

Lecture 4: an introduction to MCMC

Ben Lambert¹

`ben.lambert@some.ox.ac.uk`

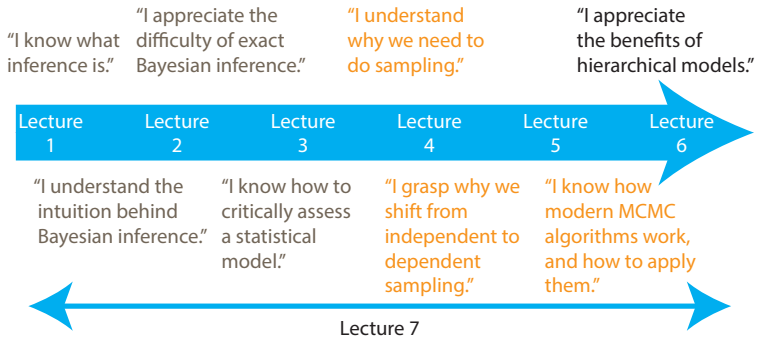
¹Somerville College
University of Oxford

Monday 14th November, 2016

Lecture outcomes

- 1 Appreciate how sampling can be used to gain insight into a distribution.
- 2 Grasp how **dependent sampling** via **MCMC** allows sampling from the posterior, even though it is un-normalised.
- 3 Understand the mechanics of Random Walk Metropolis, and how it works intuitively.
- 4 Know that judging convergence of chains to the posterior is *hard*.
- 5 However, recognise that using multiple chains can give us more confidence in convergence.
- 6 Understand how the worth of a sampler is $\#$ effective samples per second.

Our progress in the overall course



- 1 Recap from last lecture
- 2 Understanding a distribution by sampling from it
- 3 Introducing dependent sampling
- 4 Random Walk Metropolis
- 5 Judging convergence of chains to posterior
- 6 Effective sample size

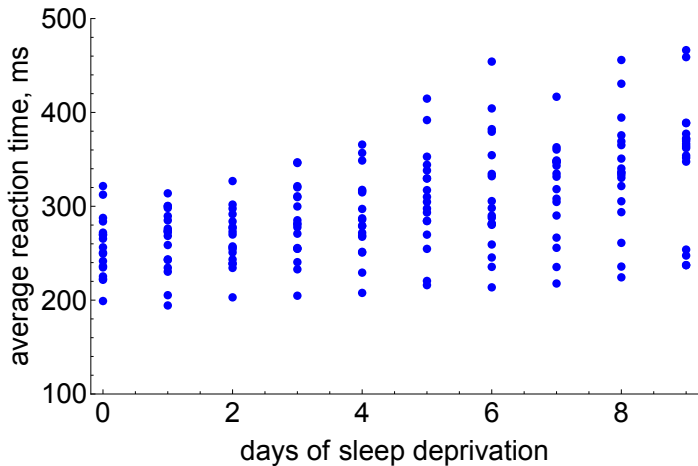
Sleep deprivation study: posterior predictive checks

- Data from a laboratory experiment that measured the effect of sleep deprivation on cognitive performance¹.
- 18 subjects within a population of interest - long-distance lorry drivers - volunteered to participate in the 10 day experiment.
- Subjects were restricted to 3 hours of sleep per night.
- On each day the subjects' reaction time across a range of cognitive tasks were measured.



Sleep deprivation study: model aims

- Build a model to explain the effect of sleep deprivation on reaction times.



Sleep deprivation study: model

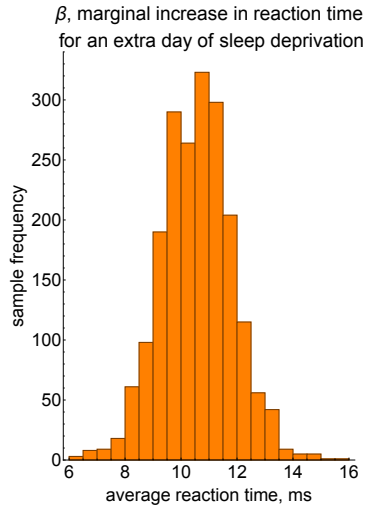
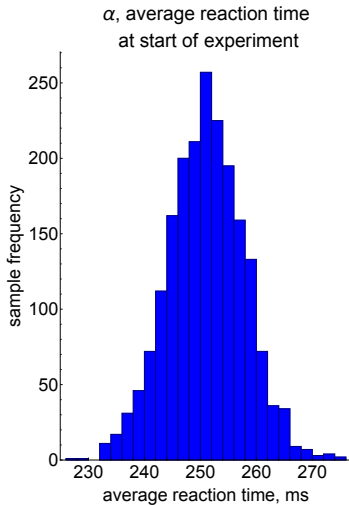
On the basis of the previous graph, assume:

$$R(t) \sim N(\alpha + \beta t, \sigma) \quad (1)$$

where $R(t)$ is the reaction time for a sleep deprivation of t days.

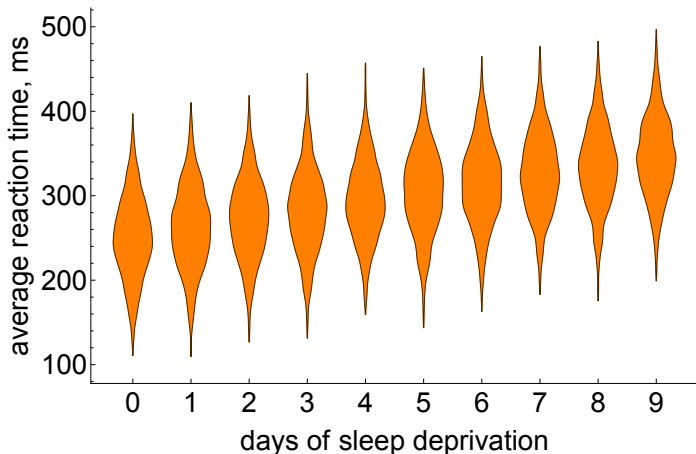


Sleep deprivation model: results



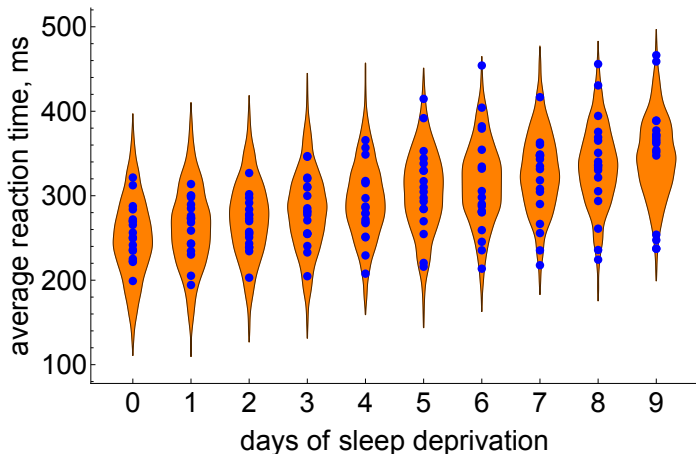
Sleep deprivation: posterior predictive checks

- **Question:** how well does model fit data?
- **Answer:** use posterior predictive distribution to check fit.



Sleep deprivation: posterior predictive checks

- **Question:** how well does model fit data?
- **Answer:** use posterior predictive distribution to check fit.

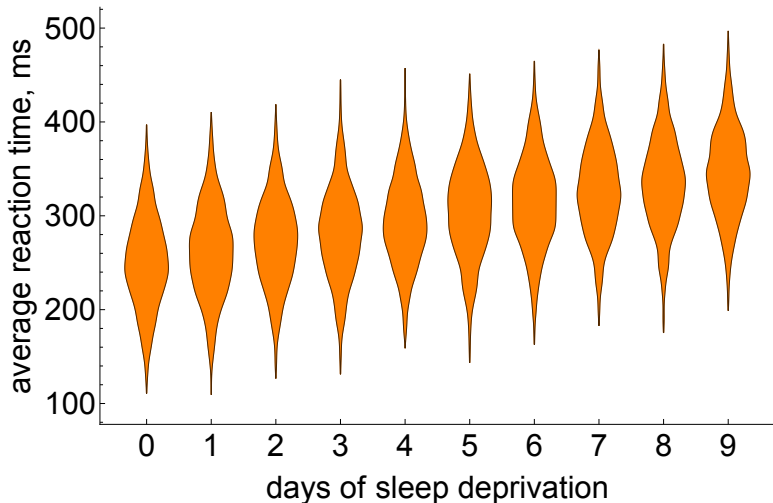


Sleep deprivation: posterior predictive checks

- Model seems to fit data well...but is this false hope?
- **Question:** how does it fare at the **individual** subject level?

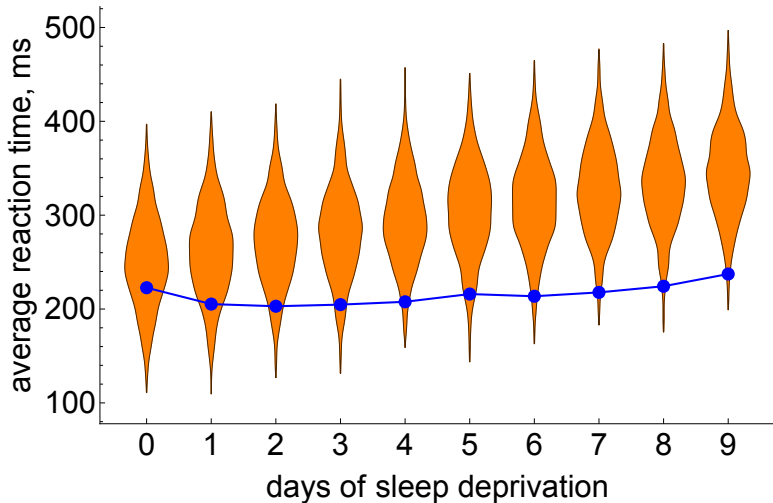
Sleep deprivation: posterior predictive checks

Posterior predictive distribution (again).



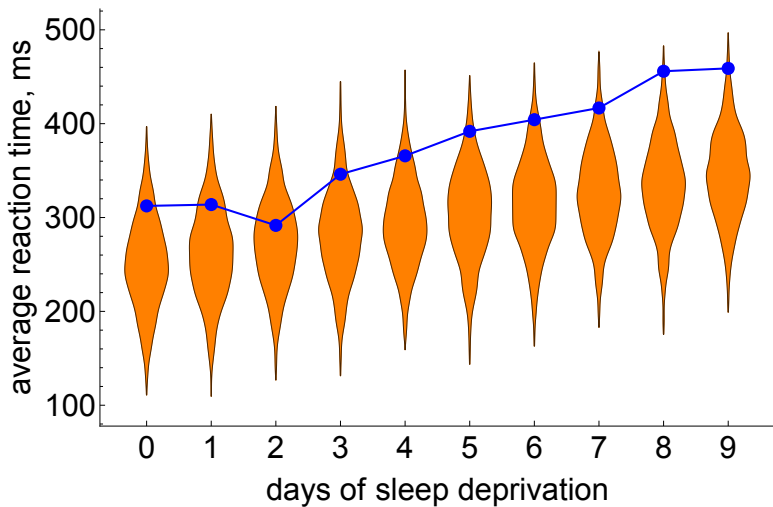
Sleep deprivation: posterior predictive checks

One participant.



Sleep deprivation: posterior predictive checks

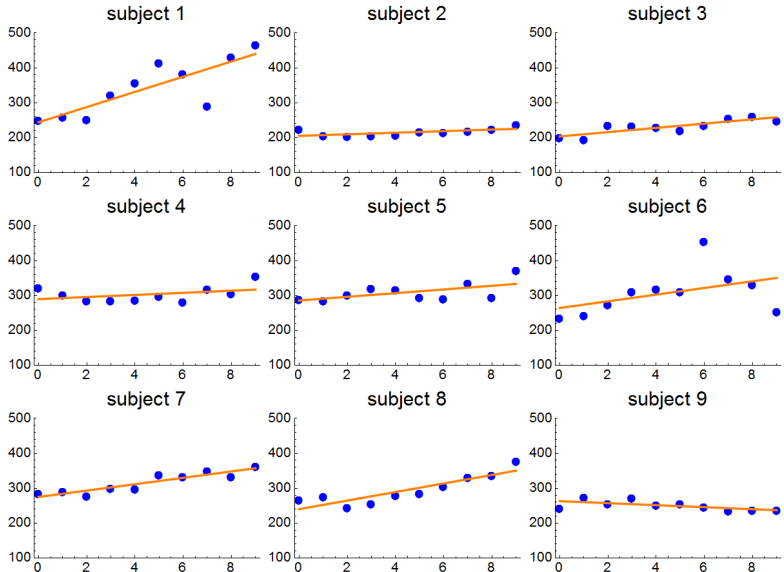
And another.



Sleep deprivation: data visualisation

- Whilst “pooled” model fits data ok on first glances it fails to explain variation at the individual level.
- Could have avoided this issue if we chose a more appropriate data visualisation in the first place!

Sleep deprivation: data visualisation



Sleep deprivation: problem and solution

Problems:

- $\Rightarrow$ considerable heterogeneity in response of individual participants to sleep deprivation:
 - In about 60% of cases there was a large increase in reaction times over the 10 days of the experiment.
 - For a significant proportion there wasn't much of a change.
 - For one case (subject 9) there was actually a decline!
- There was also heterogeneity in "level" of reaction times at start of experiment.

Solution:

- Allow for differences in response and level between subjects.
- Best way to do this is using a hierarchical model (lectures 6 and 7).

Posterior predictive checks: summary

- **Premise:** if model fits then data simulated from the posterior predictive distribution should look similar to the real data.
- There is considerable choice over which aspect of the simulated data to compare with real.
- No hard rules for this choice but should be guided by a model's eventual use.
- Essential to choose right visualisation **before** you start modelling!

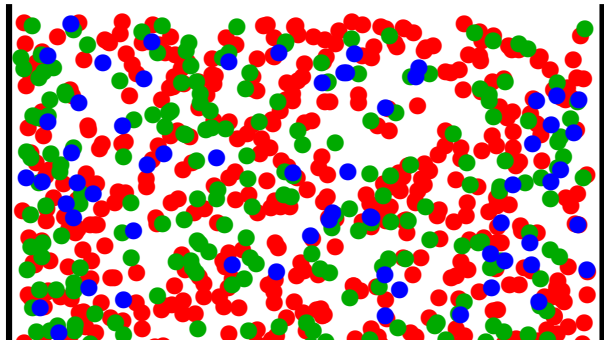
"All posterior predictive checks are equal but some posterior predictive checks are more equal than others."



- 1 Recap from last lecture
- 2 Understanding a distribution by sampling from it**
- 3 Introducing dependent sampling
- 4 Random Walk Metropolis
- 5 Judging convergence of chains to posterior
- 6 Effective sample size

What is (independent) sampling and how can it give insight to distributions?

- Suppose we have a large (infinite) urn filled with coloured balls.
- The number of colours and the frequencies of each are **unknown**.
- **Question:** how can we determine the underlying probability distribution of ball colour?

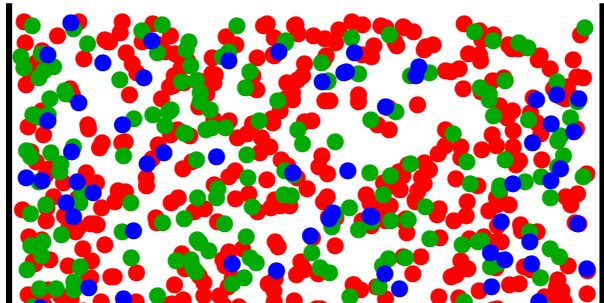


What is (independent) sampling and how can it give insight to distributions?

Answer: we draw lots of balls from the urn and count the **sampled** frequencies!

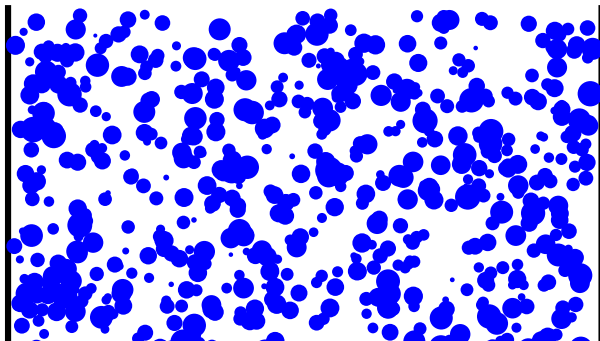
What is (independent) sampling and how can it give insight to distributions?

- Drawing one ball from the urn is the act of taking a single **sample**.
- If the balls in the urn are swishing about then the colour of the next ball does not depend on the current ball's colour.
- Here the samples are (conditionally-) **independent**.
- Independent sampling gives us a very **efficient** way of gaining insight into a distribution.



Sampling from a continuous distribution

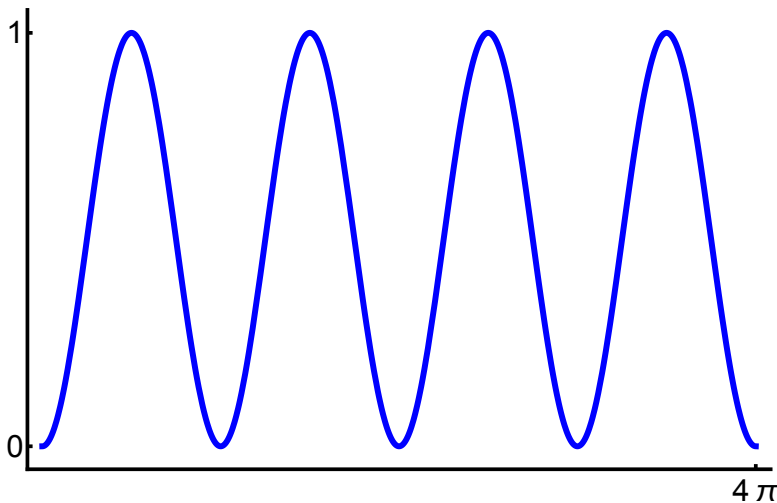
- Suppose we have a large (infinite) urn filled with balls of differing sizes.
- The distribution of sizes is **unknown**.
- **Question:** can we use same method to determine the underlying probability distribution of ball size? **Answer:** yes!



Sampling from a continuous distribution

Generating independent samples: sine curve

Question: how can we generate independent samples from the following (un-normalised) PDF?



Generating independent samples: sine curve

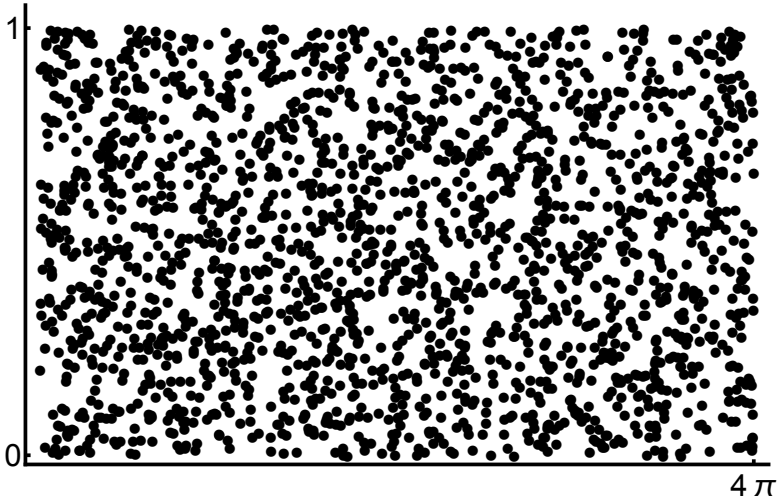
Answer: do the following a large number of times:

- 1 Generate **x** coordinates: uniformly-distributed points from $(0, 4\pi)$; where 4π is the domain of the function.
- 2 Generate **y** coordinates: uniformly-distributed points from $(0, 1)$; where 1 is the maximum value of the function.
- 3 If $y < p(x)$, then **accept** **x** coordinate as a sample.
- 4 If $y \geq p(x)$, then **reject** **x** coordinate as a sample.

Known as **Rejection** sampling.

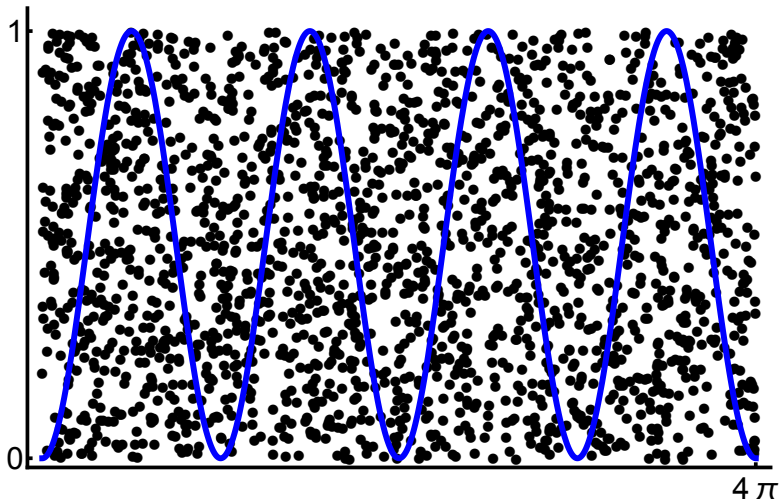
Generating independent samples: sine curve

Generate x and y coordinates.



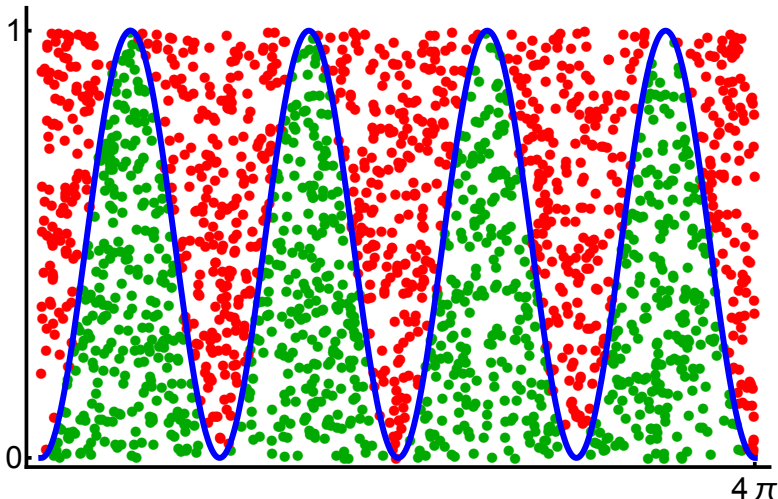
Generating independent samples: sine curve

Overlay pdf.



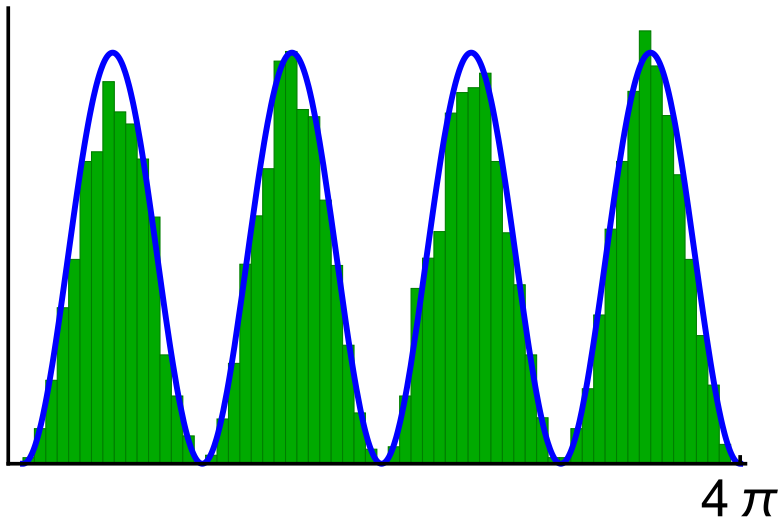
Generating independent samples: sine curve

Accept x coordinates as samples if $y < p(x)$.



Generating independent samples: sine curve

The resultant samples.



Generating independent samples: cow distribution

- Rejection sampling works well for this (fairly) simple problem.
- **Question:** how well does it generalise to more complex distributions?

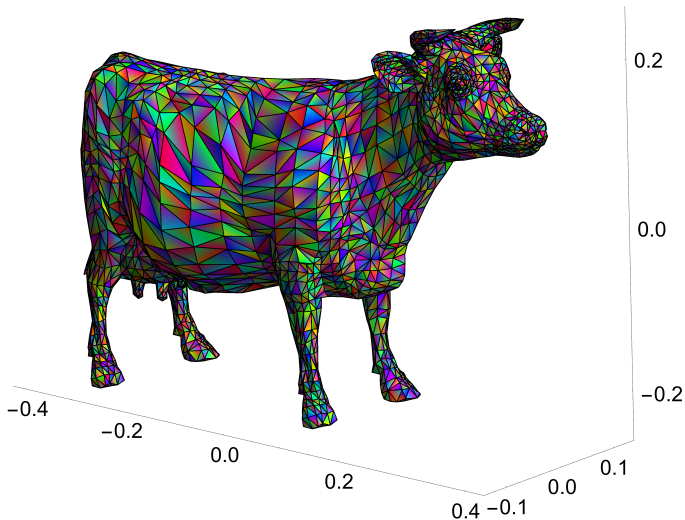
Suppose we have the following pdf from which we want to generate independent samples:

$$f(x, y, z) = \begin{cases} \frac{1}{V_{\text{cow}}}, & \text{if } (x, y, z) \in \text{cow} \\ 0, & \text{otherwise} \end{cases}$$

$\implies$ uniform within cow boundaries.

Generating independent samples: cow distribution

Cow boundaries.



Generating independent samples: cow distribution

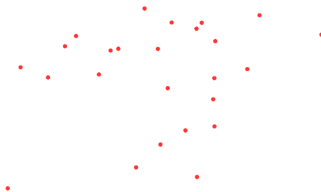
$$f(x, y, z) = \begin{cases} \frac{1}{V_{\text{cow}}}, & \text{if } (x, y, z) \in \text{cow} \\ 0, & \text{otherwise} \end{cases}$$

- **Question:** how can we generate uniformly distributed samples within the cow?
- **Answer:** use rejection sampling! So do the following:
 - 1 Generate uniformly-distributed $(\mathbf{x}, \mathbf{y}, \mathbf{z})$ within bounding box containing cow.
 - 2 Generate uniformly-distributed w on $(0, \frac{1}{V_{\text{cow}}})$.
 - 3 If $w < f(x, y, z) \implies$ accept; otherwise reject.

Note that since all points in cow will have $w < f(x, y, z)$, and all points outside $w > f(x, y, z)$ rejection sampling is simple; we reject all points outside cow, and accept all in it!

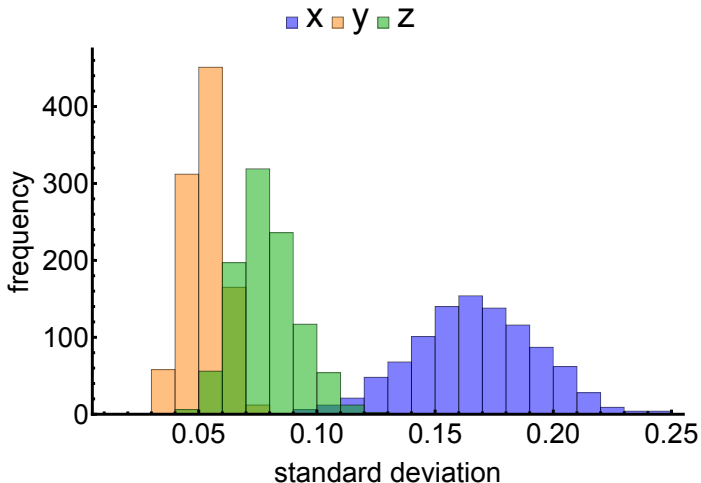
Generating independent samples: cow distribution

$n = 100$



Generating independent samples: cow distribution

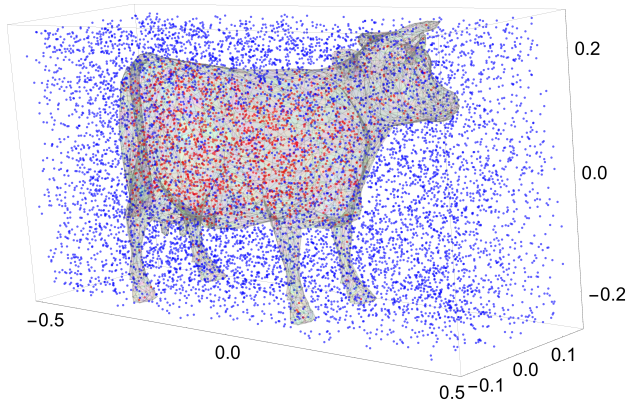
Use the samples to estimate the standard deviation of the distribution in all three dimensions.



Generating independent samples: cow distribution

Question: what's the problem with rejection sampling?

Answer: inefficiency that increases (exponentially) with number of dimensions.



Why do sampling in the first place?

Typically we want to calculate the posterior mean of some parameter, θ_1 :

$$\begin{aligned} E(\theta_1|X) &= \int_{\Theta_1} \int_{\Theta_{-1}} \theta_1 \times p(\theta_1, \theta_{-1}|X) d\theta_{-1} d\theta_1 \\ &= \int_{\Theta_1} \theta_1 \times p(\theta_1|X) d\theta_1 \end{aligned}$$

where θ_{-1} corresponds to the $d - 1$ other parameters of the model.

This integral (the top line) is just too difficult to calculate exactly for all but the simplest models $\implies$ we instead use sampling to approximate it!

Why is generating independent samples difficult?

- **Rejection sampling** requires generation of a large number of random points to produce relatively few samples.
- This inefficiency increases (exponentially) with the dimensionality of the distribution; i.e. for posteriors with more parameters.
- Other methods exist (inverse-transform sampling and importance sampling, for example) but they suffer from complexity and/or inefficiency issues.
- We cannot calculate the denominator so are unable to use some of these methods.
- Even if we had the denominator the complexity of most models means that independent sampling isn't possible.

Is sampling finished?



- 1 Recap from last lecture
- 2 Understanding a distribution by sampling from it
- 3 Introducing dependent sampling**
- 4 Random Walk Metropolis
- 5 Judging convergence of chains to posterior
- 6 Effective sample size

What is dependent sampling?

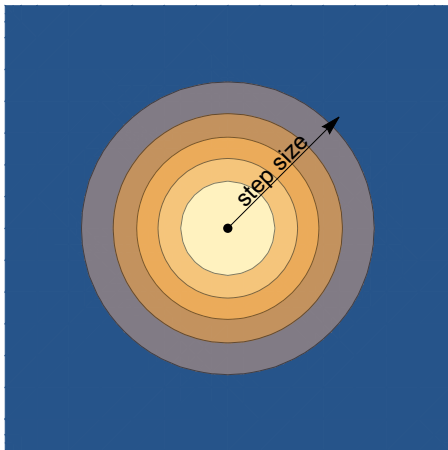
Definition:

“A sampling algorithm where the next sample **depends** on the current value.”

And the list of all (accepted) positions of the sampler form the sample.

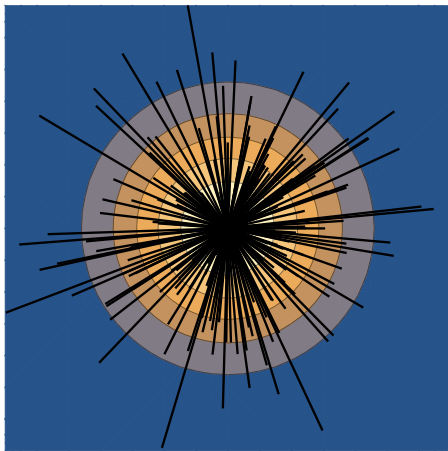
Example dependent sampler: choose a new position based on a local “jumping” distribution

Suppose the next value of the sampler is drawn from a 2d normal distribution centred on our current position.

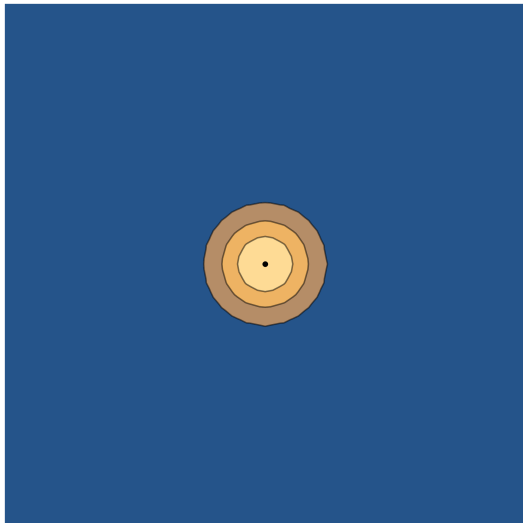


Example dependent sampler: next steps

Showing 200 example steps.



Visualising the path of a dependent sampling



Dependent samplers as Markov Chains (Monte Carlo)

- Where to step next is determined via a distribution *conditional* on the current parameter value.
- This stepping is probabilistic $\implies$ *Monte Carlo*.
- The conditional distribution only depends on the current value of the sampler meaning it is memoryless about past path.
- This memoryless means that the path of the sampler is a *1st order Markov Chain*.



Open questions

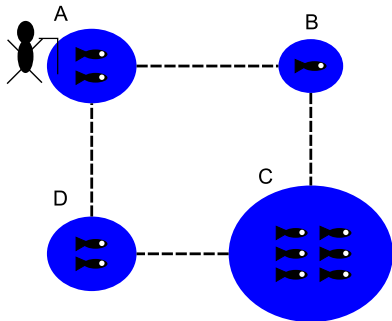
How can we decide on the:

- ① Starting position.
- ② Jumping distribution's shape.

To ensure convergence to the posterior distribution? Especially because we cannot compute the posterior itself!

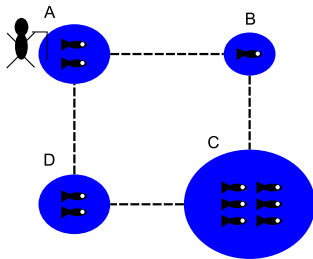
- 1 Recap from last lecture
- 2 Understanding a distribution by sampling from it
- 3 Introducing dependent sampling
- 4 Random Walk Metropolis**
- 5 Judging convergence of chains to posterior
- 6 Effective sample size

David Robinson's fishing



- David Robinson (a more fortunate cousin of Robinson Crusoe) is marooned on an island.
- Access to four freshwater lakes of different sizes; each with a supply of fish.

David Robinson's fishing



- Robinson does not know the amount of fish in each lake.
- He also does not know the number of lakes!
- However, the amount of fish in each lake is proportionate to its area.
- From a particular lake he can see the two adjoining lakes, and can estimate their area.

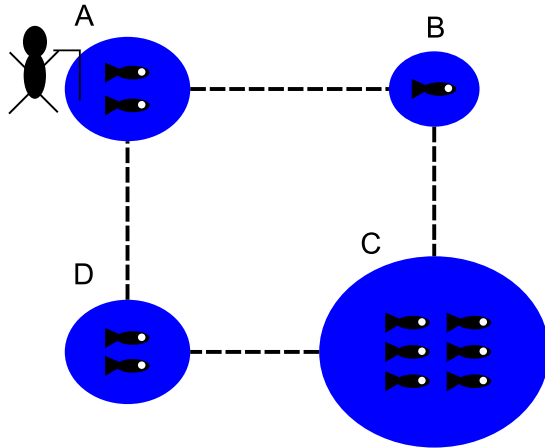
David Robinson's fishing

- He has a terrible memory (too much coconut toddy), and each day forgets any estimates of lake size he made previously.
- He wants to fish (at maximum) one new lake per day.
- He possesses a coin and a solar-powered calculator that can generate (pseudo-)random numbers uniformly distributed between 0 and 1.
- He is initially “washed up” next to lake A.



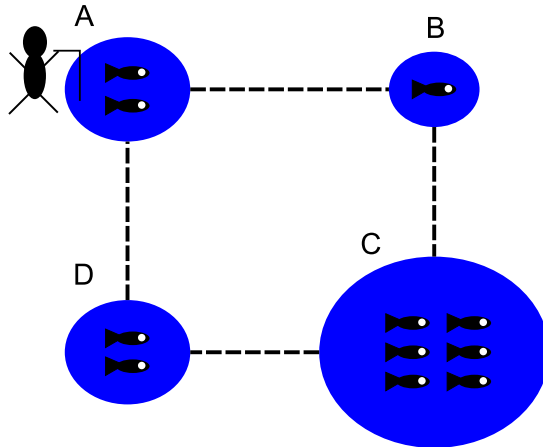
David Robinson's fishing

- **Question:** What strategy should he use to fish as sustainably as possible?



David Robinson's fishing

- **Remember:** Robinson doesn't know the # of lakes, nor the amount of fish in each!

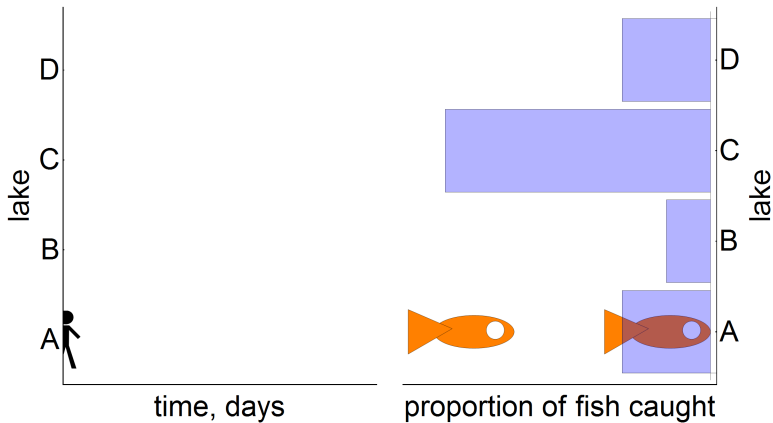


David Robinson's fishing

Answer: visit each lake in proportion to the fish it contains, by doing the following:

- 1 Each night he flips the coin.
- 2 If it's heads (tails) he proposes a move to the neighbouring lake in the clockwise (anticlockwise) direction.
- 3 Calculates the ratio of the size of the proposed lake to the current one.
- 4 Compares the ratio with a (pseudo-)random number from the calculator.
- 5 If the ratio exceeds the generated number, he moves. If not, he stays put and fishes the same lake tomorrow.

David Robinson's fishing: does it work?



David Robinson's fishing: summary

- Robinson lacked knowledge of *numbers* of fish in each lake and the number of lakes.
- Knows that the number of fish in each lake is proportionate to its size.
- His memory stops him remembering the exact sizes.
- Each night he flips a coin; heads (tails) $\implies$ consider clockwise (anticlockwise) neighbouring lake.
- Estimates ratio of size of selected lake to current one.
- If ratio exceeds a uniform random number he moves. If not he stays where he is.
- After about 100 days his “random” strategy is quite similar from an “omniscient” one.

Defining Random Walk Metropolis

Robinson's strategy is an example of the "Random Walk Metropolis" algorithm. This has the following form:

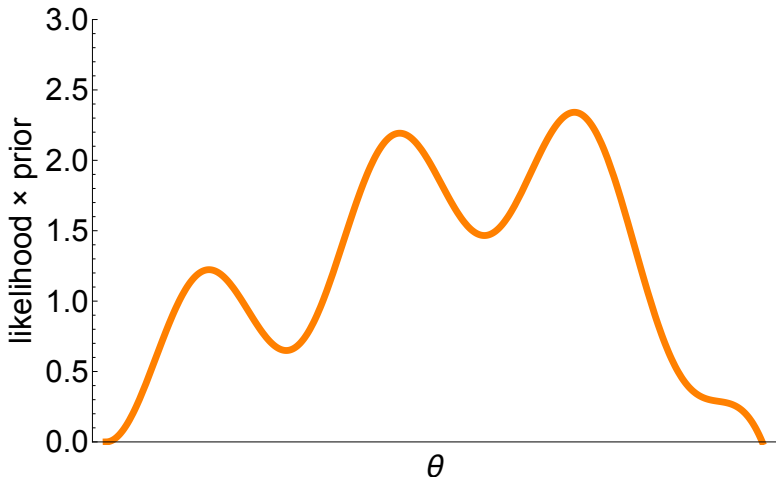
- Generate a random starting location θ_0 .
- Iterate the following for $t = 1, \dots, T$:
 - Propose a new location from a jumping distribution:
 $\theta_{t+1} \sim J(\theta_{t+1}|\theta_t)$.
 - Calculate the ratio:

$$r = \frac{\text{likelihood}(\theta_{t+1}) \times \text{prior}(\theta_{t+1})}{\text{likelihood}(\theta_t) \times \text{prior}(\theta_t)} \quad (2)$$

- Compare r with a uniformly-distributed number u between 0 and 1.
- If $r \geq u \implies$ we move.
- Otherwise, we remain at our current position.

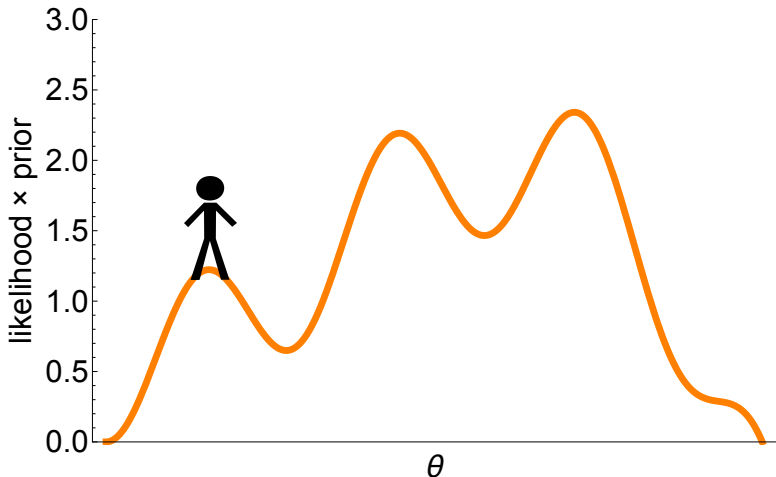
Defining Random Walk Metropolis

Start with the un-normalised density.



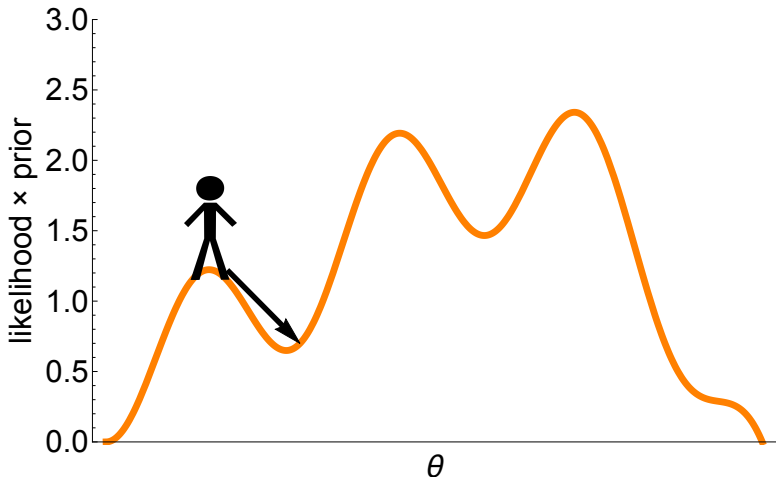
Defining Random Walk Metropolis

Select a random starting location.



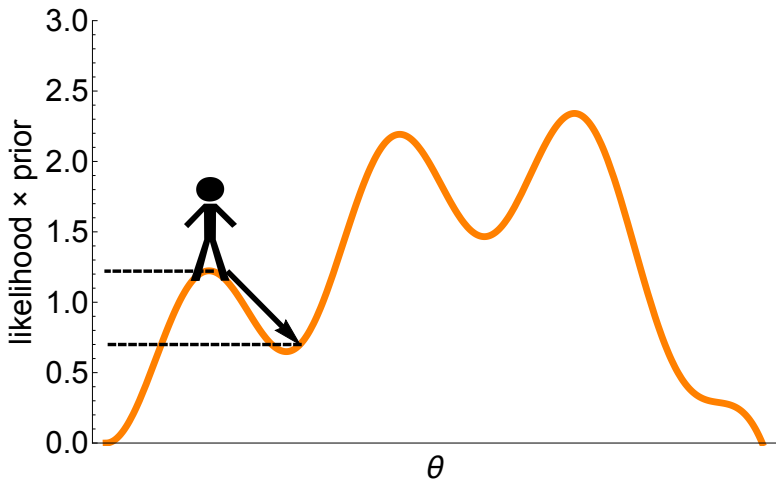
Defining Random Walk Metropolis

Propose a new location using jumping distribution.



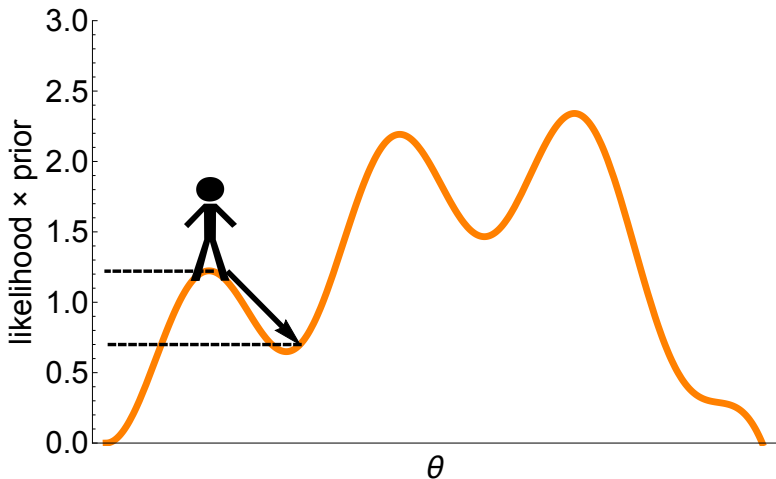
Defining Random Walk Metropolis

Calculate ratio of likelihood $\times$ prior at proposed to current location, and find $r \approx 0.58$.



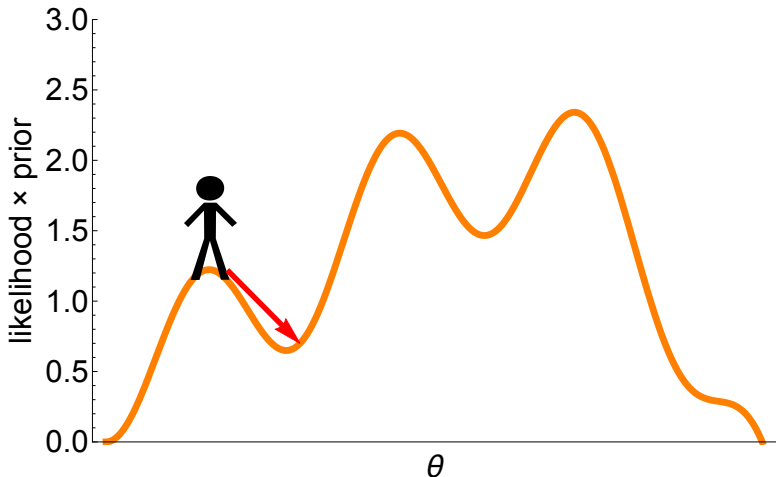
Defining Random Walk Metropolis

Compare $r \approx 0.58$ with random real between 0 and 1. For example suppose we obtain $u = 0.823$.



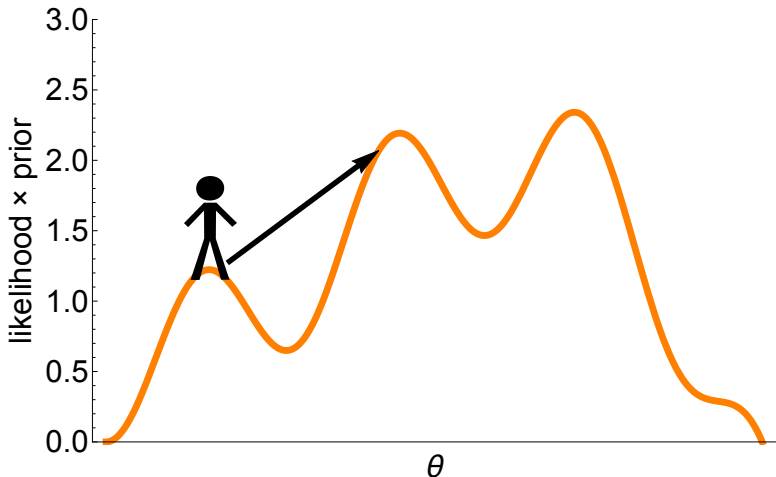
Defining Random Walk Metropolis

Since $r < u$ we remain at our original location.



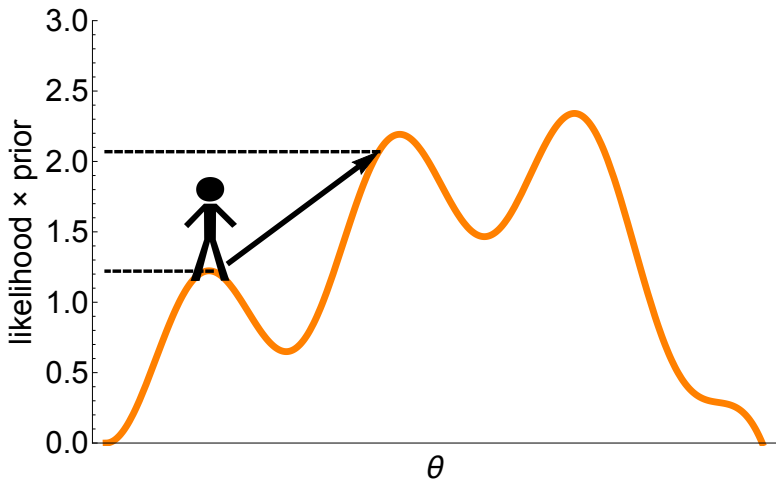
Defining Random Walk Metropolis

Generate a new proposed step using jumping distribution.



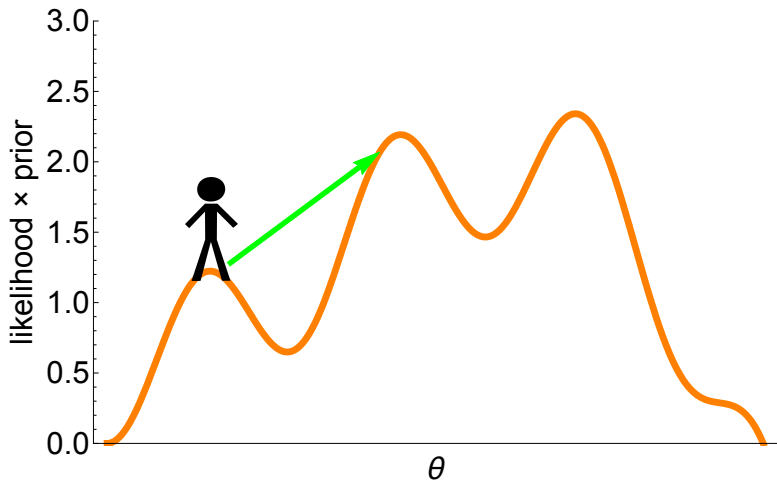
Defining Random Walk Metropolis

Calculate ratio of likelihood $\times$ prior at proposed to current location, and find $r \approx 1.75$.



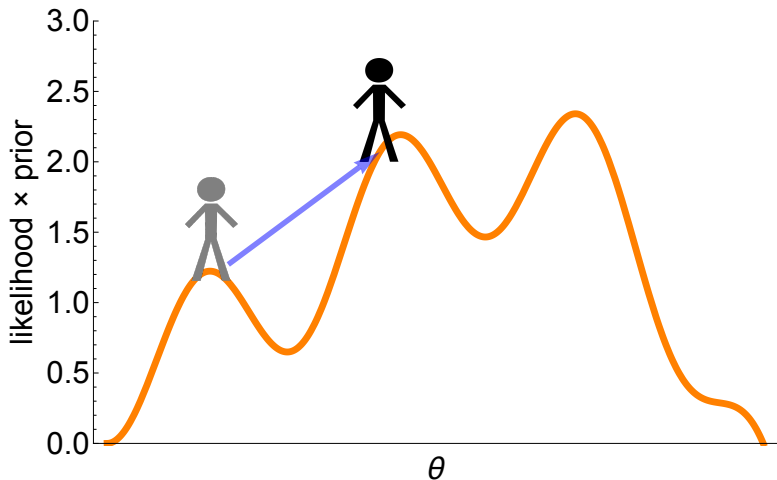
Defining Random Walk Metropolis

Since $r > 1$ (maximum possible u) $\implies$ we move to new location.



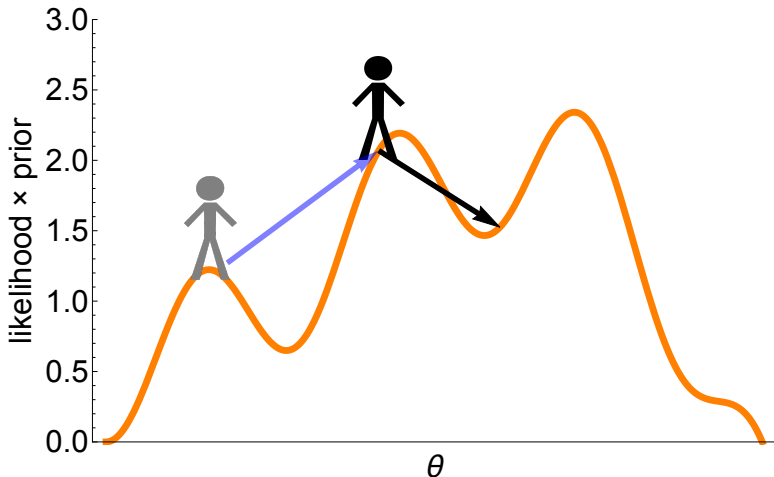
Defining Random Walk Metropolis

Since $r > 1$ (maximum possible u) $\implies$ we move to new location.



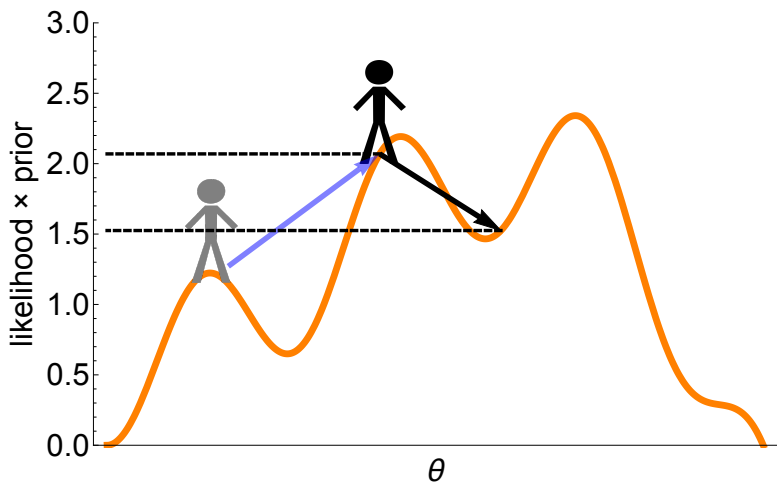
Defining Random Walk Metropolis

Propose a new step using jumping distribution.



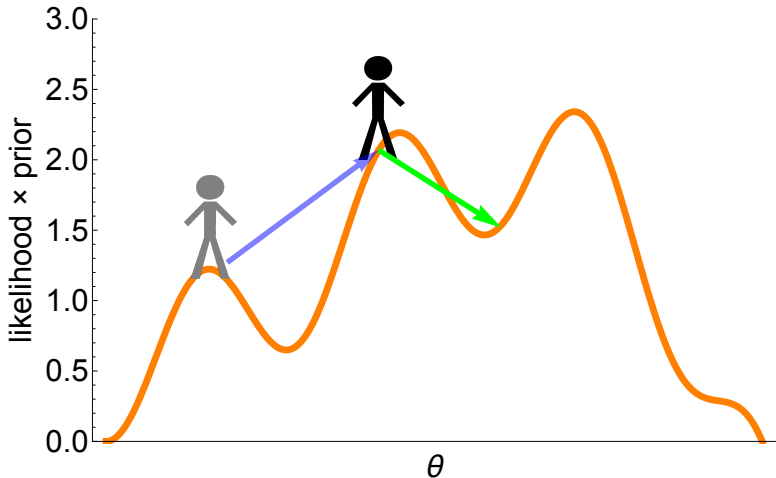
Defining Random Walk Metropolis

Calculate $r \approx 0.75$.



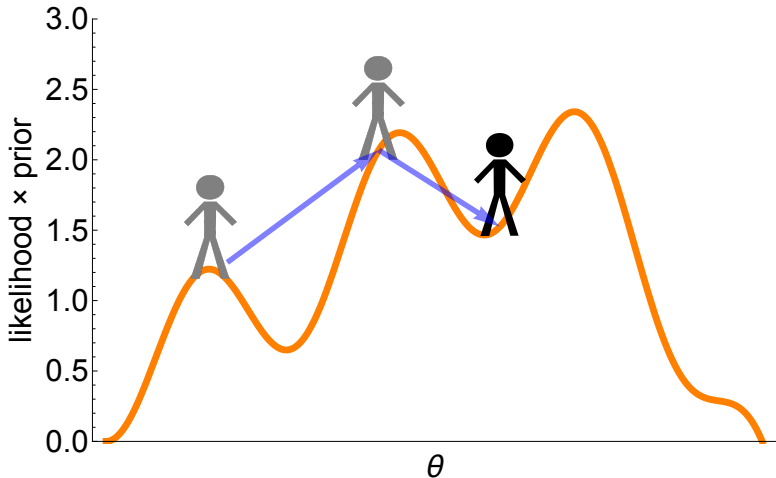
Defining Random Walk Metropolis

Generate $u = 0.278 < r \implies$ we move!



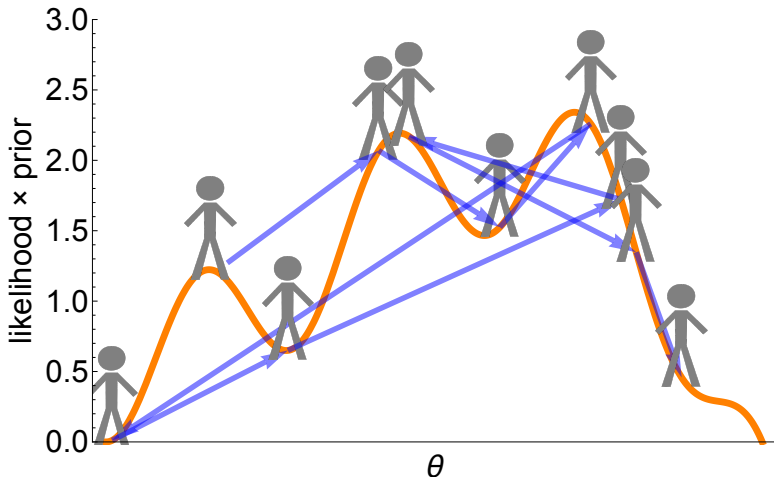
Defining Random Walk Metropolis

Generate $u = 0.278 < r \implies$ we move!



Defining Random Walk Metropolis

Repeat a large number of times.



Random Walk Metropolis: benefits

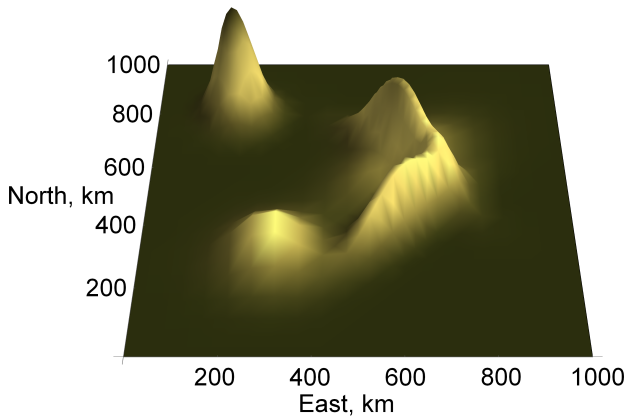
- Under quite general conditions the Random Walk Metropolis sampler converges **asymptotically** to the posterior.
- However for a sufficiently large sample size the sampling distribution may be practically indistinguishable from the true posterior.
- The algorithm requires us to be able to calculate the ratio:

$$r = \frac{\text{likelihood}(\theta_{t+1}) \times \text{prior}(\theta_{t+1})}{\text{likelihood}(\theta_t) \times \text{prior}(\theta_t)} \quad (3)$$

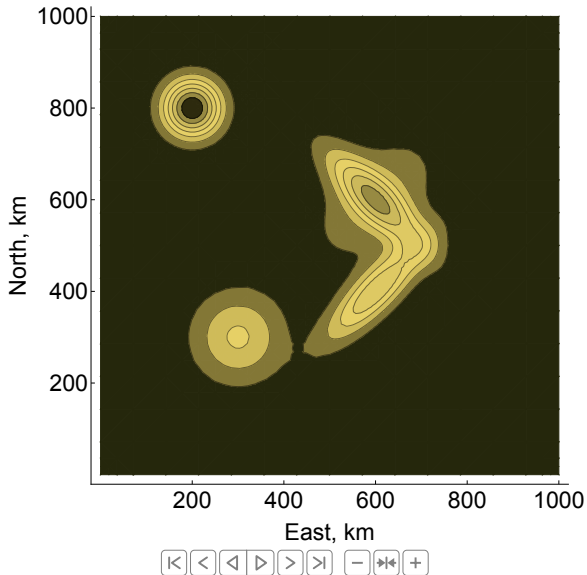
- The ratio uses **only** the numerator of Bayes' rule $\implies$ we side-step calculating the denominator!

Random Walk Metropolis in action

Can we use Random Walk Metropolis to sample from the continuous distribution below?

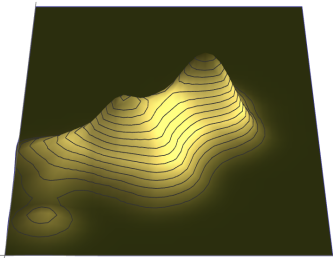


Random Walk Metropolis in action

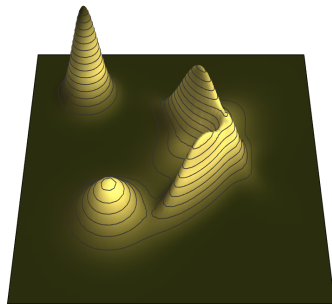


Random Walk Metropolis in action

Metropolis, $n = 100$



actual density



Random Walk Metropolis: short summary

- Algorithm works by starting in a randomly-determined position in parameter space.
- In each iteration we generate a proposed (local) step from our current position.
- We then move based the ratio of the proposed **un-normalised** posterior to our current location $\implies$ no need to calculate troublesome denominator.
- The path of our positions over time forms our **sample**.
- If we repeat the above for a (large) number of steps $\implies$ sampling distribution $\approx$ posterior.
- **Question:** what's the function of the accept/reject rule we use in the Metropolis algorithm?

The importance of the accept/reject rule

Let's try out three different accept/reject rules to see how they fare.

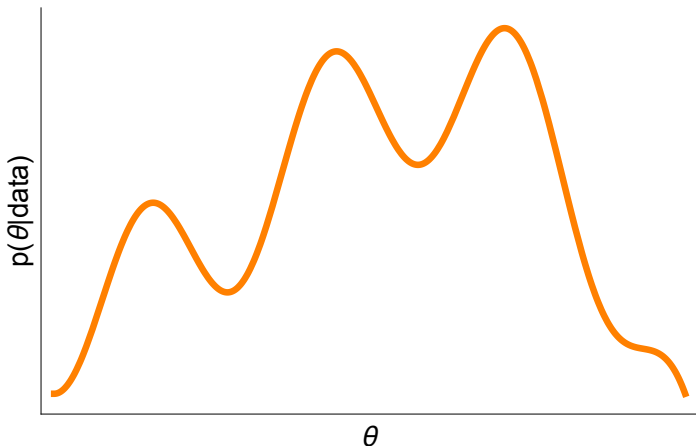
- **Drunkard's rule:** we always move irrespective of value of un-normalised density at new location versus current position.
- **Edmund Hilary's rule:** calculate

$$r = \frac{p(X|\theta_{t+1})p(\theta_{t+1})}{p(X|\theta_t)p(\theta_t)} \quad (4)$$

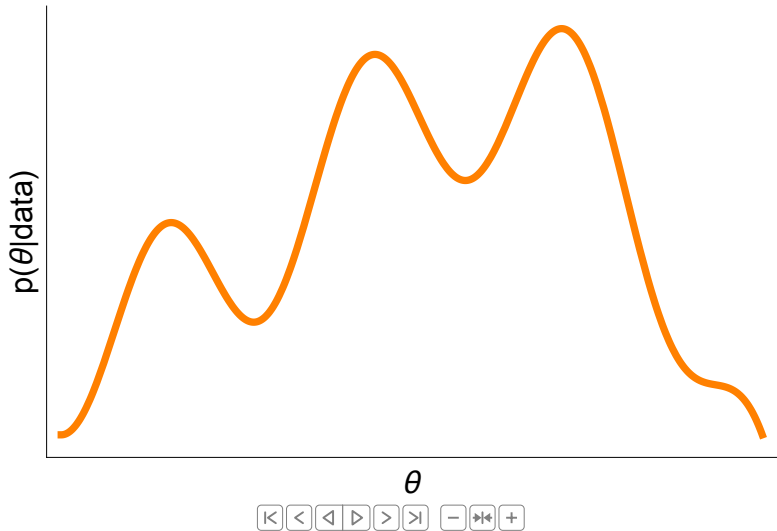
- If $r > 1$ we move; otherwise don't.
- **Metropolis rule:**
 - If $r > u \sim \text{Unif}(0, 1)$, then move to new location.
 - Otherwise stay in current position.

An example un-normalised posterior

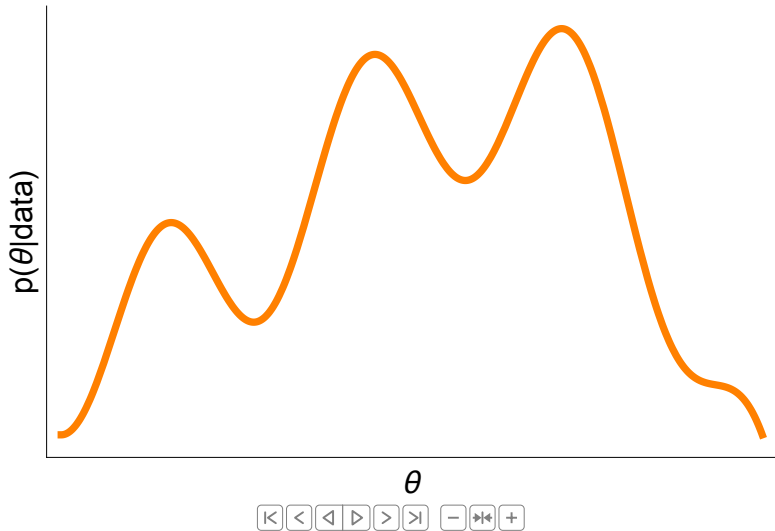
Start with the below distribution and try each different stepping rule.



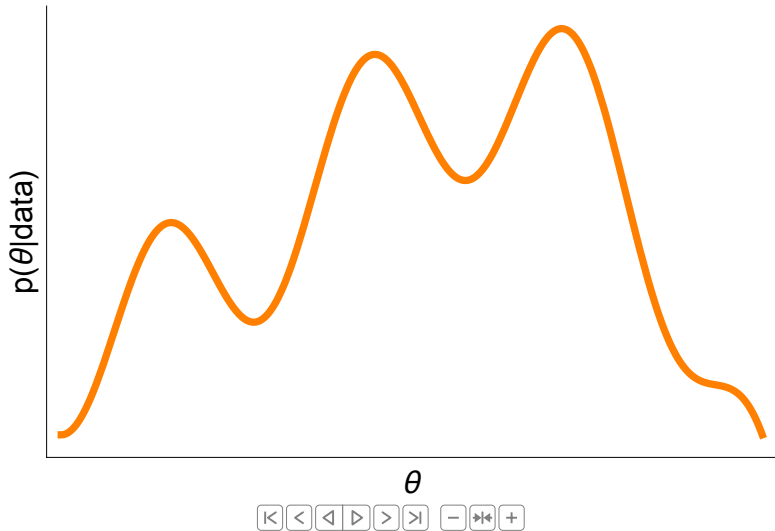
Drunkard's walk



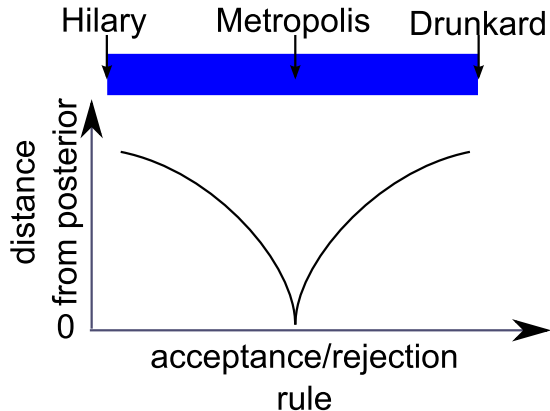
Hilary's ascent



Random Walk Metropolis



Accept reject rule: summary



Only the Metropolis accept/reject rule allows sampling from each point in exact proportion to the posterior height.

The intuition behind Random Walk Metropolis

Consider the ratio of the posterior density at point θ_{t+1} to θ_t :

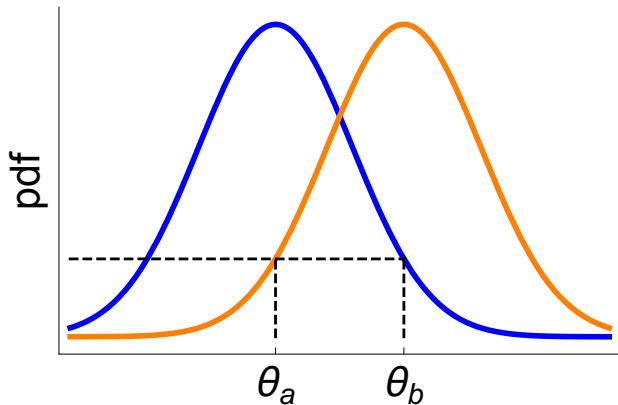
$$\begin{aligned} r &= \frac{p(\theta_{t+1}|X)}{p(\theta_t|X)} \\ &= \frac{\frac{p(X|\theta_{t+1})p(\theta_{t+1})}{p(X)}}{\frac{p(X|\theta_t)p(\theta_t)}{p(X)}} \\ &= \frac{p(X|\theta_{t+1})p(\theta_{t+1})}{p(X|\theta_t)p(\theta_t)} \end{aligned}$$

So the ratio of the numerators of Bayes' rule is **identical** to the ratio of the actual posteriors.

$\implies$ if we use r to guide our stepping we will be sampling (eventually) from the posterior.

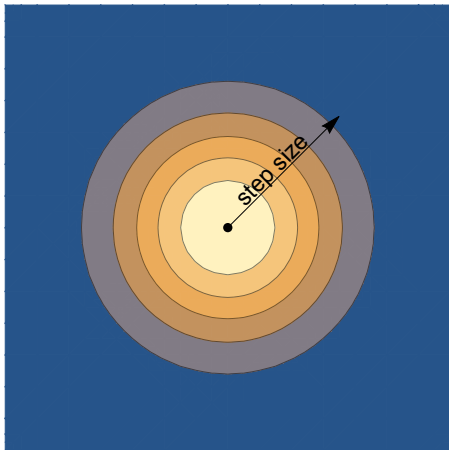
How do we choose the jumping distribution?

- Sometimes called the “proposal distribution”.
- In Random Walk Metropolis we use a symmetric distribution (relaxed in Metropolis-Hastings):
 $\implies J(\theta_a|\theta_b) = J(\theta_b|\theta_a)$



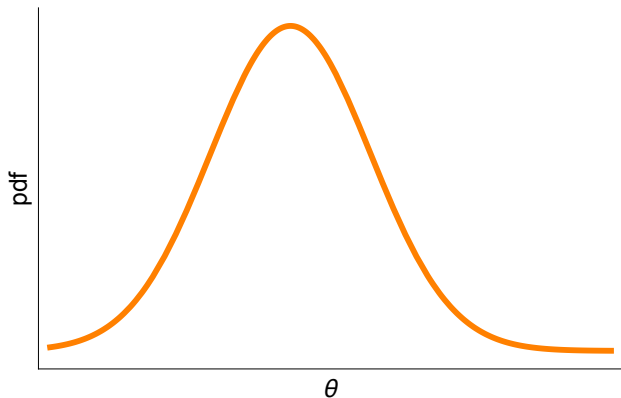
The importance of step size

Question: how should we decide on the jumping kernel's step size?



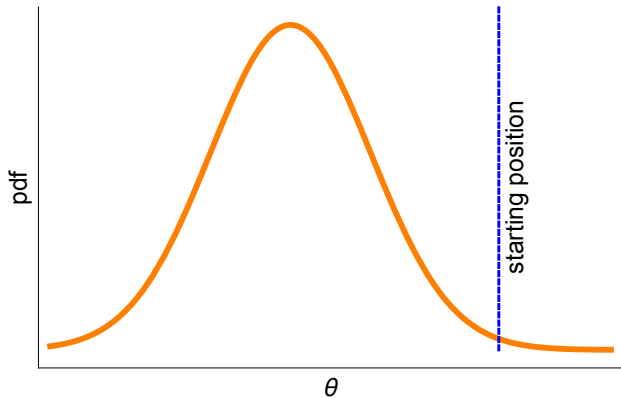
Another example posterior distribution

Assume a unimodal distribution from which we want to sample.

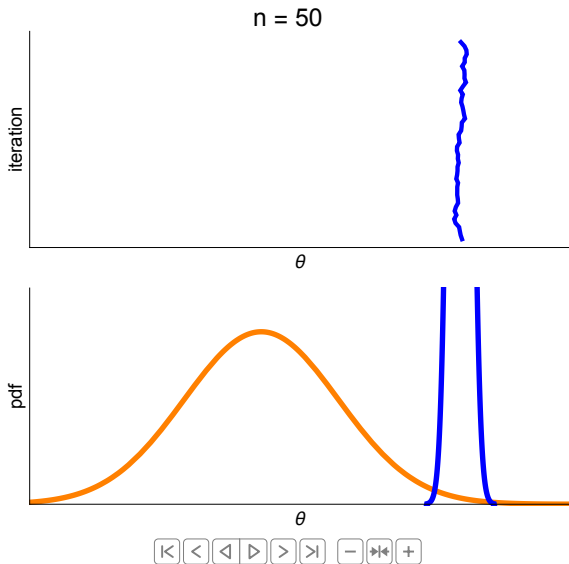


Another example posterior distribution

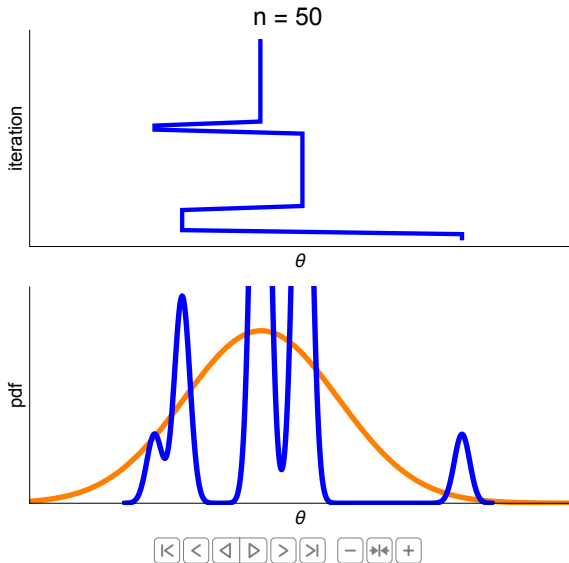
Start three algorithms with different step sizes at same point.



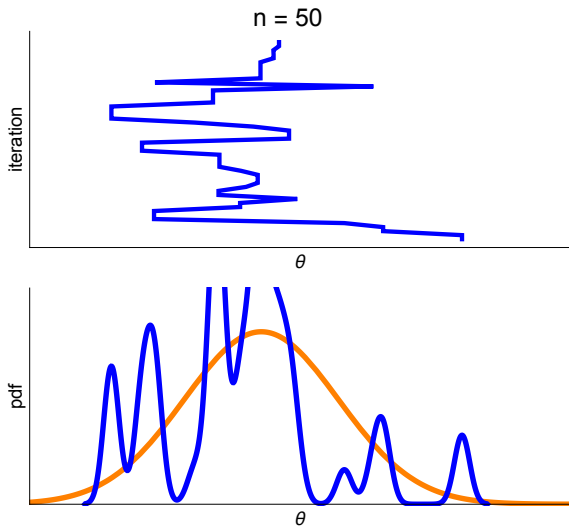
The importance of step size: too small



The importance of step size: too large



The importance of step size: just right



Step size: summary

- Whilst step size does not affect asymptotic convergence, it does affect finite sample performance.
- If step size is too small we do not find the typical set (area of high probability mass).
- If step size is too large we find the typical set, but do not explore it efficiently.
- Therefore do an initial run of sampler to find optimal step size before starting proper.

- 1 Recap from last lecture
- 2 Understanding a distribution by sampling from it
- 3 Introducing dependent sampling
- 4 Random Walk Metropolis
- 5 Judging convergence of chains to posterior**
- 6 Effective sample size

What do we mean by convergence?

Recap the steps of Metropolis:

- ① Propose an initial position θ_0 using a initial proposal distribution $\pi(\theta) \neq p(\theta|X)$.
- ② For $t = 1, \dots, T$ do:
 - Propose a new location: $\theta_{t+1} \sim J(\theta_{t+1}|\theta_t)$.
 - Accept/reject move based on

$$r = \frac{p(X|\theta_{t+1})p(\theta_{t+1})}{p(X|\theta_t)p(\theta_t)} > u \sim \text{Unif}(0, 1) \quad (5)$$

What do we mean by convergence?

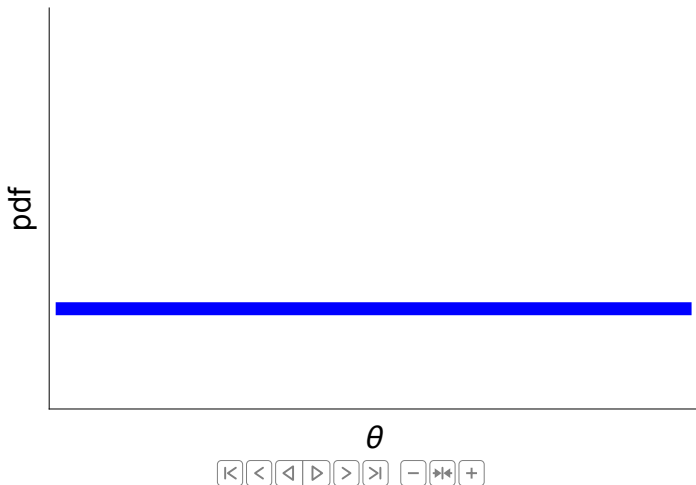


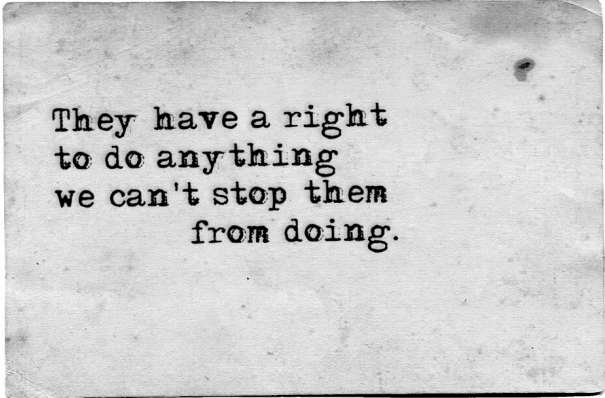
Figure: Adapted from Betancourt lecture:
www.youtube.com/watch?v=pHsuIaPbNbY.

Why do we need to monitor convergence?

- Start with an initial proposal distribution $\pi(\theta) \neq p(\theta|X)$.
- Repeatedly take steps and use the Metropolis accept/reject rule $\implies \pi(\theta_t)$; the sampling distribution at time t .
- Under a set of quite general assumptions we are guaranteed that asymptotically: $\pi(\theta_t) \rightarrow p(\theta|X)$.
- However, when practically can we assume:
 $\pi(\theta_t) \approx p(\theta|X)$?

How to measure convergence?

- To monitor convergence to the posterior $\implies$ need the posterior.
- But we don't have the posterior $\Leftarrow$ the reason we are doing the sampling in the first place!



They have a right
to do anything
we can't stop them
from doing.

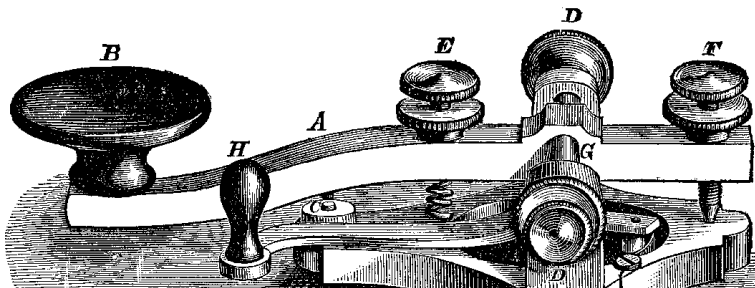
Two strategies for monitoring convergence

Strategy 1: measure distributional separation.

- For example Kullback-Leibler:

$$KL = \int p(\theta|X) \log \left(\frac{p(\theta|X)}{\pi(\theta_t)} \right) d\theta \quad (6)$$

- Motivated by information theory.
- Can use un-normalised posterior to do this.
- Again integral is too difficult to do.



Two strategies for monitoring convergence

Strategy 2: monitor the approach to a stationary distribution.

- We know asymptotically this will happen.
- By design of Metropolis stepping and accept/reject rules, we know the stationary distribution is the posterior.



Monitoring convergence of a single chain

Initial idea:

- Compare summaries (mean, variance, etc.) of sampling distribution for a chain at time t with itself at time $t + T$.
- If their rate of change is below a threshold $\implies$ convergence.

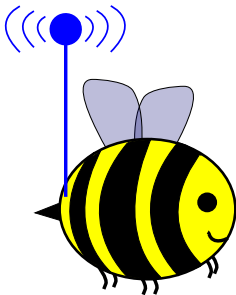
Monitoring convergence of a single chain

Question: What is the problem with this idea?

Convergence monitoring: Bob's bees

Thought experiment:

- Imagine a house of unknown shape.
- We have an unlimited supply of bees, each equipped with a GPS tracker allowing us to accurately monitor their position.
- **Question:** How can we use these to estimate the shape of the house?



Convergence monitoring: Bob's bees

Answer:

- Release one (at a random location in the house) and monitor its path over time.
- Stop/collect bee after summary measures of its path stop changing.

Convergence monitoring: single bee



Convergence monitoring: single bee, a bit later



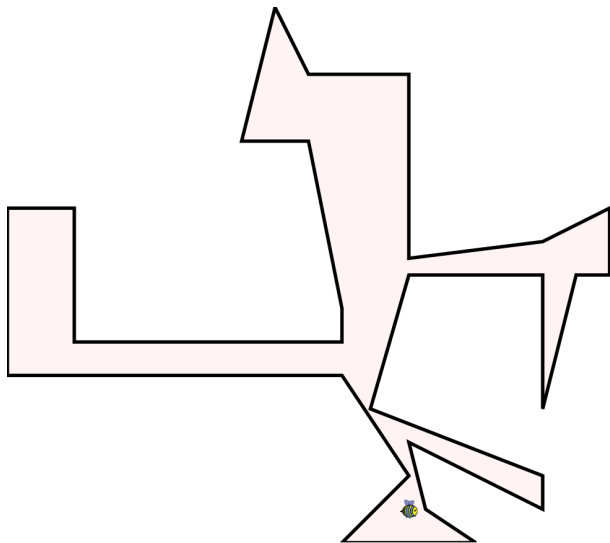
Convergence monitoring: single bee, a bit bit later



Convergence monitoring: single bee

Question: what's the actual shape of the house?

Convergence monitoring: single bee

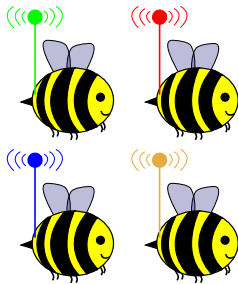


Single chain problems: summary

- One way to monitor convergence is to look for convergence in a single chain's summary statistics.
- This method is very susceptible to the curse of hindsight problem ("Now we've definitely converged on the posterior. We hadn't a minute ago.")
- Particularly because chains often get stuck in subregions of θ space.

The solutions: lots of bees

- Release lots of bees starting at dispersed locations in parameter space.
- Stop recording when an individual bee's path is indistinguishable from all others'.



Convergence monitoring: multiple bees



Convergence monitoring: multiple bees (a lot later)



Multiple chain convergence monitoring: summary

- Start a number of chains in random dispersed locations in θ space.
- Chains do *not* interact with one another (in Metropolis).
- Run each sampler until it is hard to distinguish one chain's path from all others'.
- Less susceptible to “curse of hindsight”, since we can see if chains aren't mixing.
- Not foolproof! There still may be an area of high probability mass that we miss. However, less likely to fail compared to a single chain.
- The more chains, the better!

Multiple chain convergence monitoring: open questions

- ① How to determine “random dispersed locations” at which to start the chains?
 - Ideally use an initial proposal distribution similar to posterior shape.
 - Otherwise a good rule of thumb is “Any point you don’t mind having in a sample is a good starting point”, Charles Geyer.
- ② Which summary statistics to monitor to determine convergence?
- ③ At what threshold are “between chain” statistics sufficiently similar?

Gelman and Rubin's $\hat{R}$

- Gelman and Rubin (1992) had the idea of comparing within-chain to between-chain variability.
- They quantified this comparison using:

$$\hat{R} = \sqrt{\frac{W + \frac{1}{n}(B - W)}{W}} \quad (7)$$

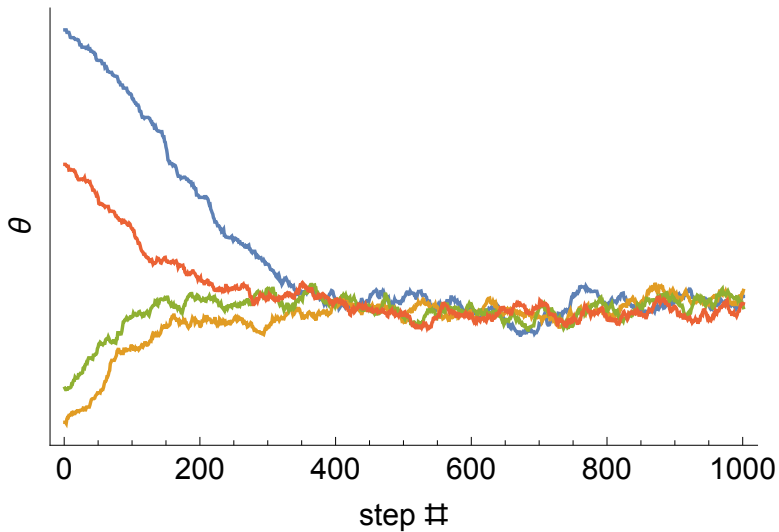
- Where “within-chain” variability, $W = \frac{1}{m} \sum_{j=1}^m s_j^2$, for m chains.
- And “between-chain” variability, $B = \frac{n}{m-1} \sum_{j=1}^m (\bar{\theta}_j - \bar{\theta})^2$.
- When we start $B \gg W$ since we start in an overdispersed position.
- In convergence $B \rightarrow W \implies \hat{R} \rightarrow 1$ (in practice $\hat{R} < 1.1$ usually suffices).

Warm up period

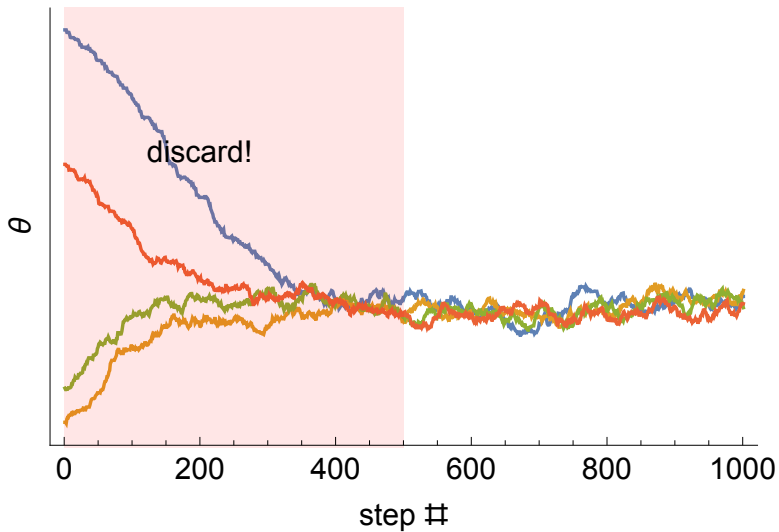
- The initial proposal distribution is *not* the posterior.
- We therefore discard the beginning part of the chain called the “warm up” to lessen the effect of the starting position.
- Typically discard first half of converged chains (can also cut chains in two to monitor intra-chain convergence).



Warm up period



Warm up period



- 1 Recap from last lecture
- 2 Understanding a distribution by sampling from it
- 3 Introducing dependent sampling
- 4 Random Walk Metropolis
- 5 Judging convergence of chains to posterior
- 6 Effective sample size**

Effective sample size

- Ideally we would have *independent* samples from the posterior.
- Although in general independent sampling is not possible.
- Instead we use *dependent* sampling, where the next sample depends on the current value.
- Dependence $\implies$ less information per each additional sample.

A definition of effective sample size

“The equivalent number of samples for an independent sampler”.

Note: the worth of an algorithm is the # effective samples per second *not* # samples per second!

Measuring the cost of dependent sampling

- Create effective sample size (ESS) measure such that as dependence $\uparrow \implies \text{ESS} \downarrow$
- One way to measure dependence is by the autocorrelation of the sampler's value.

$$\text{ESS}(\theta_i) = \frac{mT}{1 + 2 \sum_{\tau=1}^{\infty} \rho_{\tau}(\theta_i)} \quad (8)$$

Where m is the number of chains, and T is the number of samples *per* chain, and ρ_{τ} is the τ th order autocorrelation for θ_i .

Summary

- ① Sampling can be used to gain insight into a distribution.
- ② Independent sampling from posterior not generally possible
 $\implies$ shift to dependent sampling.
- ③ Random Walk Metropolis is a MCMC algorithm that allows *dependent* sampling from the posterior.
- ④ The efficiency of Metropolis depends on choosing the right step size.
- ⑤ Monitoring of sampler's convergence to the posterior is non-trivial.
- ⑥ The use of multiple chains makes it harder to make a mistake although not impossible.
- ⑦ We value $\#$ effective samples per second, *not* $\#$ samples.

Worth bearing in mind

“MCMC is most definitely not a panacea and should be used with caution, almost as a last resort. It is just that we are at the last resort for interesting models.”

— Chris Holmes

“An approximate answer to the right problem is worth a good deal more than an exact answer to an approximate problem.”

— John Tukey

Worth bearing in mind

- For some simple models we may still be able to do exact inference.
- For bigger models characterised by multi-modal posteriors
⇒ sampling does not work; so use other methods
(variational Bayes, for example.)

Reading list

Steady eddies:

- Chapters 10 (sampling) and 11 (Metropolis and Gibbs) from “Bayesian data analysis”, by Gelman et al. (2014), 3rd edition.
- Chapters 3 (sampling) and 4 (linear models) from “Statistical Rethinking” by McElreath.
- Chapter 7 (MCMC) from “Doing Bayesian data analysis”, by Kruschke, 2nd edition.

Reading list

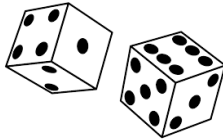
Eddie the Eagle:

- “Monte Carlo is fundamentally unsound”, O’Hagan (1987), *The Statistician*.
- “Bayesian Monte Carlo”, Rasmussen and Ghahramani (2002), *Advances in neural information processing systems*.
- “Methods for approximating integrals in statistics with special emphasis on Bayesian integration problems”, Evans and Swartz (1995), *Statistical Science*.
- “Probabilistic numerics and uncertainty in computations”, Hennig et al. (2015), *Proceedings of the Royal Society A*.
- “Patterns of performance degradation and restoration during sleep restriction and subsequent recovery: A sleep dose-response study”, Belenky et al. (2003), *Journal of sleep research*

Not sure I understand?

Hierarchy of samplers:

MC:



MCMC:



Not sure I understand?

MC ?:

