Applied Bayesian Analysis

Introduction to MCMC

Monte Carlo

Monte Carlo methods

- Integrate or sample from a function or distribution
- Ultimately, evaluate characteristics of a posterior distribution
- Generally refers to simulation techniques
- For most methods we apply, the distribution need not be standardized
- Sampling randomly allows us to apply statistics to interpret the results when the analytical work is too complex

Monte Carlo integration problem

Consider the generic problem of evaluating an integral of the following form:

$$\mathfrak{I} = \int_R h(x)\ f(x)\ dx\ = \mathbb{E}_f[h(X)]$$

where x and R are uni- or multidimensional, f is a distribution that can be expressed in a closed form, and h is a function

The phrase f is a closed form means f can written out as an expression

Monte Carlo Principle

Use a sample $(x_1, \ldots, x_m)$ from the density f to approximate the integral $\mathfrak{I}$ by the empirical average

$$\overline{h}_m = \frac{1}{m} \sum_{j=1}^{m} h(x_j)$$

Under some regularity conditions, the average will converge to the integral,

$$\overline{h}_m \longrightarrow \mathbb{E}_f[h(X)],$$

by the Strong Law of Large Numbers

Rejection sampling

Definition

- Plot the distribution (PDF), which we will assume is univariate for now and might not be standardized
- We first simulate a random variable from a different (easy) distribution
- Using an algorithm, we determine whether to accept or reject the random variable
- At the end of the simulation, use only random variables that we have accepted

The horizontal location of all accepted points represent a random draw from the distribution
If $k = f(x_1)/f(x_2)$, then we should be k times more likely to observe x_1 than x_2

Example– Beta(2,2) distribution

The distribution

- $f(x) \propto x(1-x)$ for $x \in [0,1]$, and zero otherwise
- We will plot $x(1-x)$
- We will use a uniform distribution to simulate random draws

Procedure

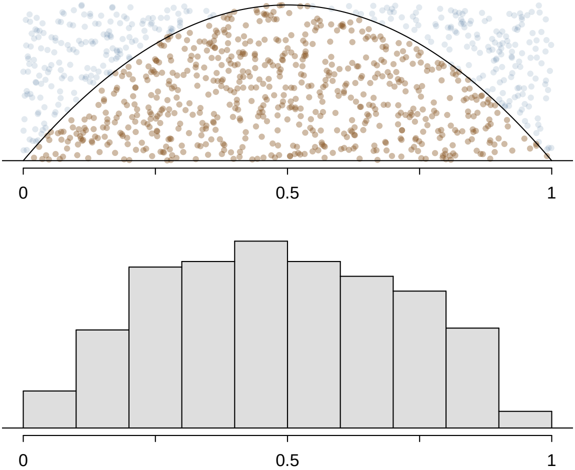
- Throw darts in $[0,1] \times [0,0.25]$
- Our darts generator:

```
x    <- runif(1000, 0, 1)     uniform 0 to 1
y    <- runif(1000, 0, 0.25)  height from 0 to 0.25
```

- Check if accept or reject:

```
acc <- (x > 0 & x < 1) & (y > 0 & y < x*(1-x))
hist(x[acc])
```

Darts simulation – Beta(2,2) distribution



The point of sampling

Obtaining observations, for a purpose

- When we sample from a distribution, we can study characteristics of the distribution by examining the sample
- If the distribution sampled from is a posterior density, then the posterior mean, variance, median, and other summaries can be accurately estimated

A bridge to

- Markov chains
- Metropolis algorithm (our first MCMC algorithm)
- Algorithm convergence considerations (in part)

Markov Chain Monte Carlo Methods

Complexity of most models encountered in Bayesian modeling

Standard simulation methods not good enough a solution

New technique at the core of Bayesian computing, based on *Markov chains*

Algorithms based on Markov chains

Idea: simulate from a posterior density $\pi(\cdot|x)$ [or any density] by producing a Markov chain

$$(\theta^{(t)})_{t\in\mathbb{N}}$$

whose stationary distribution is

$$\pi(\cdot|x)$$ posterior

Translation

sample enough time markov chain

For t large enough, $\theta^{(t)}$ is approximately distributed from $\pi(\theta|x)$, no matter what the starting value $\theta^{(0)}$ is *[Ergodicity]*.

Markov chains

A random quantity that changes in sequence (x_1, x_2, ...) such that

- The possible values of x are called "states"
- Movement to the next state only depends on the current state

$$P(x_{n+1} = x | x_n, x_{n-1}, ..., x_1) = P(x_{n+1} = x | x_n)$$

- Example: In many board games, player behavior only relies on the current state
- Markov chains can exist in a discrete or continuous space

Why a detour to Markov chains is helpful

- Foundational ideas form a basis for building MCMC methods, especially detailed balance
- Simulate from a Markov chain that moves towards $P(\theta|y)$

Example

Five node system

- States: 1 through 5

$$k-1 \overset{0.4}{\rightarrow} k \qquad P(k|k-1) = 0.4 \text{ for } k = 2,3,4,5$$

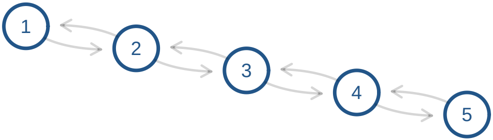
$$k-1 \overset{0.6}{\leftarrow} k \qquad P(k-1|k) = 0.6 \text{ for } k = 2,3,4,5$$

$$1 \overset{0.6}{\rightarrow} 1 \qquad P(1|1) = 0.6$$

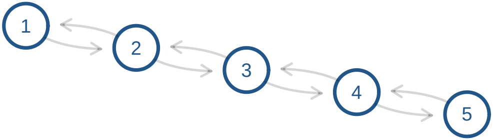
Probability of staying 1 given at state 1 is 0.6

$$5 \overset{0.4}{\rightarrow} 5 \qquad P(5|5) = 0.4$$

Probability of staying 5 given at state 5 is 0.4



Stationary Distribution



If $x_1 = 1$, estimate $P(x_{1000} = 1)$
Should it matter much if $x_1 = 5$?

The stationary distribution of a Markov chain is the limiting distribution of $P(x_k = i)$

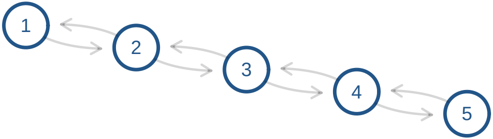
- For k large,

$$P(x_k = i) \approx P(x_{k+1} = i)$$

Example: detailed balance condition

One way to think about this problem

- We put a large number of marbles in each state, and each marble moves independently of the others and according to the system probabilities
- Eventually, the marbles will distribute through the system and reach some stationary distribution



Example: detailed balance condition

The number of marbles moving from any state i to $i+1$ must be about the same as the number moving from $i+1$ to i

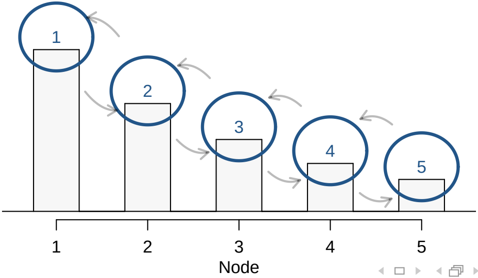
- If it isn't, then the system hasn't "settled"
- This realization provides a detailed balance equation:

$$P(i)T(i,i+1) = P(i+1)T(i+1,i)$$

where $T(i,j)$ is the transition probability:

$$T(i,j) = P(x_{k+1} = j | x_k = i)$$

probability of move
from a to b is the
same
as the prob of moving
from b to a back

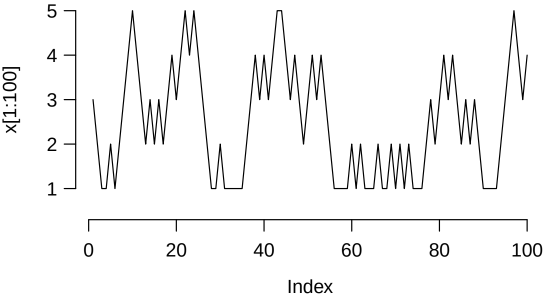


Example – simulation

Could use detailed balance to solve for the stationary distribution directly

Or, Initialize and iterate

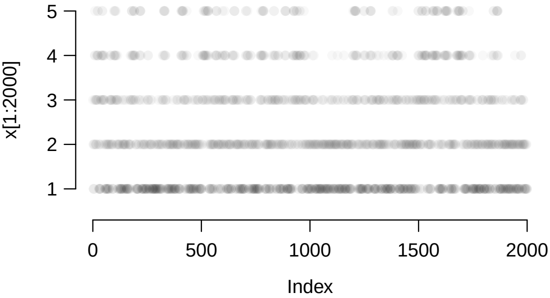
- Initialize as $x_1 = 1$
- Use the previously defined transition probabilities



Example: simulation

Simulation of 2000 points

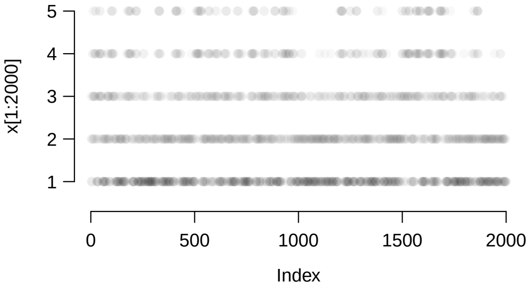
- Each point after some “burn in” is like a sample from the stationary distribution
- But what issue exists within these sample draws from the stationary distribution?



Correlation of successive draws

In the simulation, there is a correlation between x_k and x_{k+1}

- There is also a correlation between x_k and x_m for any k and m, but $Cor(x_k, x_m)$ tends to be small when $|k-m|$ is large if the chain has certain properties
- This autocorrelation is a characteristic we will regularly encounter



Detailed balance condition

Detailed balance is a special condition

- Generally, Markov chains do not satisfy detailed balance
- Those Markov chains where it does hold are called reversible
- The most commonly used MCMC techniques are built directly from the detailed balance condition
- If i and j are in the state space, then the general formula for detailed balance is

$$P(i)\, T(i,j) = P(j)\, T(j,i)$$

Markov chains

Movements in continuous space

- Markov chains can be built over continuous state spaces using something like transition probabilities
- If a space has an infinite number of states, we can be certain we cannot visit them all
- The goal is not to visit every state
- Generally we can be satisfied with taking a sample of the underlying distribution and analyzing the sample

Transitioning to MCMC

Markov Chain Monte Carlo

- We are given the stationary distribution, which is typically a posterior $P(\theta|y)$
- Our goal is to construct a Markov chain that samples from $P(\theta|y)$
- MCMC methods define ways to create a proper Markov chain where the observations are from the stationary distribution

MCMC methods exploit detailed balance

- In simple five-state example, we started with transition probabilities and searched for a stationary distribution
- In MCMC, instead of using the transition probabilities to find the stationary distribution, we use the stationary distribution to define transition probabilities

the opposite way (know the stationary distribution (posterior) => transition prob)

Metropolis algorithm

Mission

- Take sample from a possibly difficult to characterize distribution, in our application this will always be a posterior distribution
- The distribution we want to sample from is called the target distribution Posterior distribution

Remarks

- Earlier the x_k moved around in discrete spaces using transition probabilities
- Sometimes the x_k would stay in one spot for a time unit, i.e. $x_k = x_{k+1}$
- We need to define how the x_k move so that, when we look at the observations, it looks like they were sampled from the target distribution

Metropolis algorithm

Strategy

- Create a proposal function f_p that defines how we try to move around in the support space (where $f_p \neq 0$)
- Based on the current location x_k, propose moving to a new location y
- Use the detailed balance equation to determine the acceptance probability (R) of the move
  - Move to $x_{k+1} = y$ with probability R
  - Stay put at $x_{k+1} = x_k$ with probability $1 - R$

probability of don't move is 1-R

Metropolis algorithm – proposal function

Proposal function

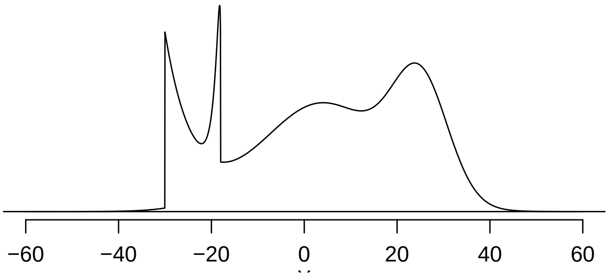
- For the Metropolis algorithm, we choose f_p to be symmetric
- Proposal is jump from current location: $y = x_k + u_{k+1}$, where $u_{k+1} \sim f_p$
- Example: $u_{k+1} \sim U(-1, 1)$
- Example: $u_{k+1} \sim N(0, \sigma^2 = 0.25)$
- The probability of proposing w when at z is the same as the probability of proposing z when at w <=> same

The acceptance probability (R) is chosen to ensure detailed balance is maintained

Example

Target distribution g

- g is an odd and unclassified distribution
- g is not standardized, and finding the proper standardization constant may be difficult
- A brute force approach would probably work in one-dimension
- That luxury won't typically be available in multiple dimensions



Example

What we are given

- A target distribution, g posterior distribution (fix and known)
- We initialize the Markov chain at $x_1 = 0$ (the choice of 0 is arbitrary) at the middle

Proposal function

- We propose moving to a new location around the current location:

$$y \sim U(x_1 - 10, x_1 + 10)$$

Example

What we are given

- A target distribution, g
- We initialize the Markov chain at $x_1 = 0$ (the choice of 0 is arbitrary)

Proposal function

- We propose moving to a new location around the current location:

$$y \sim U(x_1 - 10, x_1 + 10)$$

Here the proposal function f_p is uniform on $(-10, 10)$

Identifying x_2

- Either move to y ($x_2 = y$) or stay at zero ($x_2 = x_1$)
- Use detailed balance to determine whether we accept or not

Example: transition probability

Identifying a proper transition probability

- The unknowns: $T(0, y)$ and $T(y, 0)$
- Detailed balance:

$$g(0)\, T(0, y) = g(y)\, T(y, 0)$$

- The ratio of the transition probabilities is fixed:

$$\frac{T(0, y)}{T(y, 0)} = \frac{g(y)}{g(0)}$$

but we can control
T(0, y) / T(y, 0)

Example: transition probability

Determining $T(0, y)$ and $T(y, 0)$:

- The ratio of the transition probabilities is fixed:

$$\frac{T(0, y)}{T(y, 0)} = \frac{g(y)}{g(0)}$$

- Typically this ratio is not one, meaning either $T(0, y) > T(y, 0)$ or vice-versa
- Whichever is bigger, we will define as 1 set T(y, 0) = 1, Then T(0, y) = g(y)/g(0)
- With the larger transition probability set at 1, the other transition probability is determined by the equation above

Setting the larger probability to 1 maximizes how often transitions can occur when using a particular f_p

Example: transition probability

Determining $T(0, y)$ and $T(y, 0)$:

- If this right ratio is less than 1, define $T(y, 0) = 1$, which implies

$$T(0, y) = \frac{g(y)}{g(0)} T(y, 0) = \frac{g(y)}{g(0)}$$

- Else the ratio is greater than 1, so define $T(0, y) = 1$, which implies

$$T(y, 0) = \frac{g(0)}{g(y)} T(0, y) = \frac{g(0)}{g(y)}$$

- In summary, * Important is here:

$$T(0, y) = \min\left\{\frac{g(y)}{g(0)}, 1\right\}$$

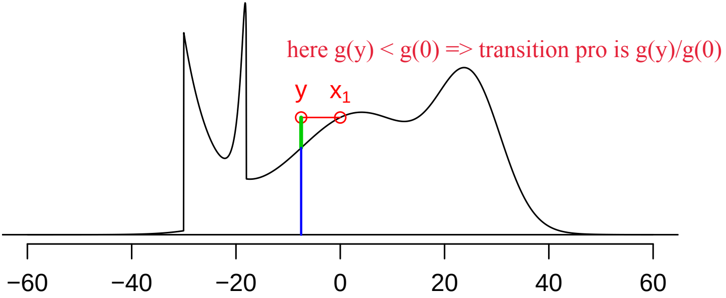
The acceptance probability is chosen to ensure detailed balance is satisfied

Example: transition probability

Summarizing the transition probability (example with $y = -7.5$)

- Move to the proposed point y from 0 with probability

$$\min\left\{\frac{g(y)}{g(0)}, 1\right\}$$

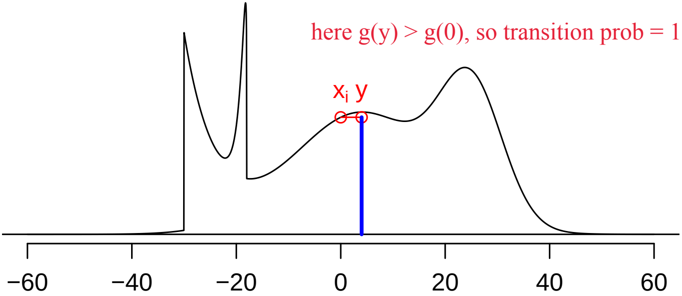


Example: transition probability

Summarizing the transition probability (example with $y = 3.5$)

- Move to the proposed point y from 0 with probability

$$\min\left\{\frac{g(y)}{g(0)}, 1\right\}$$



Example: transition probability

Generally

- If at location x_k, propose a new location y using the proposal function
- Set $x_{k+1} = y$ with probability

$$\min\{R, 1\}$$

where $R = \frac{g(x_{k+1})}{g(x_k)}$

Typically a uniform random variable on $(0, 1)$ is generated and compared to R to determine if the move is accepted for each k

- Generate a new random uniform variable for each proposed transition

Example: coding

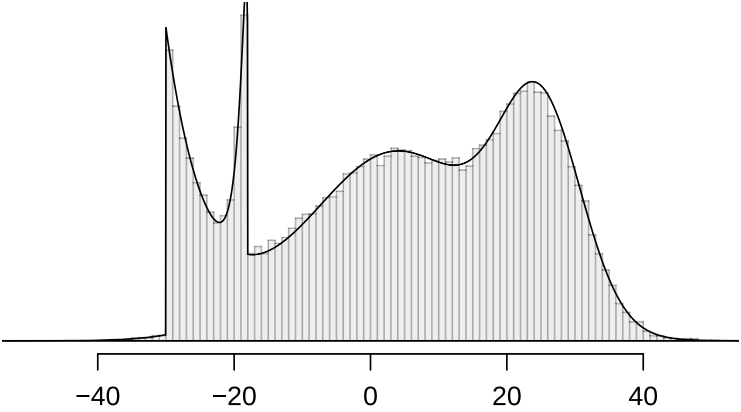
```
n   <- 10^5
x   <- 0   sample
acc <- 0   how many time we accecpt it

for(i in 2:n){                  previous location x[i-1]
propose a new value
  xNew <- runif(1, x[i-1]-10, x[i-1]+10)
  R    <- g(xNew)/g(x[i-1])  ratio      g(x) is the value of the wired
                                        distribution at x
  if(runif(1) < R){
    x[i] <- xNew  accept the new move
    acc  <- acc + 1
  } else {
    x[i] <- x[i-1]   stay at the same spot
  }
}

acc/n
```

Example: simulation results

- Moves were accepted about 80% of the time in this simulation



Example: two dimensions

Target distribution

- g is a function in $\mathbb{R}^2$
- Not standardized and it is unknown if it is unimodal

Consideration

- The transition can be based on a one variable at a time or moves can be proposed in both variables simultaneously

Example: two proposal functions

One variable at a time

- Propose a change in x, then propose a change in y
- Repeat, moving only in one dimension at a time

Both variables simultaneously

- Could try a uniform proposal distribution around (x_k, y_k)
- A multivariate normal model is another approach

Example: pseudocode for simultaneous movement

Initialize

- Specify a number of n sample points, and set $x_1 = y_1 = 0$

For $n-1$ iterations in a loop

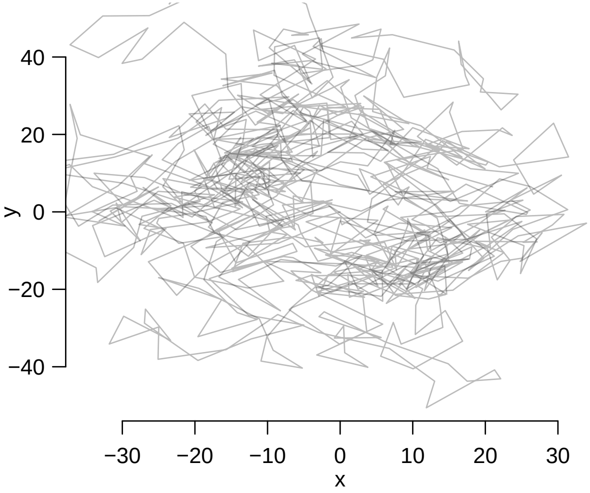
- Propose a new location, picking x_{new} and y_{new} according to a proposal function
- Identify the ratio $R = g(x_{new}, y_{new})/g(x_k, y_k)$ similar to previous ratio
- Accept the new location if a random uniform on $(0,1)$ is less than R
- Otherwise, reject and set $(x_{k+1}, y_{k+1}) = (x_k, y_k)$

Summarize

- Look at samples and diagnostics

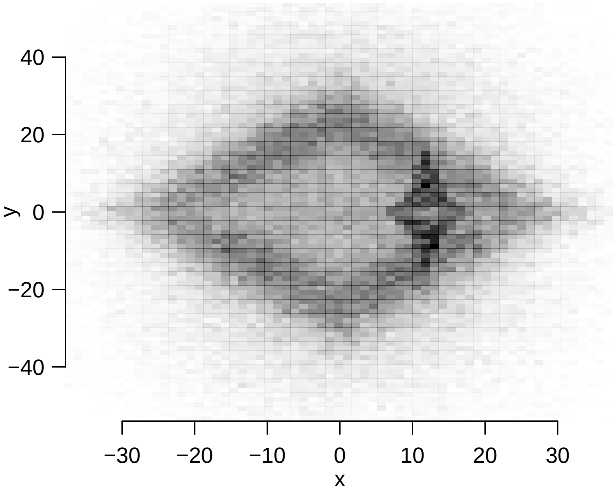
Example: results

The first 1000 observations (trace plot)



Example: results

Summary of 200,000 observations, 78% of moves accepted



Questions to consider

- Are observations from the Metropolis algorithm i.i.d.?
  not i.i.d. There are correlation between the previous one and the current
- The first observation was not random... is this a problem?
  It converges to the first observation location; but may be problematic if you choose wired location, though we will reach them at a long enough time
- Is it problematic that the acceptance rate was so low/high?
  If too low: less efficient, it will take much longer to move somewhere
  If too high: such as 98%-99%, go everywhere; generally: 20-60%, not crazy low or crazy high
- What changes could be tried to attempt to improve the algorithm?
  not make jump too small, or too high; change the size the jump by changing the proposal function, but cannot change transition rate
- What does improve mean in this context?
  correctly and efficient
- Was the sample large enough? Would a smaller sample have been sufficient?
  Yes, in this case. There is no one single right answer. Look at the trace plot or autocorrelation; if autocorrelation super high, the effect sample size is lower than you expected

Convergence assessment

Question: How many iterations do we need to run???

Convergence assessment

Question: How many iterations do we need to run???

- **Rule # 1** There is no absolute number of simulations, i.e. $1,000$ is neither large, nor small.
- **Rule # 2** It takes [much] longer to check for convergence than for the chain itself to converge.
- **Rule # 3** MCMC is a *"what-you-get-is-what-you-see"* algorithm: it fails to tell about unexplored parts of the space.
- **Rule # 4** When in doubt, run MCMC chains in parallel and check for consistency.

Convergence assessment

Question: How many iterations do we need to run???

- **Rule # 1** There is no absolute number of simulations, i.e. $1,000$ is neither large, nor small.
- **Rule # 2** It takes [much] longer to check for convergence than for the chain itself to converge.
- **Rule # 3** MCMC is a *"what-you-get-is-what-you-see"* algorithm: it fails to tell about unexplored parts of the space.
- **Rule # 4** When in doubt, run MCMC chains in parallel and check for consistency. Hope they are overlapped (want to them to go together); if they are not consistent, you may not trust either of them

Many "quick-&-dirty" solutions in the literature, but not necessarily trustworthy.

Examining the proposal function

Proposal function

- The proposal function was chosen (somewhat) arbitrarily
- This function must be symmetric around the current location x_k for the Metropolis algorithm

What can we change? Why would we change each of these?

- The distribution itself
  - Generally we want to use continuous proposal distributions for continuous target distributions
- The variability of the distribution
  - What are the pros/cons if the variability was made smaller/larger?

    it is about the size of your jump;

Examining the proposal function

More variability

- Improve opportunity to travel around much of the distribution in only a few moves
- May propose a jump "out of the distribution" where we are unlikely to accept the jump
- Can be useful in finding hidden features

Less variability

may most time stay at the nearby space; but likely to accept the jump

- Basically the opposite of above
- Note that small moves are rarely rejected since $g(x) \approx g(x + \epsilon)$ when ϵ is small (for many functions)

Which is better: to accept nearly all jumps or to try for big jumps?

- A happy balance works best

How we define "best"

Properties of the Markov chain

- The observations $x_1, x_2, \ldots$ are *not* independent
- Observations close in the sequence are generally correlated
- A good MCMC sample tends to keep this correlation low (there are also other criteria)
- We can measure these correlations to provide a helpful guide to the mixing of the the Markov chain

Autocorrelation function (ACF)

Auto-correlation function

- Consider the correlation between observations i steps (lags) apart in the sequence:

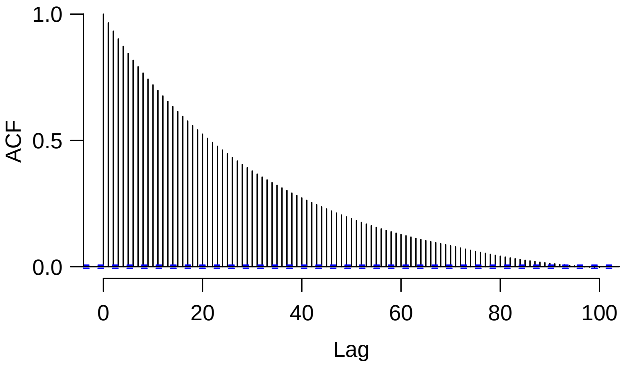
$$c_i = Cor(x_k, x_{k+i}), \quad k \text{ large}$$

- These correlations – $c_1, c_2, \ldots$ – are useful in describing how our sequence is moving around in the distribution
- This is a special case of the autocorrelation function (ACF)
- The general case would not assume c_i is constant for k, but this approximation is good when k is large

Autocorrelation function

We compute a sample ACF (from Example 1)

- The ACF below is equal to zero at about lag 100



Starting value

In our Metropolis algorithm, a starting value is assigned

- What is an "optimal" starting value?
- Is such a thing relevant?

Starting value

In our Metropolis algorithm, a starting value is assigned

- What is an "optimal" starting value?
- Is such a thing relevant?

One simple solution:

- Drop out the first "many" runs, where "many" might correspond to the number of lags it takes for the ACF to reach zero several times over
- e.g. $10 * 100$ lags $\rightarrow$ drop x_1, x_2, ..., x_{1000} for Example 1
- The set of initial runs that are dropped is called the burn in
- Would it be bad if we didn't remove a burn in?

cut out the steps takes you to converge
sometime, if you include the burn in, you might get the incorrect inference; generally, people usually burn in

Recap of convergence considerations

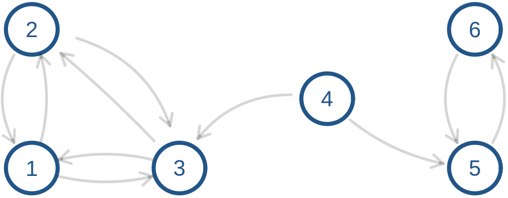
Choice of a distribution

- We must choose a symmetric proposal function for the Metropolis algorithm, where the normal distribution and scaled t distributions are common choices
- A chosen variance of the function should balance rejecting proposed values with attempts at large jumps in the Markov chain

Recap of convergence considerations

Starting value

- Ideally, the starting value will not affect the Markov chain after the burn in
- However, problems can arise in special situations:



- What problems might we encounter?
- More convergence considerations will be discussed next lecture, along with two more MCMC algorithms

Metropolis-Hastings

Similarities with Metropolis

- Same ultimate goal: sample from the target distribution
- We continue to propose and accept moves around the target distribution

Differences

- The proposal function may not be symmetric
- Because the proposal function is not symmetric, the acceptance probability needs a little help adjusting

Remark

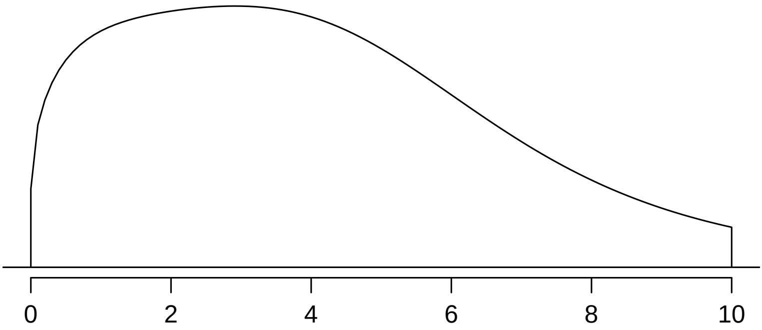
- The Metropolis algorithm is a special case of the Metropolis-Hastings algorithm

Example

Simple case: proposal function is a uniform distribution

- The target distribution will be $g(x)$, shown below
- The support of this distribution is on $(0, 10)$

propose value at 12, density 0;



Example: Proposal function

Proposal function

- We may not want to propose impossible values for the distribution
- If x_i is between $[3, 7]$, use uniform proposal: $f_p(x) = U(x - 3, x + 3)$
- If $x_i < 3$, modify the uniform proposal: $f_p(x) = U(0, x + 3)$
- Similarly for $x_i > 7$, use: $f_p(x) = U(x - 3, 10)$

always be inside

Why this proposal function changes the game

- $P(y = 1|x_i = 3.5) = 1/6$
- $P(y = 3.5|x_i = 1) = 1/4$

non-symmetric; from 3.5 to 1 is different from from 1 to 3.5
now, they don't cancel out

- In all continuous cases, these are really densities, but for simplicity we call them probabilities

Example: Detailed balance

Rearranging, and defining $\widetilde{R}$

P(accept y | x) is similar to transition probability

$$\widetilde{R} = \frac{P(\text{accept } y|x)}{P(\text{accept } x|y)} = \frac{P(y)\ P(\text{propose } x|y)}{\underbrace{P(x)}_{\text{R1}}\ \underbrace{P(\text{propose } y|x)}_{\text{R2}}}$$

p(y)/p(x) is the target distribution

- We can compute the right side based solely on the proposal function and the target distribution

Acceptance probability

- If $\widetilde{R} < 1$, set $P(\text{accept } y|x) = \widetilde{R}$
- As before, if $\widetilde{R} \geq 1$, then accept y as a move from x
- Reasoning behind choice of the acceptance probability is identical to that described for the Metropolis algorithm

Example: Code

```
n   <- 5*10^4
x   <- rep(5, n)
acc <- 0

for(i in 2:n){
   y  <- fp(x[i-1])
   R1 <- g(y)/g(x[i-1])
   R2 <- fpDens(y, x[i-1])/fpDens(x[i-1], y)
   if(runif(1) < R1*R2){
      x[i] <- y
      acc  <- acc + 1
   } else {
      x[i] <- x[i-1]
   }
}
```

from posteriors

propose new value based on you start on the previous on

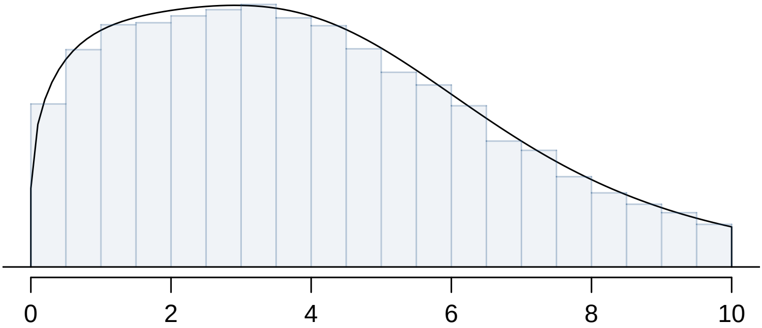
ratio of density

R1, R2 is the right side of equation in the previous slides

Example: simulation results

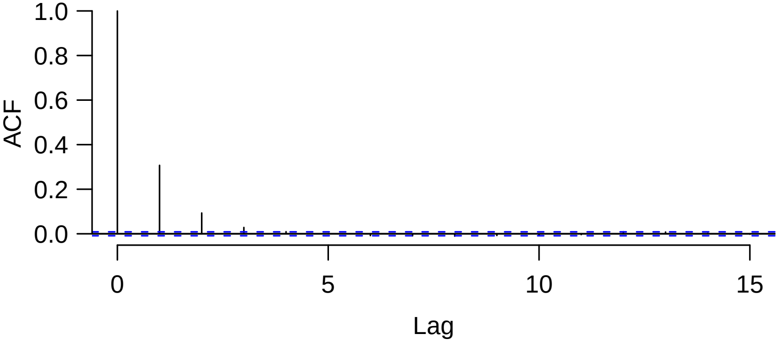
Metropolis-Hastings results

- Acceptance rate: 69%



Example: ACF

- When is the ACF approximately zero?
- What might be a reasonable burn in (according to the plot)?
- What other consideration might we consider in a burn in?



Framework for Metropolis-Hastings

Necessary for asymmetric proposal functions

A framework built on detailed balance

- $P(x)T(x,y) = P(y)T(y,x)$
- The transition probability from a location x to y is different

Acceptance probabilities chosen to ensure detailed balance

- The transition probability is broken down into two pieces:

$$T(x,y) = P(\text{propose } y|x) * P(\text{accept } y|x)$$

- Both the proposal probability (described by f_p) and the target density are known

Framework for Metropolis-Hastings

Transition probability, moving from x to y

- The new detailed balance equation:

$$\frac{T(x,y)}{T(y,x)} = \frac{P(\text{propose } y|x)P(\text{accept } y|x)}{P(\text{propose } x|y)P(\text{accept } x|y)} = \frac{P(y)}{P(x)}$$

- Moving the known quantities to the right side:

$$\widetilde{R} = \frac{P(\text{accept } y|x)}{P(\text{accept } x|y)} = \frac{P(\text{propose } x|y)}{P(\text{propose } y|x)}\frac{P(y)}{P(x)}$$

Acceptance probability

- We accept a proposed move to y from x with probability

$$\min\{\widetilde{R}, 1\}$$

Recall: ACF

Proposal value and starting location

- Adjusting the proposal function changes how the Markov chain moves through the distribution
- An acceptance rate close to 100% isn't ideal for Metropolis / M-H (are the jumps small or big in such a case?)
- Depending on how odd the target distribution is shaped, different starting values may result in meaningfully different sample distributions

What we are missing

- It is not clear if the Markov chain truly explores the entire space
- The chain can get trapped in one section of the distribution while the rest goes unexplored

Another consideration: exploration of the space

Mixing

- The mixing of the chain is how well it propagates through the distribution
- In lower dimensional spaces (especially 1-D and 2-D), it is relatively easy to visually inspect for mixing

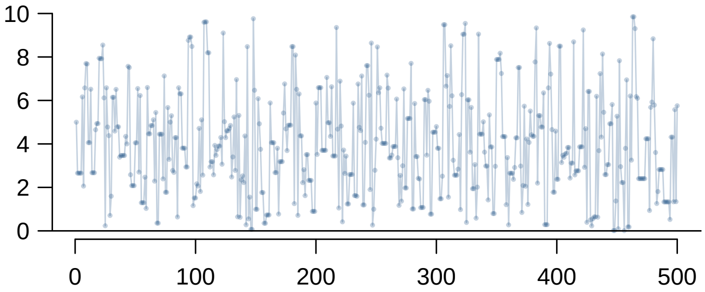
Trace plots

- Plotting the path of the Markov chain in a graph
- Visually inspect the propagation of a Markov chain in 1-D and 2-D by plotting the Markov chain over time

Trace plots

Using x from Example 1

```
> plot(x, type='l', col='#22558844')
> points(x, pch=20, col='#22558844', cex=0.5)
```



Mixing it up

What can be controlled

- The proposal function
- Starting values
- The structure of the algorithm, e.g. the order of parameter sampling, grouping parameters, etc.

How to apply these tools

- Vary the proposal function and starting values
- Use the ACF and trace plots as tools for evaluation
- Try a variety of starting values and check that they all converge to the same distribution

Remember: The MCMC procedure (including the proposal distribution) is not part of the *model*, it's just a tool for computing estimates!

Gibbs sampling

Sample situation

- Given a distribution $g(x, y)$
- Unable to sample observations (x, y) directly from g
- If it is easy to sample x from $g(x|y)$ and y from $g(y|x)$, then Gibbs sampling can be easily accomplished

General case

- In Gibbs sampling, we move around in a state space in one (or more) variables at a time, sampling from conditional distributions

Gibbs sampling algorithm

When Gibbs sampling is useful

- There is some overall distribution $g(\theta_1, \theta_2, ..., \theta_k|y)$ to be sampled from
- We are able to directly sample from the conditional distributions:
  - $g_1(\theta_1|\theta_2, \theta_3, \theta_4, ...\theta_k, y)$
  - $g_2(\theta_2|\theta_1, \theta_3, \theta_4, ...\theta_k, y)$
  - ...
- Occasionally, we may be able to sample multiple parameters simultaneously,
  e.g. $g_{i,j}(\theta_i, \theta_j|\theta_l$ where $l \notin \{i, j\}, y)$
- Sampling from joint conditionals is generally even better

Gibbs sampling algorithm

Initialize and iterations

- Initialize the parameters: $\theta^{(0)} = (\theta_1^{(0)}, \theta_2^{(0)}, ..., \theta_k^{(0)})$
- Sample $\theta_1^{(1)}$ from g_1, conditioned on the other most recent states of the θ_i (e.g. $\theta_2^{(0)}, ..., \theta_k^{(0)}$)
- Sample $\theta_2^{(1)}$ from g_2, conditioned on the other most recent states of the θ_i (we use $\theta_1^{(1)}$ and then the other variables in their initial state)
- etc.

Once we run through the list once, we run through again, and again, ...

We can also mix up the order through which we sample the parameters (e.g. sample θ_2, then θ_1, then θ_3, etc.)

Gibbs satisfies detailed balance

Examining transition probabilities

- Two points x and y differ only in dimension i: $x_{-i} = y_{-i}$, i.e. Gibbs proposal considers moves between x and y
- The transition probability/density from x to y:

$$g_i(y_i|y_{-i}) = \frac{g(y)}{g_{-i}(y_{-i})}$$

Detailed balance

- From above: $g_{-i}(y_{-i}) = \frac{g(y)}{g_i(y_i|y_{-i})}$
- Similarly: $g_{-i}(x_{-i}) = \frac{g(x)}{g_i(x_i|x_{-i})}$
- The marginal densities on "-i" are equal: $\frac{g(x)}{g_i(x_i|x_{-i})} = \frac{g(y)}{g_i(y_i|y_{-i})}$
- Equivalently:

$$g(x)\ g_i(y_i|y_{-i}) = g(y)\ g_i(x_i|x_{-i}) \ \rightarrow \ g(x) f_p(y|x) = g(y) f_p(x|y)$$

Example: uniform on a strip in $\mathbb{R}^2$

We want to sample observations from a uniform distribution

- $g(x, y) \propto 1$ if $x \in (-5, 5)$ *and* $y \in (3x - 1, 3x + 1)$
- Otherwise $g \equiv 0$ for all other (x, y)
- This is a toy example: we could sample points from the region directly

Example: setting up the Gibbs sampler

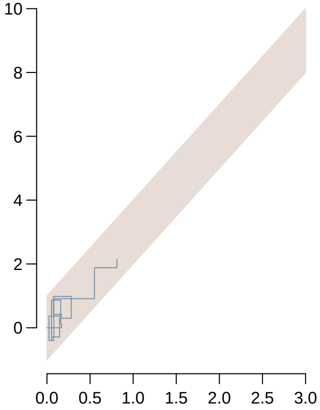
Identify the conditional distributions to sample from

- Given x, we can sample y from a uniform on $(3x-1, 3x+1)$
- Given y, sample x from a uniform on $\left(\min\left\{0, \frac{y-1}{3}\right\}, \max\left\{5, \frac{y+1}{3}\right\}\right)$

Example: First 25 iterations

Initialize and run

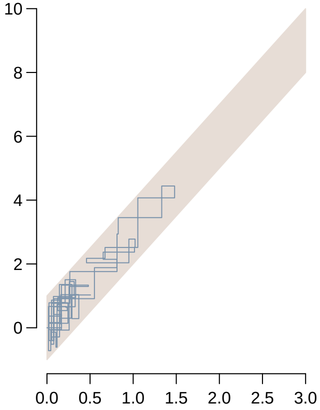
- Initialize at $(0, 0)$
- Perform 25 moves, where we alternate sampling from the conditionals



Example: First 100 iterations

Initialize and run

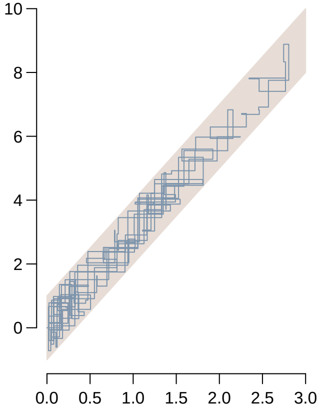
- Initialize at $(0, 0)$
- Perform 100 moves, where we alternate sampling from the conditionals



Example: First 250 iterations

Initialize and run

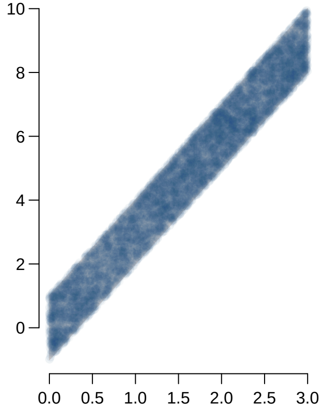
- Initialize at $(0, 0)$
- Perform 250 moves, where we alternate sampling from the conditionals



Example: After 10,000 iterations

Considerations

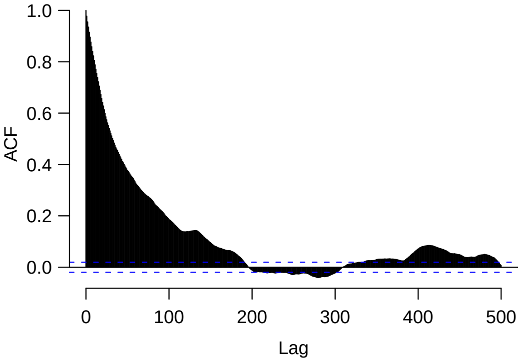
- Does it look like it has explored the entire space?
- Based on the 25, 100, and 250 first iterations, was it difficult to explore the entire space?



Example: Examining the ACF

Remarks

- The ACF of the first variable (x) is shown
- If we looked further into the ACF, we would see further evidence of oscillation (but it eventually settles)



Why do we need Gibbs?

We generally do not need Gibbs when...

- We can easily sample from the joint distribution directly (e.g. multivariate normal)

When/why we use Gibbs

- The conditionals can be built to have a parametric form that can easily be sampled
- Gibbs offers the advantage of an acceptance rate of 100%
- Sometimes multiple dimensions can be sampled simultaneously, increasing speed and how fast we move through the distribution
- Gibbs can be combined with Metropolis-Hastings, e.g. Gibbs for some variables, M-H for others

Recap on Gibbs Sampling

Acceptance rate: 100%

- Gibbs Sampling is a process of sampling from conditional distributions
  - If $\theta = (\theta_1, \theta_2, \theta_3)$, we can sample $\theta_1^{(i)}$ using

$$g_1\left(\theta_1|\theta_2^{(i-1)}, \theta_3^{(i-1)}\right)$$

  - Similarly for θ_2 and θ_3:

$$\theta_2^{(i)} \sim g_2\left(\theta_2|\theta_1^{(i)}, \theta_3^{(i-1)}\right)$$
$$\theta_3^{(i)} \sim g_3\left(\theta_3|\theta_1^{(i)}, \theta_2^{(i)}\right)$$

Normal, mean and variance unknown

Problem setup

- Data follows a normal model
- Mean and variance are unknown
- We will ultimately work with conditional posteriors
- As we will see later in the course, we can actually sample directly from the posterior (but Gibbs here works nearly as well)

Choosing a (conditionally) conjugate prior

- The priors will be constructed by examining one variable at a time
- The prior information/data about the mean and variance are independent

Normal, mean and variance unknown

The likelihood: $y_i \sim N(\mu, \sigma^2)$ for $i = 1, ..., 20$

$$
\begin{aligned}
P(y|\mu,\sigma^2) &= \prod_{i=1}^{20} (2\pi\sigma^2)^{-1/2} \exp\left\{-\frac{(y_i-\mu)^2}{2\sigma^2}\right\} \\
&\propto \left((\sigma^2)^{-1/2}\right)^{20} \exp\left\{-\sum_{i=1}^{20} \frac{(y_i-\mu)^2}{2\sigma^2}\right\} \\
&= (\sigma^2)^{-10} \exp\left\{-\frac{1}{2\sigma^2}\left[\sum_{i=1}^{n}(y_i-\bar{y})^2 + n(\bar{y}-\mu)^2\right]\right\}
\end{aligned}
$$

Choosing a (conditionally) conjugate prior

- Examine the likelihood, thinking only of μ as a variable
  What would a reasonable conjugate look like? normal
- Examine the likelihood, thinking only of σ^2 as a variable
  What is a helpful conjugate? inverse gamma

A conjugate for μ

The likelihood: $y_i \sim N(\mu, \sigma^2)$ for $i = 1, ..., 20$

$$P(y|\mu,\sigma^2) \propto (\sigma^2)^{-10} \exp\left\{-\frac{1}{2\sigma^2}\left[\sum_{i=1}^{n}(y_i-\bar{y})^2 + n(\bar{y}-\mu)^2\right]\right\}$$
$$\propto \exp\left\{-\frac{1}{2\sigma^2}\sum_{i=1}^{n}(y_i-\bar{y})^2\right\}\exp\left\{-\frac{1}{2\sigma^2}n(\bar{y}-\mu)^2\right\}$$
$$\propto \exp\left\{-\frac{1}{2\sigma^2}n(\bar{y}-\mu)^2\right\}$$

What is an appropriate conjugate when considering only μ?

A conjugate for μ

The likelihood: $y_i \sim N(\mu, \sigma^2)$ for $i = 1, ..., 20$

$$P(y|\mu,\sigma^2) \propto (\sigma^2)^{-10} \exp\left\{-\frac{1}{2\sigma^2}\left[\sum_{i=1}^{n}(y_i - \bar{y})^2 + n(\bar{y}-\mu)^2\right]\right\}$$

$$\propto \exp\left\{-\frac{1}{2\sigma^2}\sum_{i=1}^{n}(y_i-\bar{y})^2\right\} \exp\left\{-\frac{1}{2\sigma^2}n(\bar{y}-\mu)^2\right\}$$

$$\propto \exp\left\{-\frac{1}{2\sigma^2}n(\bar{y}-\mu)^2\right\}$$

P(y | miu)

What is an appropriate conjugate when considering only μ?

- Normal distribution
- Suppose prior information suggests the location parameter, μ, can be reasonably modeled via $N(40, 9)$

A conjugate for σ^2

The likelihood: $y_i \sim N(\mu, \sigma^2)$ for $i = 1, ..., 20$

$$P(y|\mu, \sigma^2) = \left(\frac{1}{\sigma^2}\right)^{10} \exp\left\{-\frac{1}{\sigma^2}\frac{1}{2}\left[\sum_{i=1}^{n}(y_i - \bar{y})^2 + n(\bar{y} - \mu)^2\right]\right\}$$

The likelihood, just thinking about σ^2 as variable

- Takes the form of an inverse-gamma distribution (look at it like $1/\sigma^2$ is the variable and it will look like a gamma distribution)
- Inverse gamma:

$$P(\sigma^2|\alpha, \beta) \propto \left(\frac{1}{\sigma^2}\right)^{\alpha+1} \exp\left\{-\frac{\beta}{\sigma^2}\right\}$$

- We will suppose some background information suggests parameters $\alpha = 16.22, \beta = 172.2$, which results in the distribution having mean 10 and variance 9.

Joint posterior

Recap

- What is important is that we are going to obtain a posterior where we can sample from the conditional distributions

Total prior

$$P(\mu, \sigma^2) = [\ \mu \sim \mathrm{N}(40,\ 9)\ ] \ * \ [\ \sigma^2 \sim \text{Inv.-gamma}(16.22,\ 172.2)\ ]$$

$$\propto \left[\exp\left\{-\frac{(\mu-40)^2}{2*9}\right\}\right] * \left[\left(\frac{1}{\sigma^2}\right)^{17.22} \exp\left\{-\frac{172.2}{\sigma^2}\right\}\right]$$

Likelihood

$$P(y|\mu, \sigma^2) = (\sigma^2)^{-10} \exp\left\{-\frac{1}{2\sigma^2}\left[\sum_{i=1}^{n}(y_i - \bar{y})^2 + n(\bar{y}-\mu)^2\right]\right\}$$

Joint posterior

$$
\begin{aligned}
&P(\mu,\sigma^2|y)\\
&\quad\propto P(y|\mu,\sigma^2) * P(\mu,\sigma^2)\\
&\quad\propto (\sigma^2)^{-10}\exp\left\{-\frac{1}{2\sigma^2}\left[\sum_{i=1}^{n}(y_i-\bar{y})^2+n(\bar{y}-\mu)^2\right]\right\}\\
&\qquad\left[\exp\left\{-\frac{(\mu-40)^2}{2*9}\right\}\right] * \left[\left(\frac{1}{\sigma^2}\right)^{17.22}\exp\left\{-\frac{172.2}{\sigma^2}\right\}\right]\\
&\quad=(\sigma^2)^{-27.22}\exp\left\{-\frac{1}{2\sigma^2}\left[\sum_{i=1}^{n}(y_i-\bar{y})^2+n(\bar{y}-\mu)^2+346.4\right]\right\}\\
&\qquad\exp\left\{-\frac{(\mu-40)^2}{2*9}\right\}
\end{aligned}
$$

Conditional posterior for σ^2

$$P(\sigma^2|y,\mu)$$

$$\propto (\sigma^2)^{-27.22}\exp\left\{-\frac{1}{2\sigma^2}\left[\sum_{i=1}^{n}(y_i-\bar{y})^2+n(\bar{y}-\mu)^2+346.4\right]\right\}$$

$$\exp\left\{-\frac{(\mu-40)^2}{2*9}\right\}$$

$$\propto \left(\frac{1}{\sigma^2}\right)^{26.22+1}\exp\left\{-\frac{1}{2\sigma^2}\left[\sum_{i=1}^{n}(y_i-\bar{y})^2+n(\bar{y}-\mu)^2+346.4\right]\right\}$$

If σ^2 represents the random variable, then this is proportional to inverse-gamma distribution with parameters

$$\alpha_{post}=\alpha_{prior}+n/2=26.22$$

$$\beta_{post}=\beta_{prior}+\frac{1}{2}\sum_{i=1}^{n}(y_i-\bar{y})^2+\frac{n(\bar{y}-\mu)^2}{2}$$

$$=172.2+\frac{(n-1)s^2+n(\bar{y}-\mu)^2}{2}$$

Conditional posterior for μ

$$
\begin{aligned}
&P(\mu|y,\sigma^2) \\
&\propto (\sigma^2)^{-27.22} \exp\left\{-\frac{1}{2\sigma^2}\left[\sum_{i=1}^{n}(y_i-\bar{y})^2 + n(\bar{y}-\mu)^2 + 346.4\right]\right\} \\
&\quad \exp\left\{-\frac{(\mu-40)^2}{2*9}\right\} \\
&\propto \exp\left\{-\frac{1}{2}\frac{n}{\sigma^2}(\bar{y}-\mu)^2\right\} \exp\left\{-\frac{1}{2}\frac{1}{9}(\mu-40)^2\right\}
\end{aligned}
$$

This is a normal-normal conjugate case, i.e. posterior $\sim N(\mu_1, \sigma_1^2)$:

$$
\begin{aligned}
\mu_1 &= \frac{w_1\ \bar{y} + w_2\ 40}{w_1 + w_2} \\
\sigma_1^2 &= (w_1 + w_2)^{-1} = \left(\frac{n}{\sigma^2} + \frac{1}{9}\right)^{-1}
\end{aligned}
$$

The Gibbs Sampler for the conditional posteriors

Conditional sampling distributions:

- $g_\mu(\mu|y,\sigma^2)$ – Normal distribution, $N(\mu_1,\sigma_1^2)$

  $\mu_1 = \frac{w_1\bar{y}+w_2 40}{w_1+w_2}$

  $\sigma_1^2 = (w_1+w_2)^{-1} = \left(\frac{n}{\sigma^2}+\frac{1}{9}\right)^{-1}$

- $g_{\sigma^2}(\sigma^2|y,\mu)$ – Inverse-gamma

  $\alpha = \alpha_{prior} + n/2$

  $\beta = \beta_{prior} + \frac{(n-1)s^2+n(\bar{y}-\mu)^2}{2}$

Initialize

- Set $\mu^{(0)}$ and $(\sigma^2)^{(0)}$ to reasonable starting values
- May choose the estimates from prior: $\mu^{(0)} = 40$ and $(\sigma^2)^{(0)} = 10$

Recursion: for $i = 1, 2, ..., N$

- Sample $\mu^{(i)}$ from $g_\mu(\mu|y,(\sigma^2)^{(i-1)})$
- Sample $(\sigma^2)^{(i)}$ from $g_{\sigma^2}(\sigma^2|y,\mu^{(i)})$

Gibbs example – initialize

```
#===> Data <===#
y <- 34 + 25*rt(20, 25)

#===> Summaries <===#
yn <- length(y)
yM <- mean(y)
yV <- var(y)

#===> Algorithm <===#
N  <- 1e4
mu <- rep(40, N)
s2 <- rep(9, N)

#===> Prior Info <===#
pM <- 40
w2 <- 1/9
pA <- 16.22
pB <- 172.2
```

Gibbs example – iterate

```
#===> Iterate <===#
for(i in 2:N){
  w1     <- yn / s2[i-1]
  normP1 <- (w1*yM + w2*pM)/(w1+w2)
  normP2 <- 1/(w1+w2)
  mu[i]  <- rnorm(1, normP1, sqrt(normP2))

  gamP1 <- pA + yn/2
  temp  <- yV*(yn-1) + yn*(yM - mu[i])^2
  gamP2 <- pB + temp/2
  s2[i] <- 1/rgamma(1, gamP1, gamP2)
}
mu <- mu[-(1:100)]
s2 <- s2[-(1:100)]
```

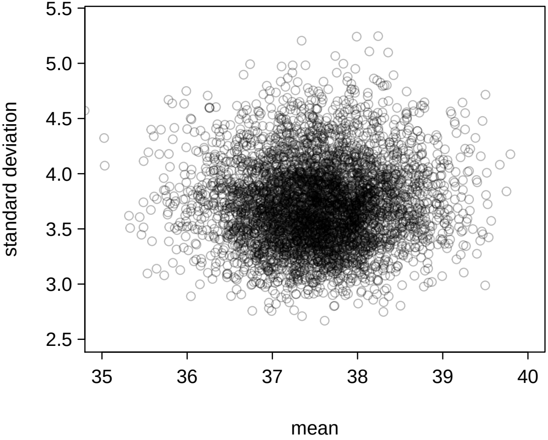
conditional posterior

if not use mu[i] but use mu[i-1], wrong result

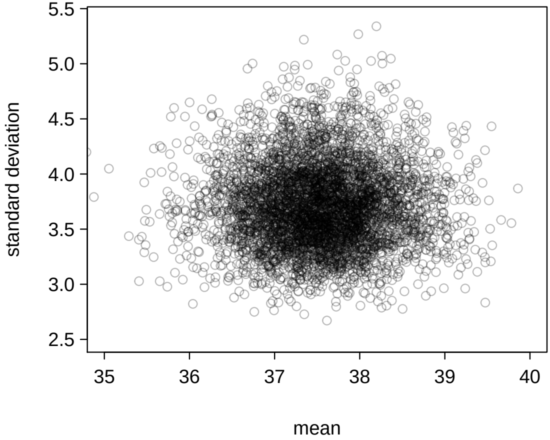
as inverse gamma

Specially note that `s2[i-1]` is used to obtain `mu[i]`,
but `mu[i]` is used to obtain `s2[i]`

Gibbs example – correct results



Gibbs example – wrong results



If `mu[i-1]` was used to obtain `s2[i]`

Convergence considerations

Proposal function

- Vary the proposal function, and check if the chain converges faster or propagates better

Starting values

- Try several
- If some converge to a different mode, the proposal function must be chosen so a single Markov chain explores both modes

Tools at your disposal

- Autocorrelation function (ACF)
- Trace plots
- Multiple chains (coming later)

Function of the Day – stop, warning

Stop a function or output a warning

```
> mySeq <- function(x, y, n=10){
+   if(x == y){
+     warning("x==y, consider using 'rep' function")
+   }
+   seq(x, y, length.out=n)
+ }
> mySeq(5,5)
 [1] 5 5 5 5 5 5 5 5 5 5
Warning message:
In mySeq(5, 5) :  x==y, consider using 'rep' function
```

Coding Tip of the Day – White Space

Use white space wisely

- Balance concerns about code density
  - Very disperse can be difficult to browse
  - Overly dense code might force a reader to review the code like a book fashion
- Keep each layer in code (e.g. commands in a `for`-loop) in proper alignment
- In general, do not automatically indent code without reason

Careful use of white space is related to alignment of assignment characters