphi})$.
- Prior: $\log(\nu) \sim \mathcal{N}(\mu_\nu, \sigma_\nu^2)$ $\qquad \boldsymbol{\beta}, \tau^2 \mid \nu \sim \mathrm{mNIX}(\alpha, \gamma, \boldsymbol{\lambda}, \boldsymbol{\Sigma})$

  As $\sigma_\nu, \boldsymbol{\Sigma} \to \infty$ and $\alpha, \gamma \to 0$ this becomes $\pi(\boldsymbol{\varphi}) \propto 1/\tau^2$
- To check $p(\boldsymbol{\beta}, \tau^2 \mid \nu, \boldsymbol{x}, \boldsymbol{y}) = \mathrm{mNIX}(\hat{\alpha}, \hat{\gamma}, \hat{\boldsymbol{\lambda}}, \hat{\boldsymbol{\Sigma}})$, note that for any $\boldsymbol{a} \in \mathbb{R}^2$,

$$\tau^2 \mid \nu, \boldsymbol{x}, \boldsymbol{y} \sim \text{Inv-Gamma}(\hat{\alpha}, \hat{;}\gamma), \qquad \frac{\boldsymbol{a}'\beta - \boldsymbol{a}'\hat{\boldsymbol{\lambda}}}{\sqrt{\hat{\gamma}/\hat{\alpha} \cdot \boldsymbol{a}'\hat{\boldsymbol{\Sigma}}\boldsymbol{a}}} \mid \nu, \boldsymbol{x}, \boldsymbol{y} \sim t_{(2\hat{\alpha})}$$

  Note that the second result integrates out τ^2.

Code Checking for Noncentral t

Notation: $\boldsymbol{\theta} = (\mu, \sigma, \eta, \nu)$, $\boldsymbol{\varphi} = (\eta, \gamma = \mu\nu^{1/2}, \tau^2 = \sigma^2\nu, \nu) = (\boldsymbol{\beta}, \tau^2, \nu)$.

3. Unconditional Updates:

- Compare to an MIID sampler with mode-quadrature normal proposals for $p(\boldsymbol{\theta} \mid \boldsymbol{y}) = p(\boldsymbol{y} \mid \boldsymbol{\theta})\pi(\boldsymbol{\theta})$.
- $p(\boldsymbol{y} \mid \boldsymbol{\theta})$ available through R's built-in function `dt` with `ncp` parameter.
- $\pi(\boldsymbol{\theta})$ obtained from $\pi(\boldsymbol{\varphi})$ through Jacobian. That is, if $f_{\boldsymbol{\varphi}}(\boldsymbol{\varphi})$ is PDF of prior on $\boldsymbol{\varphi}$, then PDF of prior on $\boldsymbol{\theta}$ is $f_{\boldsymbol{\theta}}(\boldsymbol{\theta}) = f_{\boldsymbol{\varphi}}(\boldsymbol{\varphi}) \times |\mathrm{d}\boldsymbol{\varphi}/\,\mathrm{d}\boldsymbol{\theta}|$, where

$$\frac{\mathrm{d}\boldsymbol{\varphi}}{\mathrm{d}\boldsymbol{\theta}} = \begin{bmatrix} 0 & \nu^{1/2} & 0 & 0 \\ 0 & 0 & 2\sigma\nu & 0 \\ 1 & 0 & 0 & 0 \\ 0 & \frac{1}{2}\mu\nu^{-1/2} & \sigma^2 & 1 \end{bmatrix} \qquad \Longrightarrow \qquad \left| \frac{\mathrm{d}\boldsymbol{\varphi}}{\mathrm{d}\boldsymbol{\theta}} \right| = 2\sigma\nu^{3/2}.$$

Code Checking for Noncentral t

Notation: $\boldsymbol{\theta} = (\mu, \sigma, \eta, \nu)$, $\boldsymbol{\varphi} = (\eta, \gamma = \mu\nu^{1/2}, \tau^2 = \sigma^2\nu, \nu) = (\boldsymbol{\beta}, \tau^2, \nu)$.

4. Compare to different MCMC on same posterior:

- Since this is a 4-parameter problem, probably easiest to compare to MIID sampling with normal mode-quadrature proposals.
- For more complicated problems, perhaps easier to use a general-purpose MCMC, which will be slow but easy to program.
- **Stan:** The state-of-the-art in general-purpose MCMC.
  - Stan is a programming language very similar to R to which you input an arbitrary $\log p(\boldsymbol{\theta} \mid \boldsymbol{y})$.
  - Implements and compiles in C++ a very effective MCMC algorithm called Hybrid Monte Carlo (HMC), but usually referred to as **Hamiltonian Monte Carlo**.

Hamiltonian Monte Carlo (HMC)

- **Problem:** Sample from $\boldsymbol{x} \sim p(\boldsymbol{x}) \propto \exp\{\Omega(\boldsymbol{x})\}, \qquad \boldsymbol{x} = (x_1, \ldots, x_d)$.
- **Hamiltonian Dynamics:**
  - **System Variables:**
    - *Position Variables* $\boldsymbol{x} = (x_1, \ldots, x_d)$.
    - *Momentum Variables* $\boldsymbol{v} = (v_1, \ldots, v_d)$.
    - *Phase-Space Variables* $\boldsymbol{\Gamma} = (\boldsymbol{x}, \boldsymbol{v})$.
  - **Hamiltonian Function:** $\mathcal{H}(\boldsymbol{x}, \boldsymbol{v}) = -\Omega(\boldsymbol{x}) + \frac{1}{2}\sum_{i=1}^{d} \frac{v_i^2}{m_i}$.
  - **Equations of Motion:** Consider the function $\boldsymbol{\Gamma}_t = \boldsymbol{\Gamma}(t)$ defined by the system of ordinary differential equations (ODEs) and initial conditions

$$\frac{\mathrm{d}}{\mathrm{d}t} x_i(t) = \frac{v_i(t)}{m_i}, \qquad \frac{\mathrm{d}}{\mathrm{d}t} v_i(t) = \frac{\partial}{\partial x_i}\Omega(\boldsymbol{x}_t), \qquad \boldsymbol{\Gamma}_0 = (\boldsymbol{x}_0, \boldsymbol{v}_0).$$

    Thus we have some function $\boldsymbol{\Psi} : \mathbb{R}^{2d} \times \mathbb{R} \to \mathbb{R}^{2d}$ such that $\boldsymbol{\Psi}(\boldsymbol{\Gamma}_0, t) = \boldsymbol{\Gamma}_t$.

Hamiltonian Monte Carlo (HMC)

- **Problem:** Sample from $\boldsymbol{x} \sim p(\boldsymbol{x}) \propto \exp\{\Omega(\boldsymbol{x})\}, \qquad \boldsymbol{x} = (x_1, \ldots, x_d)$.
- **Hamiltonian Dynamics:**
  - **System Variables:** $\boldsymbol{x}$ (position), $\boldsymbol{v}$ (momentum), $\boldsymbol{\Gamma} = (\boldsymbol{x}, \boldsymbol{v})$ (phase-space).
  - **Hamiltonian Function:** $$\mathcal{H}(\boldsymbol{x}, \boldsymbol{v}) = -\Omega(\boldsymbol{x}) + \frac{1}{2}\sum_{i=1}^{d} \frac{v_i^2}{m_i}.$$
  - **Equations of Motion:** Define $\boldsymbol{\Gamma}_t = \boldsymbol{\Psi}(\boldsymbol{\Gamma}_0, t)$ as solution to system of ODEs
  $$\frac{\mathrm{d}}{\mathrm{d}t} x_i(t) = \frac{v_i(t)}{m_i}, \qquad \frac{\mathrm{d}}{\mathrm{d}t} v_i(t) = \frac{\partial}{\partial x_i}\Omega(\boldsymbol{x}_t), \qquad \boldsymbol{\Gamma}_0 = (\boldsymbol{x}_0, \boldsymbol{v}_0).$$
  - **Conservation of Energy:** If $\boldsymbol{\Gamma}_t = \boldsymbol{\Psi}(\boldsymbol{\Gamma}_0, t)$, then $\mathcal{H}(\boldsymbol{x}_0, \boldsymbol{v}_0) = \mathcal{H}(\boldsymbol{x}_t, \boldsymbol{v}_t)$.
  - **Preservation of Volume:** Change of variables $\boldsymbol{\Gamma}_0 \to \boldsymbol{\Gamma}_t$ has Jacobian
  $$\left|\frac{\mathrm{d}\boldsymbol{\Gamma}_t}{\mathrm{d}\boldsymbol{\Gamma}_0}\right| = \left|\frac{\mathrm{d}}{\mathrm{d}\boldsymbol{\Gamma}_0}\boldsymbol{\Psi}(\boldsymbol{\Gamma}_0, t)\right| = 1.$$

Hamiltonian Monte Carlo (HMC)

- **Problem:** Sample from $\boldsymbol{x} \sim p(\boldsymbol{x}) \propto \exp\{\Omega(\boldsymbol{x})\}, \qquad \boldsymbol{x} = (x_1, \ldots, x_d).$
- **Hamiltonian Dynamics:**
  - **Hamiltonian Function:** $\mathcal{H}(\boldsymbol{x}, \boldsymbol{v}) = -\Omega(\boldsymbol{x}) + \frac{1}{2}\sum_{i=1}^{d} \frac{v_i^2}{m_i}.$
  - **Equations of Motion:** Define $\boldsymbol{\Gamma}_t = \boldsymbol{\Psi}(\boldsymbol{\Gamma}_0, t)$ as solution to system of ODEs

$$\frac{\mathrm{d}}{\mathrm{d}t} x_i(t) = \frac{v_i(t)}{m_i}, \qquad \frac{\mathrm{d}}{\mathrm{d}t} v_i(t) = \frac{\partial}{\partial x_i}\Omega(\boldsymbol{x}_t), \qquad \boldsymbol{\Gamma}_0 = (\boldsymbol{x}_0, \boldsymbol{v}_0).$$

- **(Idealized) HMC Proposal:** Given $\boldsymbol{x}_{\mathrm{curr}}$ and $L > 0$:

1. Let $\boldsymbol{\Gamma}_0 = (\boldsymbol{x}_{\mathrm{curr}}, \boldsymbol{v}_0)$, where $v_{0i} \overset{\mathrm{ind}}{\sim} \mathcal{N}(0, m_i)$.
2. Let $\boldsymbol{x}_{\mathrm{prop}} = \boldsymbol{x}_L$, where $(\boldsymbol{x}_L, \boldsymbol{v}_L) = \boldsymbol{\Gamma}_L = \boldsymbol{\Psi}(\boldsymbol{\Gamma}_0, L)$

$\Longrightarrow$ Metropolis-Hastings acceptance rate:

Hamiltonian Monte Carlo (HMC)

- **Problem:** Sample from $\boldsymbol{x} \sim p(\boldsymbol{x}) \propto \exp\{\Omega(\boldsymbol{x})\}, \qquad \boldsymbol{x} = (x_1, \ldots, x_d).$
- **Hamiltonian Dynamics:**
  - **Hamiltonian Function:** $\mathcal{H}(\boldsymbol{x}, \boldsymbol{v}) = -\Omega(\boldsymbol{x}) + \frac{1}{2}\sum_{i=1}^{d} \frac{v_i^2}{m_i}.$
  - **Equations of Motion:** Define $\boldsymbol{\Gamma}_t = \boldsymbol{\Psi}(\boldsymbol{\Gamma}_0, t)$ as solution to system of ODEs

$$\frac{\mathrm{d}}{\mathrm{d}t} x_i(t) = \frac{v_i(t)}{m_i}, \qquad \frac{\mathrm{d}}{\mathrm{d}t} v_i(t) = \frac{\partial}{\partial x_i}\Omega(\boldsymbol{x}_t), \qquad \boldsymbol{\Gamma}_0 = (\boldsymbol{x}_0, \boldsymbol{v}_0).$$

- **(Idealized) HMC Proposal:** Given $\boldsymbol{x}_{\mathrm{curr}}$ and $L > 0$:

1. Let $\boldsymbol{\Gamma}_0 = (\boldsymbol{x}_{\mathrm{curr}}, \boldsymbol{v}_0)$, where $v_{0i} \overset{\mathrm{ind}}{\sim} \mathcal{N}(0, m_i)$.
2. Let $\boldsymbol{x}_{\mathrm{prop}} = \boldsymbol{x}_L$, where $(\boldsymbol{x}_L, \boldsymbol{v}_L) = \boldsymbol{\Gamma}_L = \boldsymbol{\Psi}(\boldsymbol{\Gamma}_0, L)$

$\Longrightarrow$ Metropolis-Hastings acceptance rate: 100%!!!

Hamiltonian Monte Carlo (HMC)

- **Problem:** Sample from $\boldsymbol{x} \sim p(\boldsymbol{x}) \propto \exp\{\Omega(\boldsymbol{x})\}, \qquad \boldsymbol{x} = (x_1, \ldots, x_d).$
- **(Idealized) HMC Proposal:** Given $\boldsymbol{x}_{\text{curr}}$ and $L > 0$:
  1. Let $\boldsymbol{\Gamma}_0 = (\boldsymbol{x}_{\text{curr}}, \boldsymbol{v}_0)$, where $v_{0i} \overset{\text{ind}}{\sim} \mathcal{N}(0, m_i)$.
  2. Let $\boldsymbol{x}_{\text{prop}} = \boldsymbol{x}_L$, where $(\boldsymbol{x}_L, \boldsymbol{v}_L) = \boldsymbol{\Gamma}_L = \boldsymbol{\Psi}(\boldsymbol{\Gamma}_0, L)$

  $\implies$ Metropolis-Hastings acceptance rate: 100%!!!
- **In Practice:**
  - Can't solve ODE exactly: discretize $\implies$ acceptance rate $\neq 1$.
  - Lots of tuning parameters: ODE solver step size, total time t, mass m.
  - Gradients: To solve ODE need $\frac{\partial}{\partial x_i}\Omega(\boldsymbol{x})$, which is a lot of programming effort.

Hamiltonian Monte Carlo (HMC)

- **Problem:** Sample from $\boldsymbol{x} \sim p(\boldsymbol{x}) \propto \exp\{\Omega(\boldsymbol{x})\}, \qquad \boldsymbol{x} = (x_1, \ldots, x_d)$.
- **(Idealized) HMC Proposal:** Given $\boldsymbol{x}_{\mathrm{curr}}$ and $L > 0$:

  1. Let $\boldsymbol{\Gamma}_0 = (\boldsymbol{x}_{\mathrm{curr}}, \boldsymbol{v}_0)$, where $v_{0i} \overset{\mathrm{ind}}{\sim} \mathcal{N}(0, m_i)$.
  2. Let $\boldsymbol{x}_{\mathrm{prop}} = \boldsymbol{x}_L$, where $(\boldsymbol{x}_L, \boldsymbol{v}_L) = \boldsymbol{\Gamma}_L = \boldsymbol{\Psi}(\boldsymbol{\Gamma}_0, L)$

  $\implies$ Metropolis-Hastings acceptance rate: 100%!!!

- **In Practice:**
  - Can't solve ODE exactly: discretize $\implies$ acceptance rate $\neq 1$.
  - Lots of tuning parameters: ODE solver step size, total time t, mass m.
  - Gradients: To solve ODE need $\frac{\partial}{\partial x_i}\Omega(\boldsymbol{x})$, which is a lot of programming effort.

  All of this is done automatically by Stan :)

Stan Examples

- **Examples:**

  1. **Curved Mean-Variance Normal:** $\sigma \sim p(\sigma \mid \boldsymbol{y})$, where

  $$\sigma \sim \chi^2_{(7)}, \qquad y_i \mid \sigma \stackrel{\text{iid}}{\sim} \mathcal{N}(\sigma, \sigma^2).$$

  2. **Banana distribution:** $\boldsymbol{x} \sim p(\boldsymbol{x} \mid \sigma, y)$, where $\boldsymbol{x} = (x_1, x_2)$ and

  $$p(\boldsymbol{x} \mid \sigma, y) \propto \exp\left\{ -\left[\frac{(y - x_1 \cdot x_2)^2}{2\sigma^2} + \frac{(x_1 - x_2)^2}{2} \right] \right\}$$

- **Key Concepts:**

  - **Testing Stan code:** Generic MCMC, so only need to check log-posterior is correct. Do this with **rstan** package functions `log_prob` and `expose_stan_functions`.

  - **Testing other code:** Stan is relatively easy to program, so use it to compare to sampling from a more specific MCMC algorithm for a particular problem (can often do better than any generic algorithm at expense of human hours).

Stan Resources

- Instructions for installing Stan in R can be found here. Follow these **to the letter** or Stan probably won't work properly!
- Full Stan documentation (tons of examples) and `rstan` package vignette (for operating Stan from within R) can be found here.
- Detailed explanation of HMC algorithm, its strengths and pitfalls, and many of its variants can be found here.