Lecture 4:
Introduction to Bayesian Statistics /MCMC methods

Bruce Walsh lecture notes
2013 Synbreed course
version 2 July 2013
Overview

• What is Bayesian Statistics?
  - Priors and posteriors

• Why Bayesian?

• Details for priors
  - Prior hyperparameters
  - Informative vs. flat priors
  - Conjugating priors

• Describing the posterior
  - Credible intervals (highest density regions)
  - “hypothesis testing” via Bayes factors
Bayesian statistics

• **Classic ("frequentist") statistics**
  - typically report POINT estimates
  - Define probabilities as the frequency of events if experiment is replicated over and over.
    • Parameters fixed, data random

• **Bayesian statistics**
  - returns a FULL DISTRIBUTION of the possible parameter values $\theta$ given the data $x$, the Posterior $p(\theta \mid x)$
    • Data fixed, parameters random
Bayes' theorem

\[
\text{Prob}(\Theta \mid x) = \frac{\text{Pr}(x \mid \Theta) \text{Pr}(\Theta)}{\text{Pr}(x)}
\]

\[
\text{Prob}(\Theta \mid x) = \text{posterior for } \Theta \text{ given the data } x.
\]

\[
\text{Prob}(x \mid \Theta) = \text{likelihood of } x \text{ given } \Theta.
\]

\[
\text{Prob}(\Theta) = \text{prior on } \Theta \text{ before the experiment.}
\]

Typically write posterior = \( C \times \text{likelihood} \times \text{prior} \)
where \( C = 1/\text{Pr}(X) \) is a constant so that the posterior integrates to one.

Can think of Bayesian statistics as a natural extension of likelihood.
\[ \text{Prob}(\Theta \mid x) = \text{posterior for } \Theta \text{ given the data } x. \]

\[ = \text{result when the data and prior information are jointly considered} \]

\[ \text{Prob}(x \mid \Theta) = \text{likelihood of } x \text{ given } \Theta. \]

\[ = \text{signal on the value of } \Theta \text{ given the observed data} \]

\[ \text{Prob}(\Theta) = \text{prior on } \Theta \text{ before the experiment.} \]

\[ = \text{prior information (distribution) of possible } \Theta \text{ before the experiment} \]

\text{A flat prior means all possible values are equally likely (no prior information)
Example: Normal with unknown mean

Known Variance and Unknown Mean
Assume the variance $\sigma^2$ is known, while the mean $\mu$ is unknown. For a Bayesian analysis, it remains to specify the prior for $\mu$, $p(\mu)$. Suppose we assume a Gaussian prior, $\mu \sim \mathcal{N}(\mu_0, \sigma_0^2)$, so that

$$p(\mu) = \frac{1}{\sqrt{2\pi\sigma_0^2}} \exp\left(-\frac{(\mu - \mu_0)^2}{2\sigma_0^2}\right)$$  \hspace{1cm} (A2.22)

The mean and variance of the prior, $\mu_0$ and $\sigma_0^2$ are referred to as hyperparameters. Here, $\mu_0$ specifies a prior location for the parameter, while $\sigma^2$ specifies our uncertainty in this prior location – the larger the variance, the greater our uncertainty.

With the prior given by Equation A2.22, we can express the resulting posterior distribution as

$$p(\mu \mid x) \propto \ell(\mu \mid x) \cdot p(\mu)$$

$$\propto \exp\left(-\frac{(\mu - \mu_0)^2}{2\sigma_0^2} - \frac{1}{2\sigma^2} \left[\sum_{i=1}^{n} x_i^2 - 2\mu \bar{x} + n\mu^2\right]\right)$$  \hspace{1cm} (A2.24a)

We can factor out additional terms not involving $\mu$ to obtain

$$p(\mu \mid x) \propto \exp\left(-\frac{\mu^2}{2\sigma_0^2} + \frac{\mu \mu_0}{\sigma_0^2} + \frac{\mu \bar{x}}{\sigma^2} - \frac{n\mu^2}{2\sigma^2}\right)$$  \hspace{1cm} (A2.24b)
Factoring in terms of $\mu$, the term in the exponential becomes

$$-\frac{\mu^2}{2} \left( \frac{1}{\sigma_0^2} + \frac{n}{\sigma^2} \right) + \mu \left( \frac{\mu_0}{\sigma_0^2} + \frac{n\bar{x}}{\sigma^2} \right) = -\frac{\mu^2}{2\sigma_*^2} + \frac{2\mu\mu_*}{2\sigma_*^2} \quad (A2.25a)$$

where

$$\sigma_*^2 = \left( \frac{1}{\sigma_0^2} + \frac{n}{\sigma^2} \right)^{-1} \quad \text{and} \quad \mu_* = \sigma_*^2 \left( \frac{\mu_0}{\sigma_0^2} + \frac{n\bar{x}}{\sigma^2} \right) \quad (A2.25b)$$

Finally, by completing the square, we have

$$p(\mu \mid x) \propto \exp \left( -\frac{(\mu - \mu_*)^2}{2\sigma_*^2} + f(x, \mu_0, \sigma^2, \sigma_0^2) \right) \quad (A2.25c)$$

The posterior density function for $\mu$ thus becomes

$$p(\mu \mid x) \propto \exp \left( -\frac{(\mu - \mu_*)^2}{2\sigma_*^2} \right) \quad (A2.26a)$$

showing that the posterior density function for $\mu$ is a normal with mean $\mu_*$ and variance $\sigma_*^2$, e.g.,

$$\mu \mid (x, \sigma^2) \sim N(\mu_*, \sigma_*^2) \quad (A2.26b)$$

Notice that the posterior density is in the same form as the prior. This occurred because the prior conjugated with the likelihood function – the product of the prior and likelihood resulted a distribution in the same family as the prior. The use of such conjugate priors (for a given likelihood) is a key concept in Bayesian analysis and we explore it more fully below.
Posterior = \[ \mu | (x, \sigma^2) \sim N(\mu_*, \sigma_*^2) \]

\[ \sigma_*^2 = \left( \frac{1}{\sigma_0^2} + \frac{n}{\sigma^2} \right)^{-1} \]

and \[ \mu_* = \sigma_*^2 \left( \frac{\mu_0}{\sigma_0^2} + \frac{n \bar{x}}{\sigma^2} \right) \]

**Signal from prior**

When variance of prior \( \sigma_0^2 \) small, strong signal, pulls posterior towards \( \mu_0 \)

Shrinks (or regresses) sample mean back towards prior mean \( \mu_0 \)

**Signal from data**

Strength of signal increases with sample size \( n \).
Bayesian = random effects

- Every parameter is a Bayesian analysis is a random effect
- Recall that random effects models allow us to deal with high-dimensional data sets
- Also gives EXACT maximum likelihood results for any sample size (given the chosen prior)
Marginal posteriors

• Often have parameters of interest and additional nuisance parameters needed for the model but of no interest

• A marginal posterior integrates the full posterior over the nuisance parameters to remove their effects

• The resulting marginal posterior fully incorporates how any uncertainly in the nuisance parameters influences the distribution of the parameters of interest
Why Bayesian?

- Historical: deep philosophical differences on the definition of probability
- Real world (today):
  - Allow for incorporation of any prior information
  - Easily handles all of the uncertainty in high-dimensional data sets (as everything is a random effect)
  - Marginal posteriors
  - Computationally feasible with MCMC resampling methods (easily allow us to generate draws from the posterior or any marginal posterior)
Priors

• The hyperparameters are the values of the parameters in the prior (e.g., the mean and variance for a gaussian prior)
• Empirical Bayes uses ML to estimate the hyperparameters given the data.
• Flat (or uninformative) priors have very high variances, spreading out prior support around some mean value.
• In the limit, the probability density function for the prior is a uniform. Care may be needed when using a uniform if the parameter range is infinite.
Conjugating priors

• A conjugating prior returns a posterior of the same distributional family, but with different parameters
  - Example for gaussian prior on mean under gaussian likelihood function

• Specific parameters in many common likelihoods have known conjugate priors
Table A22 Conjugate priors for common likelihood functions. If one uses the distribution family of the conjugate prior with its paired likelihood function, the resulting posterior is in the same distribution family (albeit, of course, with different parameters) as the prior.

<table>
<thead>
<tr>
<th>Likelihood</th>
<th>Conjugate prior</th>
</tr>
</thead>
<tbody>
<tr>
<td>Binomial</td>
<td>Beta</td>
</tr>
<tr>
<td>Multinomial</td>
<td>Dirichlet</td>
</tr>
<tr>
<td>Poisson</td>
<td>Gamma</td>
</tr>
<tr>
<td>Normal</td>
<td></td>
</tr>
<tr>
<td>$\mu$ unknown, $\sigma^2$ known</td>
<td>Normal</td>
</tr>
<tr>
<td>$\mu$ known, $\sigma^2$ unknown</td>
<td>Inverse Chi-Square</td>
</tr>
<tr>
<td>Multivariate Normal</td>
<td></td>
</tr>
<tr>
<td>$\mu$ unknown, $\mathbf{V}$ known</td>
<td>Multivariate Normal</td>
</tr>
<tr>
<td>$\mu$ known, $\mathbf{V}$ unknown</td>
<td>Inverse Wishart</td>
</tr>
</tbody>
</table>

Conjugation is very important in some of the MCMC methods Motivation for otherwise “odd” priors that you mean see in the literature.
SUMMARIZING THE POSTERIOR DISTRIBUTION

How do we extract a Bayes estimator for some unknown parameter \( \theta \)? If our mindset is to use some sort of point estimator (as is usually done in classical statistics), there are a number of candidates. We could follow maximum likelihood and use the mode of the distribution (its maximal value), with

\[
\hat{\theta} = \max_{\theta} \left[ p(\theta | x) \right] \quad (A2.8a)
\]

We could take the expected value of \( \theta \) given the posterior,

\[
\hat{\theta} = E[\theta | x] = \int \theta p(\theta | x) d\theta \quad (A2.8b)
\]

Another candidate is the median of the posterior distribution, where the estimator satisfies \( \Pr(\theta > \hat{\theta} | x) = \Pr(\theta < \hat{\theta} | x) = 0.5 \), hence

\[
\int_{\hat{\theta}}^{+\infty} p(\theta | x) d\theta = \int_{-\infty}^{\hat{\theta}} p(\theta | x) d\theta = \frac{1}{2} \quad (A2.8c)
\]

However, using any of the above estimators, or even all three simultaneously, loses the full power of a Bayesian analysis, as the full estimator is the entire posterior density itself. If we cannot obtain the full form of the posterior distribution, it may still be possible to obtain one of the three above estimators. However, as we will see in Appendix 3, we can generally obtain the posterior by simulation using MCMC sampling, and hence the Bayes estimate of a parameter is frequently presented as a frequency histogram from MCMC-generated samples of the posterior distribution.
Highest Density Regions (HDRs)

Given the posterior distribution, construction of confidence intervals is obvious. For example, a $100(1 - \alpha)$ confidence interval is given by any $(L_{\alpha/2}, H_{\alpha/2})$ satisfying

$$\int_{L_{\alpha/2}}^{H_{\alpha/2}} p(\theta | x) d\theta = 1 - \alpha$$

To reduce possible candidates, one typically uses highest density regions, or HDRs, where for a single parameter the HDR $100(1 - \alpha)$ region(s) are the shortest intervals giving an area of $(1 - \alpha)$. More generally, if multiple parameters are being estimated, the HDR region(s) are those with the smallest volume in the parameter space. HDRs are also referred to as Bayesian confidence intervals or credible intervals.

It is critical to note that there is a profound difference between a confidence interval (CI) from classical (frequentist) statistics and a Bayesian interval. The interpretation of a classical confidence interval is that is we repeat the experiment a large number of times, and construct CIs in the same fashion, $(1 - \alpha)$ of the time the confidence intervals will enclose the (unknown) parameter. In contrast, with a Bayesian HDR, there is a $(1 - \alpha)$ probability that the interval contains the true value of the unknown parameter. While these two intervals sound essentially identical, they are not and indeed are fundamentally (but subtly) different. Often the CI and Bayesian intervals have essentially the same value, but again the interpretational difference remains. The key point is that the Bayesian prior allows us to make direct probability statements about $\theta$, while under classical statistics we can only make statements about the behavior of the statistic if we repeat an experiment a large number of times. Given the important conceptual difference between classical and Bayesian intervals, Bayesians typically avoid using the term confidence interval.
Bayes Factors and Hypothesis Testing

In the classical hypothesis testing framework, we have two alternatives. The null hypothesis $H_0$ that the unknown parameter $\theta$ belongs to some set or interval $\Theta_0 (\theta \in \Theta_0)$, versus the alternative hypothesis $H_1$ that $\theta$ belongs to the alternative set $\Theta_1 (\theta \in \Theta_1)$. $\Theta_0$ and $\Theta_1$ contain no common elements ($\Theta_0 \cap \Theta_1 = \emptyset$) and the union of $\Theta_0$ and $\Theta_1$ contains the entire space of values for $\theta$ (i.e., $\Theta_0 \cup \Theta_1 = \Theta$).

In the classical statistical framework of the frequentists, one uses the observed data to test the significance of a particular hypothesis, and (if possible) compute a $p$-value (the probability $p$ of observing an as extreme value of the test statistic if the null hypothesis is indeed correct). Hence, at first blush one would think that the idea of a hypothesis test is trivial in a Bayesian framework, as using the posterior distribution gives us expected $p$ values, as

$$\Pr(\theta > \theta_0) = \int_{\theta_0}^{\infty} p(\theta | x) \, d\theta \quad \text{and} \quad \Pr(\theta_0 < \theta < \theta_1) = \int_{\theta_0}^{\theta_1} p(\theta | x) \, d\theta$$

The kicker with a Bayesian analysis is that we also have prior information and Bayesian hypothesis testing addresses whether, given the data, we are more or less inclined towards the hypothesis than we initially were.
For example, suppose that the prior distribution of $\theta$ is such that $\Pr(\theta > \theta_0) = 0.10$, while for the posterior distribution $\Pr(\theta > \theta_0) = 0.05$. The later is significant at the 5 percent level in a classical hypothesis testing framework, but the data only doubles our confidence in the alternative hypothesis relative to our belief based on prior information. If $\Pr(\theta > \theta_0) = 0.50$ for the prior, then a 5% posterior probability would greatly increase our confidence in the alternative hypothesis. Hence, the prior probabilities certainly influence hypothesis testing. To formalize this idea, let

$$p_0 = \Pr(\theta \in \Theta_0 \mid x), \quad p_1 = \Pr(\theta \in \Theta_1 \mid x)$$  \hspace{1cm} \text{(A2.9a)}$$

denote the probability, given the observed data $x$, that $\theta$ is in the null ($p_0$) and alternative ($p_1$) hypothesis sets. Note that these are posterior probabilities. Since $\Theta_0 \cap \Theta_1 = \emptyset$ and $\Theta_0 \cup \Theta_1 = \Theta$, it follows that $p_0 + p_1 = 1$. Likewise, for the prior probabilities we have

$$\pi_0 = \Pr(\theta \in \Theta_0), \quad \pi_1 = \Pr(\theta \in \Theta_1)$$  \hspace{1cm} \text{(A2.9b)}$$

Thus the prior odds of $H_0$ versus $H_1$ are $\pi_0 / \pi_1$, while the posterior odds are $p_0 / p_1$. 
The **Bayes factor** $B_0$ in favor of $H_0$ versus $H_1$ is given by the ratio of the posterior odds divided by the prior odds,

$$B_0 = \frac{p_0/p_1}{\pi_0/\pi_1} = \frac{p_0 \pi_1}{p_1 \pi_0} \quad (A2.10a)$$

The Bayes factor is loosely interpreted as the odds in favor of $H_0$ versus $H_1$ that are given by the data. Since $\pi_1 = 1 - \pi_0$ and $p_1 = 1 - p_0$, we can also express this as

$$B_0 = \frac{p_0(1 - \pi_0)}{\pi_0(1 - p_0)} \quad (A2.10b)$$

Likewise, by symmetry note that the Bayes factor $B_1$ in favor of $H_1$ versus $H_0$ is just

$$B_1 = 1/B_0 \quad (A2.10c)$$
Markov Chain Monte Carlo (MCMC)

Much more details in the online notes
Overview

• Simulating draws from complex distributions

• Markov chains
  - Definition
  - Irreducible, aperiodic chains
  - Stationarity

• MCMC samplers
  - Metropolis-Hasting algorithm
  - Burning in the sampler
  - Simulated annealing
  - Gibbs sampling
What is MCMC?

• Suppose we have an expression for some very complex distribution (such as a posterior)
• Ideally, it would be nice to have what amounts to a button on our calculator that generates a random drawn from this distribution
• By generating a large number of such draws, we can fully characterize the distribution
• MCMC (markov chain monte carlo) provides a way to do this
Markov Processes

- A **Markov process** is a stochastic (random) process such that the probability of having value $x$ at time $t$ simply depends on the value at time $t-1$ and not on any of the previous values.

- $\Pr(X[t] = x \mid X[0] = x_0, X[1] = x_1, \ldots, x[t-1] = x_{t-1}) = \Pr(X[t] = x \mid x[t-1] = x_{t-1})$

- Most stochastic processes we think of are Markovian.
Markov chain

• A MC is simply the realization of a Markov process over time

• If the state space (the collection of possible values) is discrete, a simple matrix equation describes the evolution of a Markov process

• Let $\pi$ be a column vector whose elements correspond to the state space, so that the $i$th element of $\pi_t$ is the probability the chain has state $i$ at time $t$.

• Let the $ij$th element of the matrix $P$ be the transition probability from state $i$ to state $j$

• $\pi_1 = P \pi_0$. Hence, $\pi_t = P^t \pi_0$
Example A3.1. Suppose the state space consists of three possible weather conditions (Rain, Sunny, Cloudy) and weather patterns follows a Markov process (of course, they do not!). Thus, the probability of tomorrow’s weather simply depends on today’s weather, and not any other previous days. If this is the case, the observation that it has rained for three straight days does not alter the probability of tomorrow weather compared to the situation where (say) it rained today but was sunny for the last week. Suppose the probability transitions given today is rainy are

\[
P(\text{Rain tomorrow} \mid \text{Rain today}) = 0.5, \\
P(\text{Sunny tomorrow} \mid \text{Rain today}) = 0.25, \\
P(\text{Cloudy tomorrow} \mid \text{Rain today}) = 0.25,
\]

The first row of the transition probability matrix thus becomes \((0.5, 0.25, 0.25)\). Suppose the rest of the transition matrix is given by

\[
P = \begin{pmatrix}
0.5 & 0.25 & 0.25 \\
0.5 & 0 & 0.5 \\
0.25 & 0.25 & 0.5
\end{pmatrix}
\]

Note that this Markov chain is irreducible, as all states communicate with each other.

The \(i\)th entry in row \(j\) is the probability of jumping to state \(i\) given the current state is \(j\).
Example (cont)

Suppose today is sunny. What is the expected weather two days from now? Seven days? Here
\(\pi(0) = (0 \ 1 \ 0)\), giving

\[ \pi(2) = \pi(0)P^2 = (0.375 \ 0.25 \ 0.375) \]

and

\[ \pi(7) = \pi(0)P^7 = (0.4 \ 0.2 \ 0.4) \]

Conversely, suppose today is rainy, so that \(\pi(0) = (1 \ 0 \ 0)\). The expected weather becomes

\[ \pi(2) = (0.4375 \ 0.1875 \ 0.375) \quad \text{and} \quad \pi(7) = (0.4 \ 0.2 \ 0.4) \]

Note that after a sufficient amount of time, the expected weather is independent of the starting value. In other words, the chain has reached a \textbf{stationary distribution}, where the probability values are independent of the actual starting value.
Stationarity

A Markov chain is said to be **irreducible** if there exists a positive integer such that $p_{ij}^{(n)} > 0$ for all $i, j$. That is, all states communicate with each other, as one can always go from any state to any other state (although it may take more than one step). Likewise, a chain is said to be **aperiodic** when the number of steps required to move between two states (say $x$ and $y$) is not required to be multiple of some integer. Put another way, the chain is not forced into some cycle of fixed length between certain states.

A Markov chain may reach a **stationary distribution** $\pi^*$, where the vector of probabilities of being in any particular given state is independent of the initial starting distribution. The stationary distribution satisfies

$$\pi^* = \pi^*P \quad (A3.9)$$

In other words, $\pi^*$ is the left eigenvalue associated with the eigenvalue $\lambda = 1$ of $P$. The conditions for a stationary distribution is that the chain is irreducible and aperiodic. When a chain is periodic, it can cycle in a deterministic fashion between states and hence never settles down to a stationary distribution (in effect, this cycling is the stationary distribution for this chain). A little thought will show that if $P$ has no eigenvalues equal to $-1$ that it is aperiodic.
A sufficient condition for a unique stationary distribution is that the **detailed balance equation** holds (for all \(i\) and \(j\)),

\[
P(j \rightarrow k) \pi_j^* = P(k \rightarrow j) \pi_k^*
\]

or if you prefer the notation

\[
P(j, k) \pi_j^* = P(k, j) \pi_k^*
\]

i.e., transitions from \(j \rightarrow k\) balance \(k \rightarrow j\)

**Key for MCMC:** One can use the mathematical form for any complex distribution to construct transition probabilities for a Markov chain whose stationary distributions are draws from the target distribution

Means we can start the chain anywhere (any initial value) and if we run it long enough (**burn in** the chain), we approach stationarity and hence draws from the target
Metropolis-Hastings

The first MCMC approach was the Metropolis-Hastings algorithm

Basic idea: generate a number and either accept or reject that number based on a function that depends on the mathematical form of the distribution we are sampling from

LW Appendix 2 shows that this generates a Markov Chain whose stationary values correspond to draws From the target distribution

MH always works, but can be VERY slow, as most values might be rejected
Suppose our goal is to draw samples from some distribution \( p(\theta) \) where \( p(\theta) = f(\theta)/K \), where the normalizing constant \( K \) may not be known, and very difficult to compute. The **Metropolis algorithm** ([Metropolis and Ulam 1949, Metropolis et al. 1953]) generates a sequence of draws from this distribution as follows:

1. Start with any initial value \( \theta_0 \) satisfying \( f(\theta_0) > 0 \).

2. Using current \( \theta \) value, sample a **candidate point** \( \theta^* \) from some **jumping distribution** \( q(\theta_1, \theta_2) \), which is the probability of returning a value of \( \theta_2 \) given a previous value of \( \theta_1 \). This distribution is also referred to as the **proposal** or **candidate-generating distribution**. The only restriction on the jump density in the Metropolis algorithm is that it is symmetric, i.e., \( q(\theta_1, \theta_2) = q(\theta_2, \theta_1) \).

3. Given the candidate point \( \theta^* \), calculate the ratio of the density at the candidate (\( \theta^* \)) and current (\( \theta_{t-1} \)) points,

\[
\alpha = \frac{p(\theta^*)}{p(\theta_{t-1})} = \frac{f(\theta^*)}{f(\theta_{t-1})}
\]

Notice that because we are considering the ratio of \( p(x) \) under two different values, the normalizing constant \( K \) cancels out.

4. If the jump increases the density (\( \alpha > 1 \)), accept the candidate point (set \( \theta_t = \theta^* \)) and return to step 2. If the jump decreases the density (\( \alpha < 1 \)), then with probability \( \alpha \) accept the candidate point, else reject it and return to step 2.
We can summarize the Metropolis sampling as first computing

$$
\alpha = \min \left( \frac{f(\theta^*)}{f(\theta_{t-1})}, 1 \right)
$$

(A3.12)

and then accepting a candidate point with probability \( \alpha \) (the **probability of a move**). This generates a Markov chain \( (\theta_0, \theta_1, \ldots, \theta_k, \ldots) \), as the transition probabilities from \( \theta_t \) to \( \theta_{t+1} \) depends only on \( \theta_t \) and not \( (\theta_0, \ldots, \theta_{t-1}) \). Following a sufficient **burn-in period** (of, say, \( k \) steps), the chain approaches its stationary distribution and (as we will demonstrate shortly), samples from the vector \( (\theta_{k+1}, \ldots, \theta_{k+n}) \) are samples from \( p(x) \).

Hastings (1970) generalized the Metropolis algorithm by using an arbitrary transition probability function \( q(\theta_1, \theta_2) = \Pr(\theta_1 \rightarrow \theta_2) \), and setting the acceptance probability for a candidate point as

$$
\alpha = \min \left( \frac{f(\theta^*) \cdot q(\theta^*, \theta_{t-1})}{f(\theta_{t-1}) \cdot q(\theta_{t-1}, \theta^*)}, 1 \right)
$$

(A3.13)

This is the **Metropolis-Hastings algorithm**. Assuming that the proposal distribution is symmetric, i.e., \( q(x, y) = q(y, x) \), recovers the original Metropolis algorithm.
Example A3.2  Consider the scaled inverse-$\chi^2$ distribution,

\[ p(\theta) = C \cdot \theta^{-n/2} \cdot \exp \left( \frac{-a}{2\theta} \right) \]

We wish to simulate draws from this distribution with \( n = 5 \) degrees of freedom and scaling factor \( a = 4 \), giving \( f(x) = x^{-5/2} \exp[-4/(2x)] \), using the Metropolis algorithm.

Suppose we take as our candidate-generating distribution a uniform distribution on (say) \( (0, 100) \). Clearly, there is probability mass above 100 for the scaled inverse-$\chi^2$, but we assume this is sufficiently small so that we can ignore it. Now let's run the algorithm. Take \( \theta_0 = 1 \) as our starting value, and suppose the uniform returns a candidate value of \( \theta^* = 39.82 \). Computing \( \alpha \),

\[ \alpha = \min \left( \frac{f(\theta^*)}{f(\theta_{t-1})}, 1 \right) = \min \left( \frac{(39.82)^{-2.5} \cdot \exp(-2/39.82)}{(1)^{-2.5} \cdot \exp(-2/2 \cdot 1)}, 1 \right) = 0.0007 \]

Since (this case) \( \alpha < 1 \), \( \theta^* \) is accepted with probability 0.007. Thus, we randomly draw \( U \) from a uniform \( (0, 1) \) and accept \( \theta^* \) if \( U \leq \alpha \). In this case, the candidate is rejected, and we draw another candidate value from the proposal distribution (which turns out to be 71.36) and continue as above. The resulting first 500 values of \( \theta \) are plotted below.
Notice that there are long flat periods (corresponding to all $\theta^*$ values being rejected). Such a chain is called **poorly mixing**.

In contrast, suppose we use as our proposal distribution a $\chi^2_1$. Here, the candidate distribution is no longer symmetric, and we must employ Metropolis-Hastings (see Example A3.3 for the details). In this case, a resulting Metropolis-Hastings sampling run is shown below. Note that the time series looks like **white noise**, and the chain is said to be **well mixing**.
Burning-in the Sampler

A key issue in the successful implementation of Metropolis-Hastings, or any other MCMC sampler, is the number of runs (steps) until the chain approaches stationarity (the length of the burn-in period). Typically the first 1000 to 5000 values of the chain are thrown out, and then various convergence tests (see below) are used to assess whether stationarity has indeed been reached.

A poor choice of starting values and/or proposal distribution can greatly increase the required burn-in time, and an area of much current research is whether an optimal starting point and proposal distribution can be found. For now, we simply offer some basic rules. One suggestion for a starting value is to start the chain as close to the center of the distribution as possible, for example taking a value close to the distribution’s mode (such as using an approximate MLE as the starting value).

A chain is said to be poorly mixing if it stays in small regions of the parameter space for long periods of time, as opposed to a well mixing chain that seems to happily explore the space. A poorly mixing chain can arise because the target distribution is multimodal and our choice of starting values traps us near one of the modes (such as multimodal posteriors can arise if we have a strong prior in conflict with the observed data). Two approaches have been suggested for situations where the target distribution may have multiple peaks. The most straightforward is to use multiple highly dispersed initial values to start several different chains (Gelman and Rubin 1992). A less obvious approach is to use simulated annealing on a single-chain.
Simulated Annealing

Simulated annealing was developed as an approach for finding the maximum of complex functions with multiple peaks where standard hill-climbing approaches may trap the algorithm at a less than optimal peak. The idea is that when we initially start sampling the space, we will accept a reasonable probability of a down-hill move in order to explore the entire space. As the process proceeds, we decrease the probability of such down-hill moves. The analogy (and hence the term) is the annealing of a crystal as temperate decreases — initially there is a lot of movement, which gets smaller and smaller as the temperature cools. Simulated annealing is very closely related to Metropolis sampling, differing only in that the probability $\alpha$ of a move is given by

$$
\alpha_{SA} = \min \left[ 1, \left( \frac{p(\theta^*)}{p(\theta_{t-1})} \right)^{1/T(t)} \right]
$$

(A3.15a)

where the function $T(t)$ is called the \textbf{cooling schedule} (setting $T = 1$ recovers Metropolis sampling), and the particular value of $T$ at any point in the chain is called the \textbf{temperature}.
Convergence

- Two issues: convergence vs. sampling low-density regions of the posterior
  - Convergence: worry that you have sampled the chain for a sufficiently long period that draws are indeed from the stationary distribution
  - Adequate sampling of low density regions: Even if convergence has been reached, need to run the sampler long enough to sample important (i.e. extreme), but low-density regions of the posterior.

- WL Appendix 2 details some of the diagnostic tools that can be used. However, none perfect
- Basic idea: either compare subsamples of the chain or different chains to access consistency
MH and Gibbs sampler

- While MH works, as mentioned it can be very slow in that most proposal values can be rejected, so that (say) 500 iterations of the MH sampler may yield only one or two values.
- The Gibbs sampler is a version where ALL draws are accepted.
- Key: all conditional distributions must have a nice form (conditionally conjugate likelihoods).
The **Gibbs sampler** (introduced in the context of image processing by Geman and Geman 1984), is a special case of Metropolis-Hastings sampling wherein the random value is always accepted (i.e. $\alpha = 1$). The task remains to specify how to construct a Markov Chain whose values converge to the target distribution. The key to the Gibbs sampler is that one only considers **univariate** conditional distributions — the distribution when all of the random variables but one are assigned fixed values. Such conditional distributions are far easier to simulate than complex joint distributions and usually have simple forms (often being normals, inverse $\chi^2$, or other common prior distributions). Thus, one simulates $n$ random variables sequentially from the $n$ univariate conditionals rather than generating a single $n$-dimensional vector in a single pass using the full joint distribution.

To introduce the Gibbs sampler, consider a bivariate random variable $(x, y)$, and suppose we wish to compute one or both marginals, $p(x)$ and $p(y)$. The idea behind the sampler is that it is far easier to consider a sequence of conditional distributions, $p(x \mid y)$ and $p(y \mid x)$, than it is to obtain the marginal by integration of the joint density $p(x, y)$, e.g., $p(x) = \int p(x, y) dy$. The sampler starts with some initial value $y_0$ for $y$ and obtains $x_0$ by generating a random variable from the conditional distribution $p(x \mid y = y_0)$. The sampler then uses $x_0$ to generate a new value of $y_1$, drawing from the conditional distribution based on the value $x_0$, $p(y \mid x = x_0)$. The sampler proceeds as follows

$$x_i \sim p(x \mid y = y_{i-1})$$  \hspace{1cm} (A3.18a)

$$y_i \sim p(y \mid x = x_i)$$  \hspace{1cm} (A3.18b)
Example A3.4. Consider the following distribution from Casella and George (1992). Suppose
the joint distribution of \( x = 0, 1, \ldots, n \) and \( 0 \leq y \leq 1 \) is given by

\[
p(x, y) = \frac{n!}{(n-x)!x!} y^{x+a-1} (1 - y)^{n-x+b-1}
\]

Note that \( x \) is discrete and \( y \) continuous. While the joint density is complex, the conditional
densities are simple distributions. To see this, first recall that a binomial random variable \( z \)
has a density proportional to

\[
p(z \mid q, n) \propto \frac{q^z(1-q)^{n-z}}{z!(n-z)!} \quad \text{for} \quad 0 \leq z \leq n
\]

where \( 0 < q < 1 \) is the success parameter and \( n \) the number of traits, and we denote
\( z \sim \text{B}(n, p) \). Likewise recall the density for \( z \sim \text{Beta}(a, b) \), a beta distribution with shape
parameters \( a \) and \( b \) is given by

\[
p(z \mid a, b) \propto z^{a-1}(1 - z)^{b-1} \quad \text{for} \quad 0 \leq z \leq 1
\]

Observe that the conditional distribution of \( x \) (treating \( y \) as a fixed constant) is \( x \mid y \sim \text{B}(n, y) \),
while \( y \mid x \sim \text{Beta}(x + \alpha, n - x + \beta) \).
The power of the Gibbs sampler is that by computing a sequence of these univariate conditional random variables (a binomial and then a beta) we can compute any feature of either marginal distribution. Suppose \( n = 10 \) and \( \alpha = 1, \beta = 2 \). Start the sampler with (say) \( y_0 = 1/2 \), and we will take the sampler through three full iterations.

(i) \( x_0 \) is obtained by generating a random \( B(n, y_0) = B(10, 1/2) \) random variable, giving \( x_0 = 5 \) in our simulation.

(ii) \( y_1 \) is obtained from a \( \text{Beta}(x_0 + \alpha, n - x_0 + \beta) = \text{Beta}(5 + 1, 10 - 5 + 2) \) random variable, giving \( y_1 = 0.33 \).

(iii) \( x_1 \) is a realization of a \( B(n, y_1) = B(10, 0.33) \) random variable, giving \( x_1 = 3 \).

(iv) \( y_2 \) is obtained from a \( \text{Beta}(x_1 + \alpha, n - x_1 + \beta) = \text{Beta}(3 + 1, 10 - 3 + 2) \) random variable, giving \( y_2 = 0.56 \).

(v) \( x_2 \) is obtained from a \( B(n, y_2) = B(10, 0.56) \) random variable, giving \( x_2 = 0.7 \).

Our particular realization of the Gibbs sequence after three iterations is thus \((5, 0.5), (3, 0.33), (7, 0.56)\). We can continue this process to generate a chain of the desired length. Obviously, the initial values in the chain are highly dependent upon the \( y_0 \) value chosen to start the chain. This dependence decays as the sequence length increases and so we typically start recording the sequence only after a sufficient number of burn-in iterations have occurred.

Note that the sample of (say) the \( x_i \) are draws from the marginal posterior for \( x \)
ABC

• **Approximate Bayesian Computation** (or ABC) is becoming more widespread, esp. in population genetics

• The idea is that the likelihood may be sufficiently complex that it is easier to simulate it (given a set of starting parameters).

• For example, consider a pop gen model for genetic variation with unknown parameters for mutation rate, pop size before a bottleneck, size after a bottleneck, and time in the past that the bottleneck occurred

• The data are measures of molecular diversity (number of segregating sites, pairwise differences)
ABC (cont)

• Under ABC, one first draws values for these parameters from some prior
• These are used to run a simulation, which returns the observed data
• If the simulation run matches the observed values, this vector of parameter values is saved. If no match, a new set is drawn
• The resulting collection of saved (not rejected) parameter values forms the posterior
One significant issue is how close is close -- with discrete data, usually want exact matches, with continuous data want the values to agree within some very small distance metric $e$.

The tradeoff is much more CPU time for closer/exact matches vs. less precision if the match is not exact.