Bayesian approaches to parameter estimation

CSCS 530 - Marisa Eisenberg

Bayesian approaches to parameter estimation

- Bayes' Theorem, rewritten for inference problems:

$$P(p \mid z) = P(params \mid data) = \frac{P(z \mid p) \cdot P(p)}{P(z)}$$

- Allows one to account for prior information about the parameters
  - E.g. previous studies in a similar population
- Update parameter information based on new data

Bayesian approaches to parameter estimation

Likelihood

Prior distribution

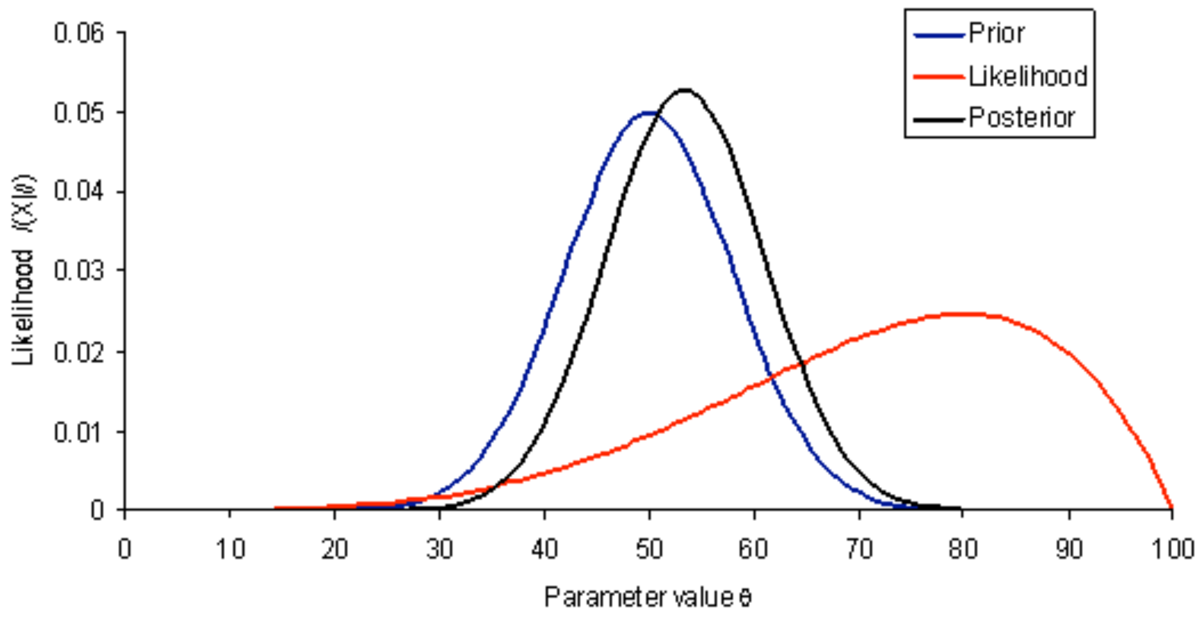
$$P(p \mid z) = P(params \mid data) = \frac{P(z \mid p) \cdot P(p)}{P(z)}$$

Normalizing constant
(can be difficult to calculate!)

$$P(z) = \int_p P(z,p)dp$$

Bayesian Parameter Estimation

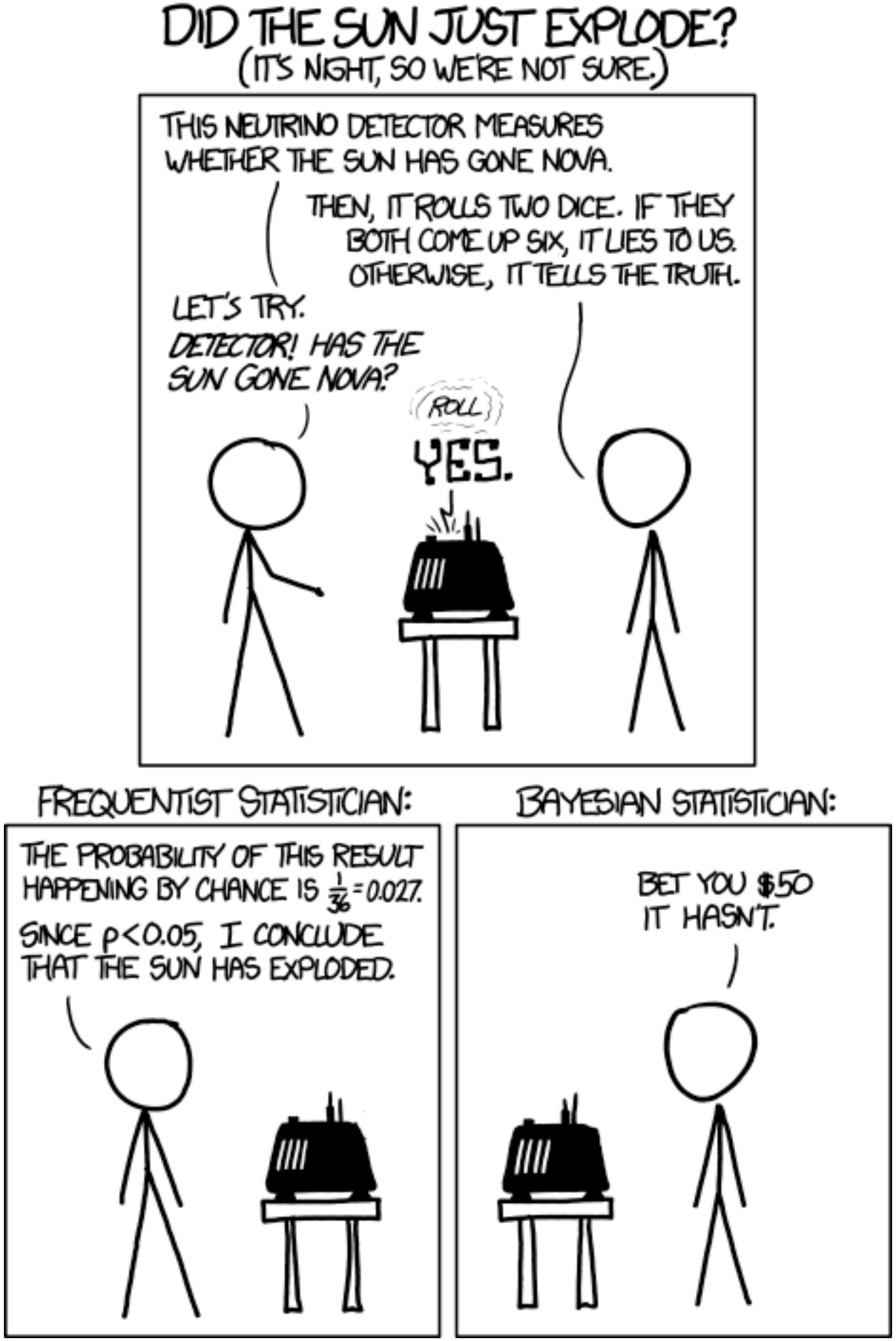
- From prior distribution & likelihood distribution, determine the posterior distribution of the parameter



- Can repeat this process as new data is available

Bayesian Parameter Estimation

- Treats the parameters inherently as distributions (belief)
- Philosophical battle between Bayesian & frequentist perspectives


DID THE SUN JUST EXPLODE?
(IT'S NIGHT, SO WE'RE NOT SURE.)
THIS NEUTRINO DETECTOR MEASURES WHETHER THE SUN HAS GONE NOVA.
THEN, IT ROLLS TWO DICE. IF THEY BOTH COME UP SIX, IT LIES TO US. OTHERWISE, IT TELLS THE TRUTH.
LET'S TRY.
DETECTOR! HAS THE SUN GONE NOVA?
ROLL
YES.
FREQUENTIST STATISTICIAN:
THE PROBABILITY OF THIS RESULT HAPPENING BY CHANCE IS 1/36 = 0.027.
SINCE p<0.05, I CONCLUDE THAT THE SUN HAS EXPLODED.
BAYESIAN STATISTICIAN:
BET YOU $50 IT HASN'T.

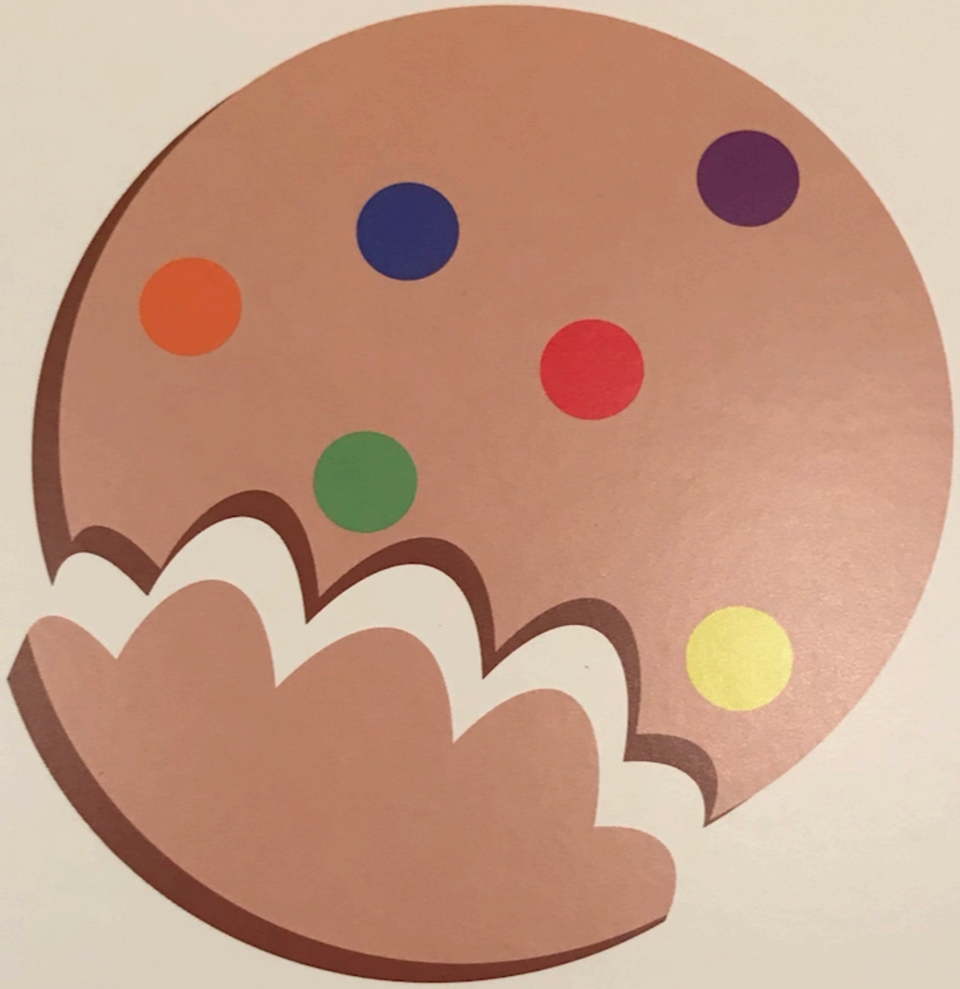
from XKCD:
http://xkcd.com/1132/

Bayesian probability for babies!

Some cookies have candy.

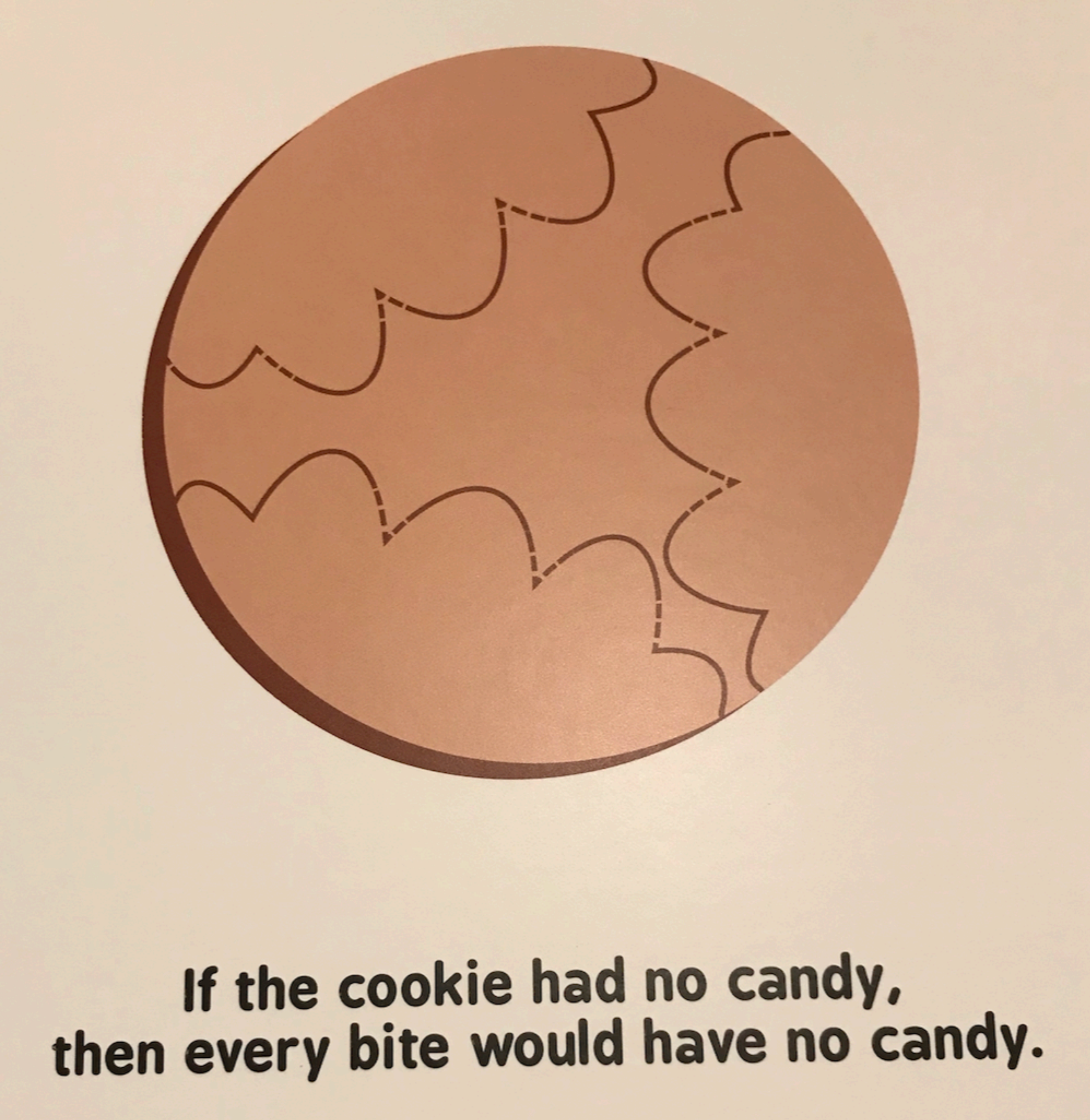
Some don't.

Take a bite. It has no candy.

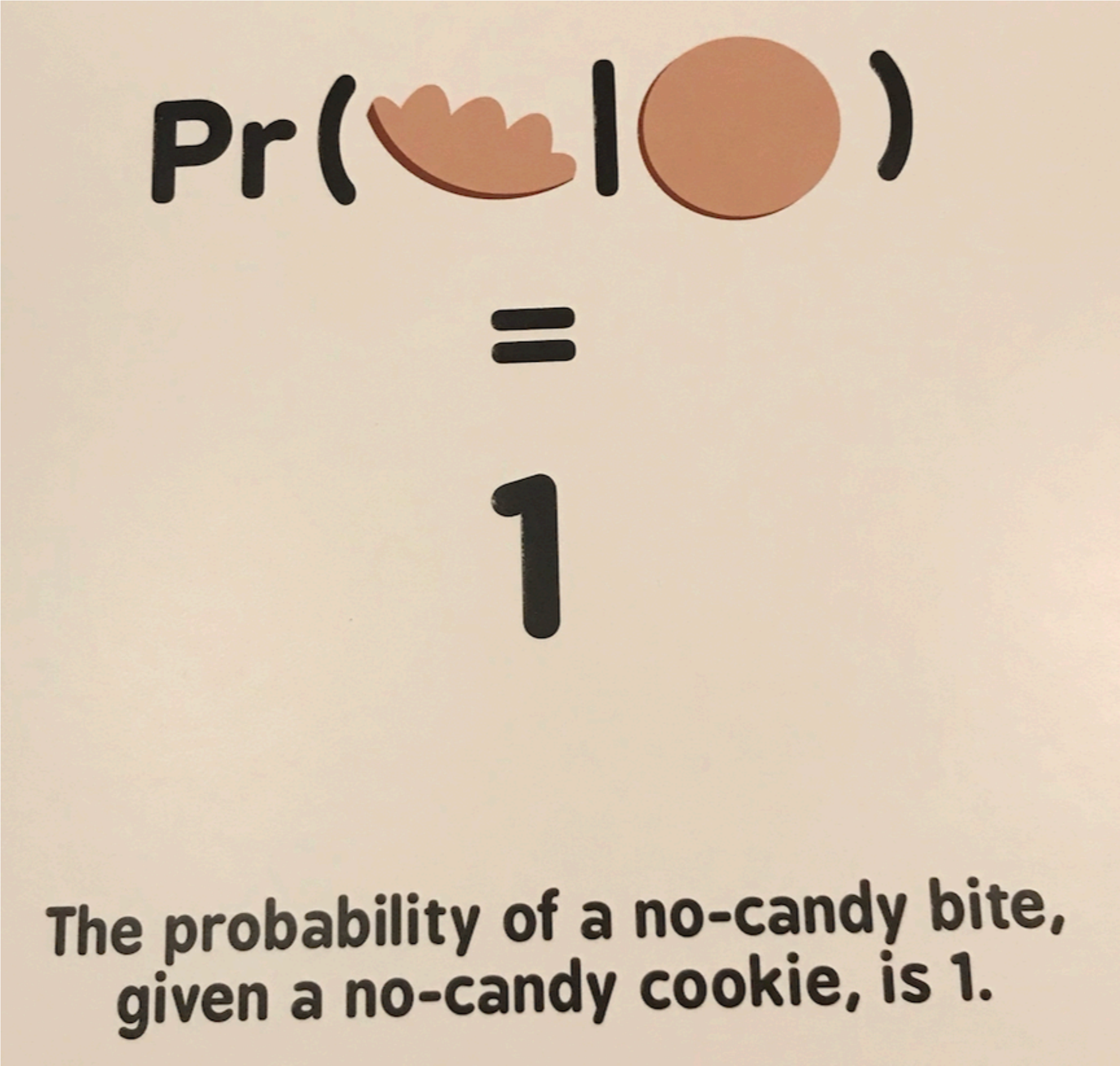


Did it come from a candy cookie?

What is the likelihood?
L(NC cookie | NC bite) = P(NC bite | NC cookie)



If the cookie had no candy,
then every bite would have no candy.



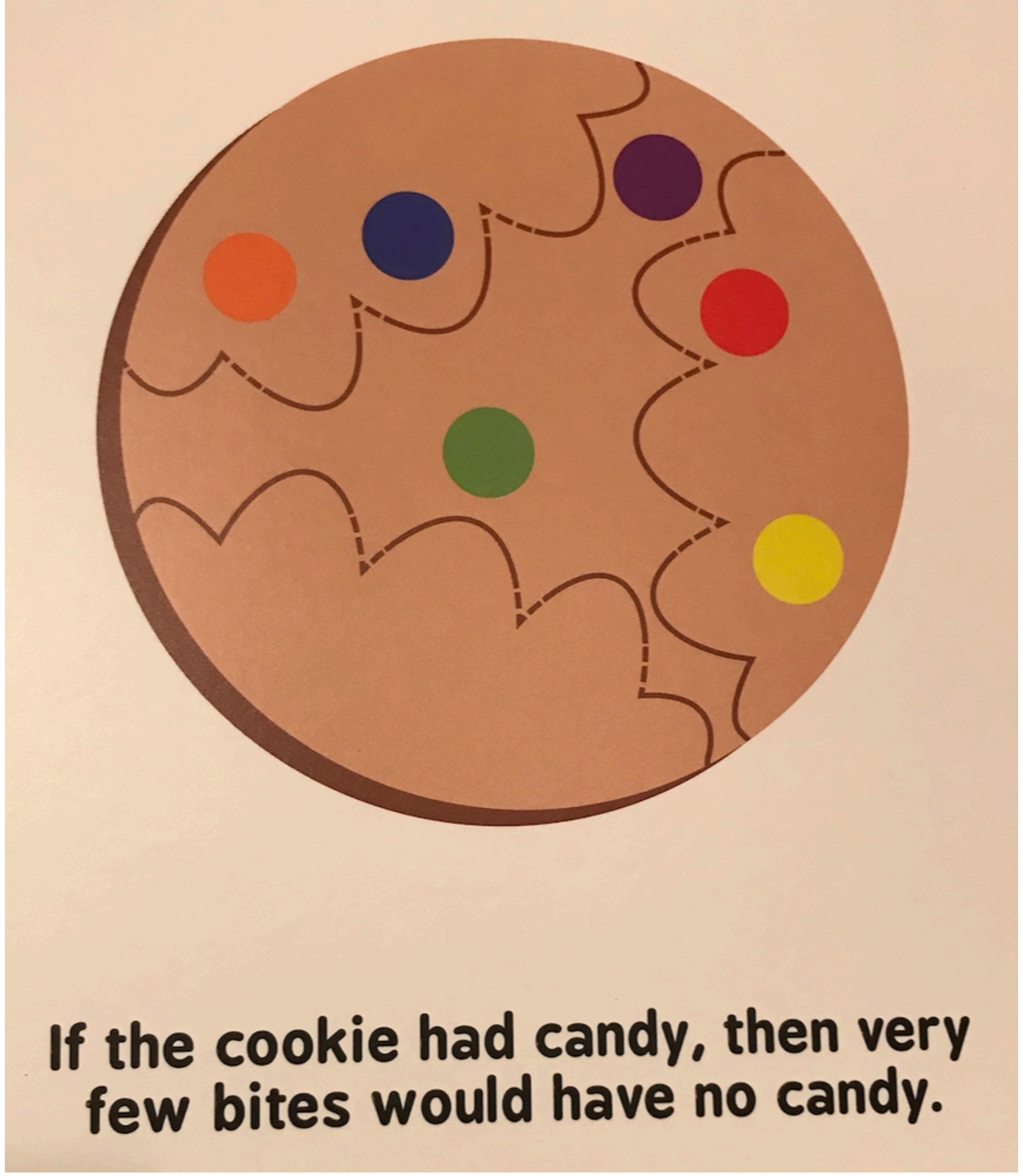
Pr(◡|●)

=

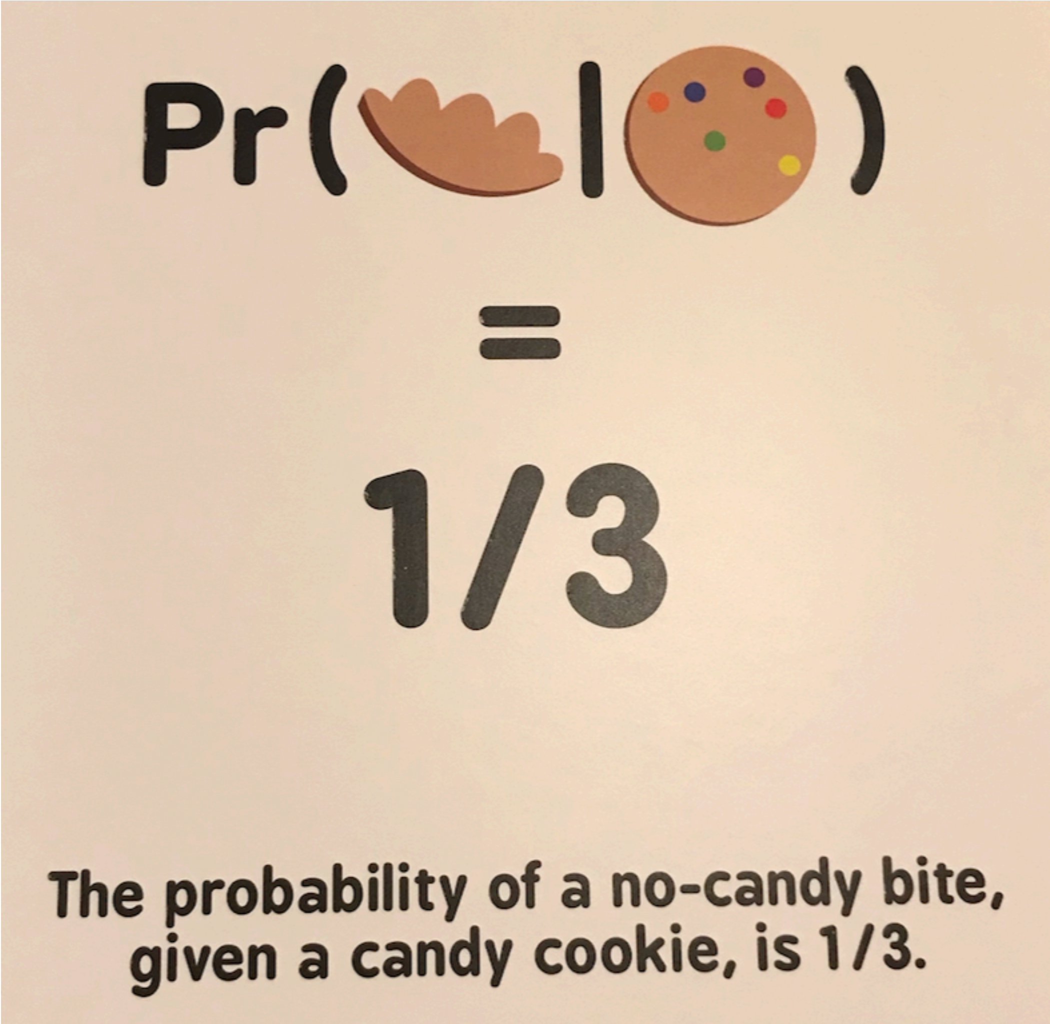
1

The probability of a no-candy bite,
given a no-candy cookie, is 1.

What is the likelihood?
L(C cookie | NC bite) = P(NC bite | C cookie)



If the cookie had candy, then very few bites would have no candy.



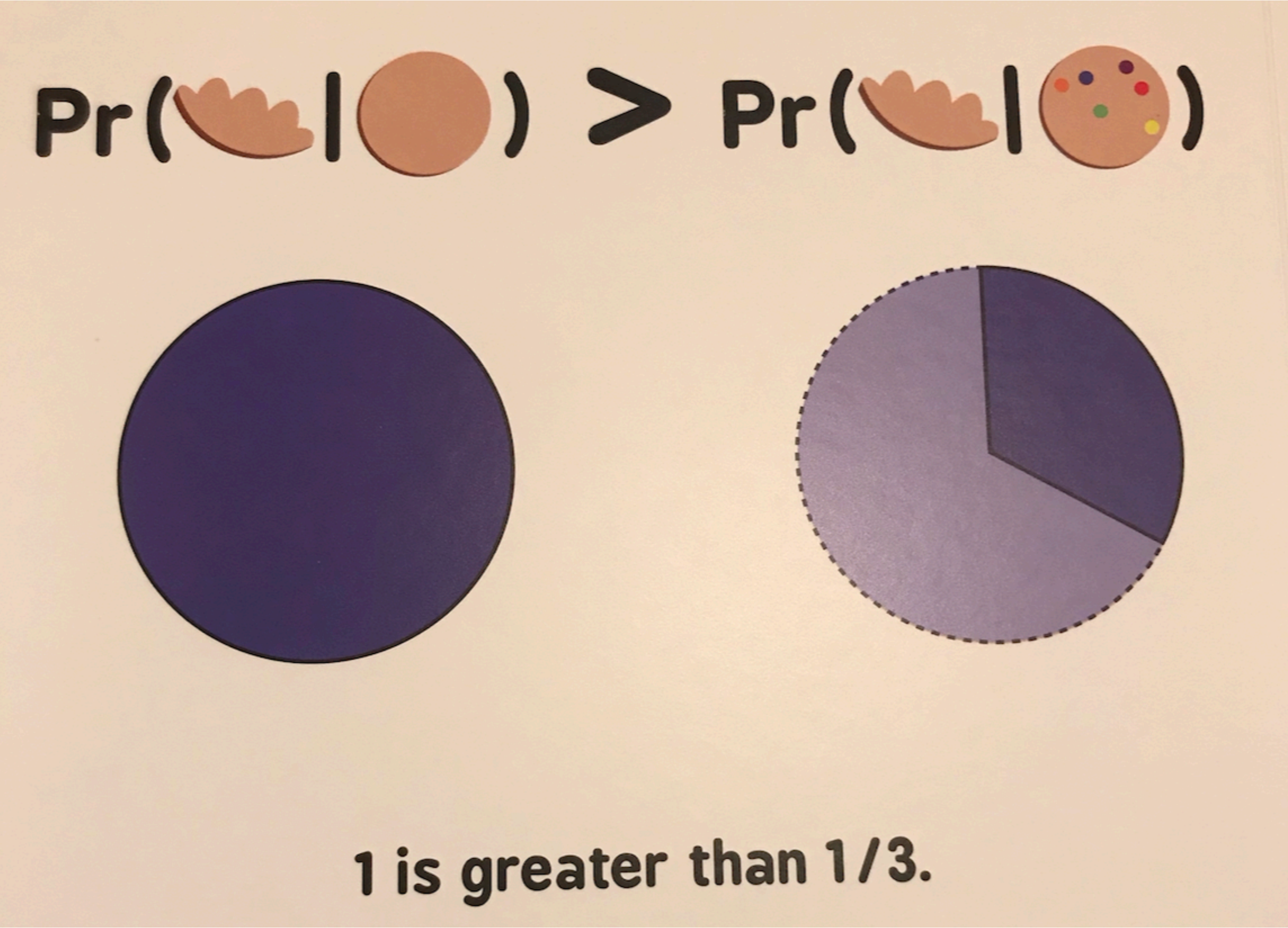
Pr(|)

=

1/3

The probability of a no-candy bite, given a candy cookie, is 1/3.

What is the maximum likelihood estimate?



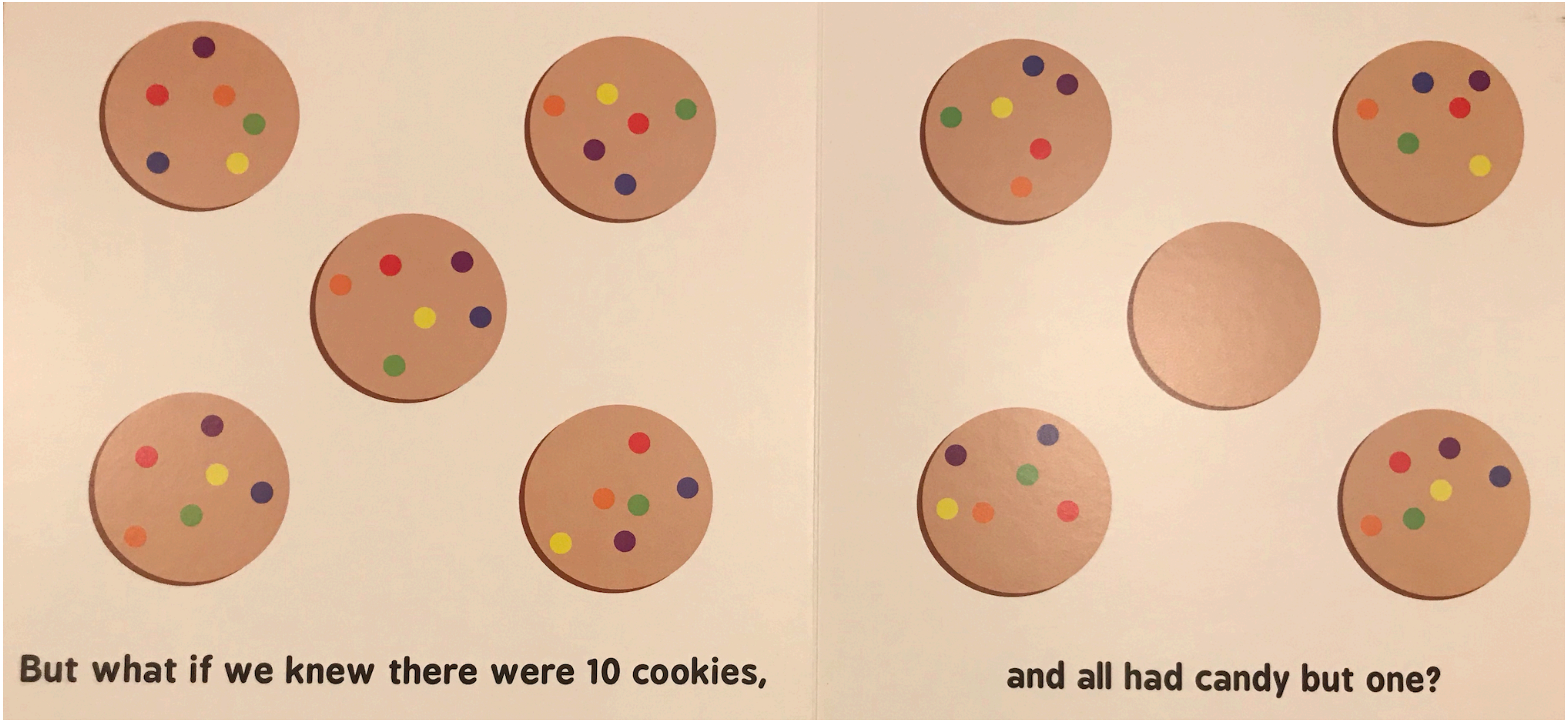
Pr(|) > Pr(|)

1 is greater than 1/3.

Maximum likelihood:

So the no-candy bite probably came from a no-candy cookie!

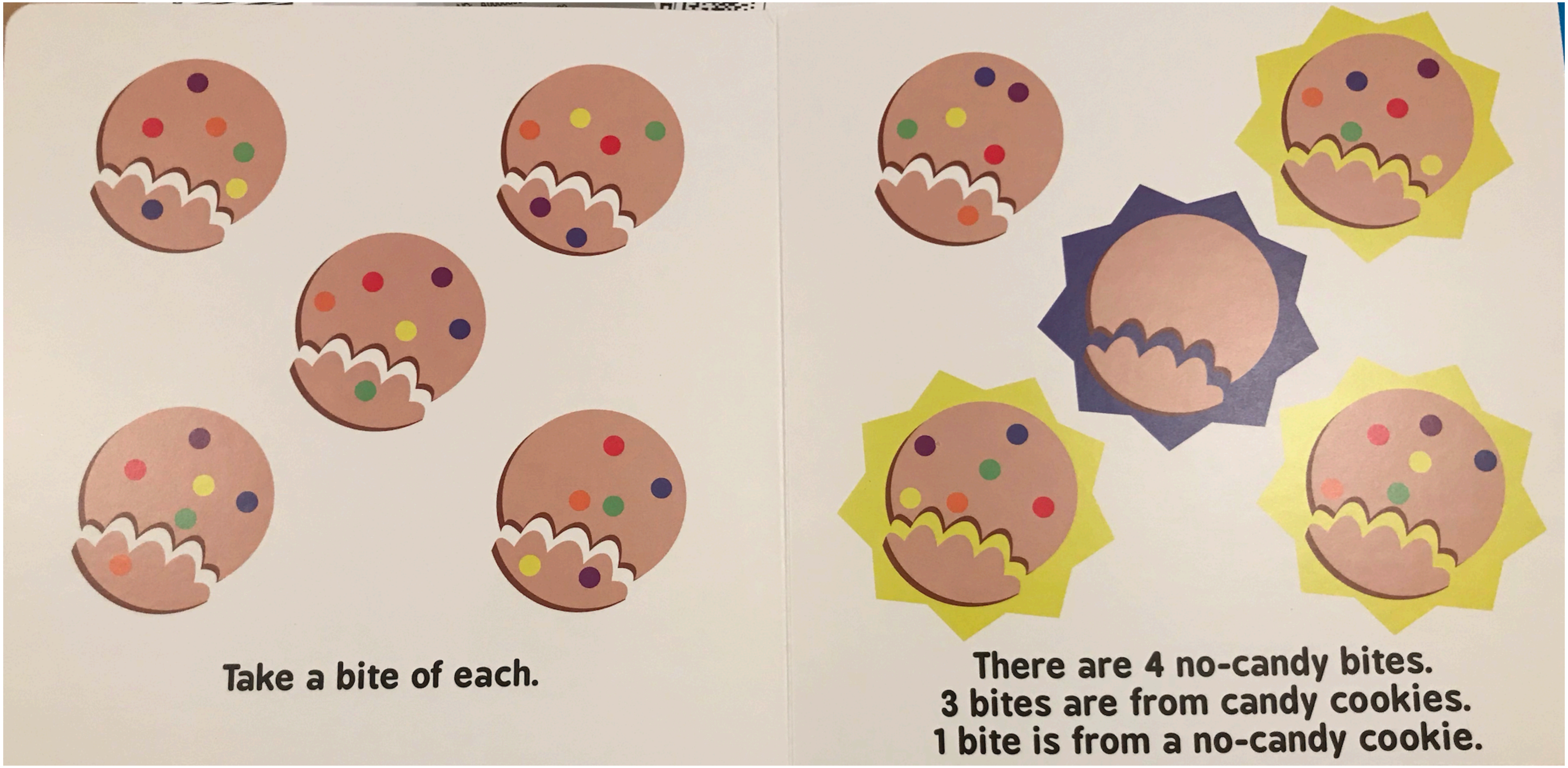
What about the prior distribution of cookies?



But what if we knew there were 10 cookies,

and all had candy but one?

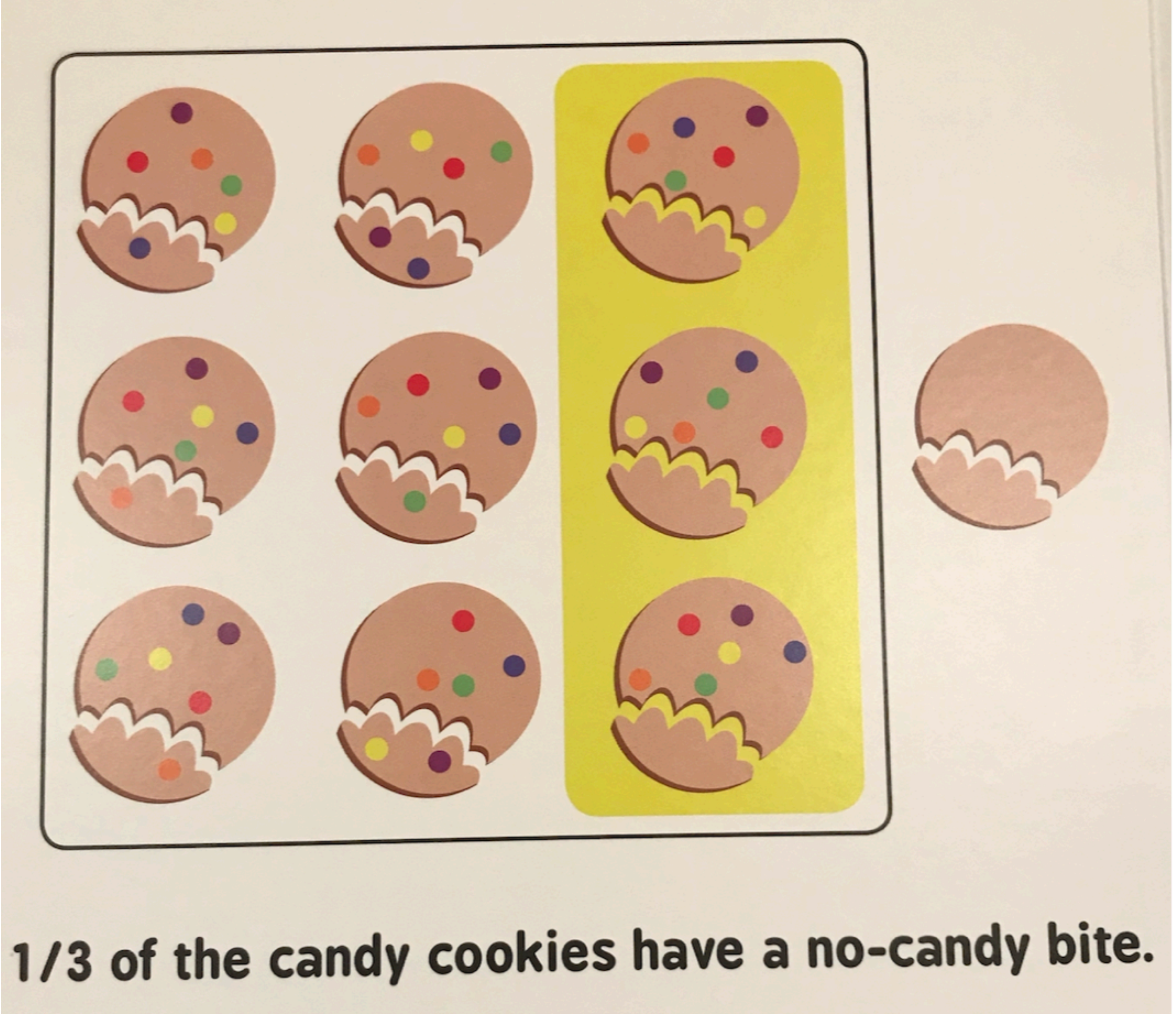
Our data (likelihood) tells us we have a no-candy bite—how many of the bites are no candy?



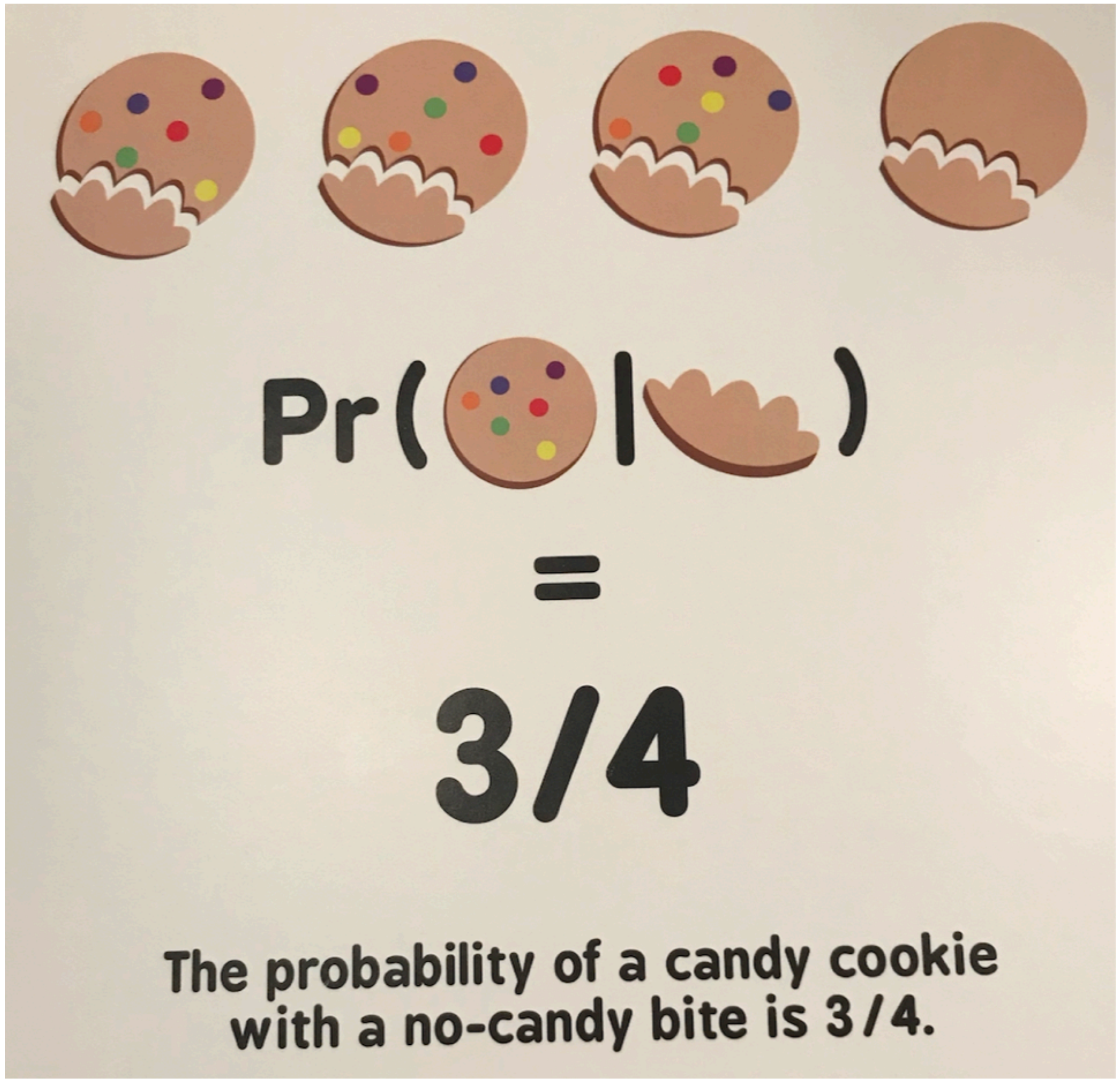
Take a bite of each.

There are 4 no-candy bites.
3 bites are from candy cookies.
1 bite is from a no-candy cookie.

1/3 of the candy cookie bites have no candy, but there are a lot more of them



1/3 of the candy cookies have a no-candy bite.

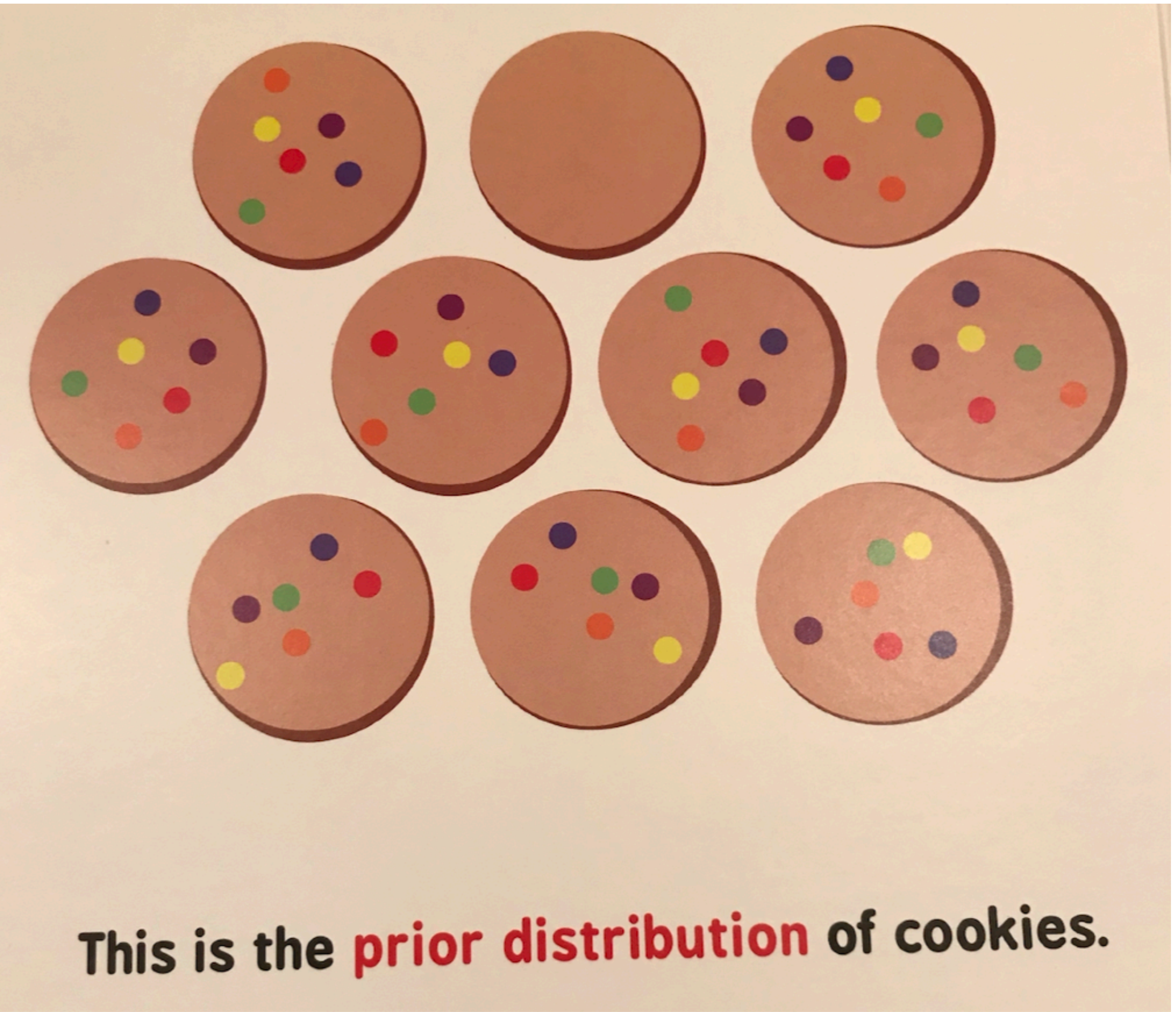


The probability of a candy cookie with a no-candy bite is 3/4.

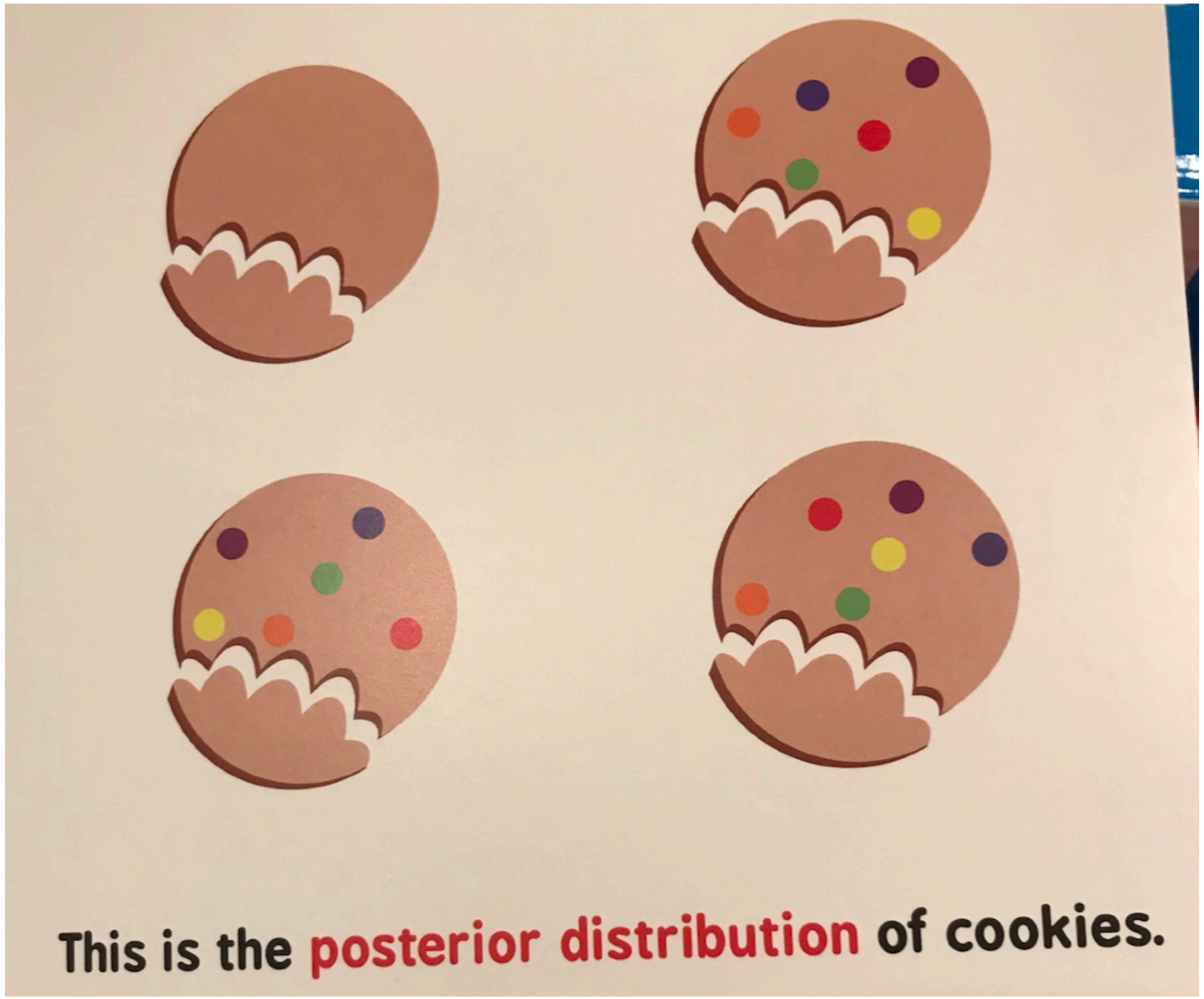
Prior x Likelihood ~ Posterior

9 x 1/3 = 3 candy cookies, vs. 1 x 1 = 1 no-candy cookie

Bayesian estimation!



This is the prior distribution of cookies.



This is the posterior distribution of cookies.

Bayesian approaches to parameter estimation

Likelihood

Prior distribution

$$P\left(p \mid z\right) = P\left(params \mid data\right) = \frac{P\left(z \mid p\right) \cdot P\left(p\right)}{P\left(z\right)}$$

Normalizing constant
(can be difficult to calculate!)

$$P(z) = \int_p P(z,p)dp$$

Bayesian Parameter Estimation

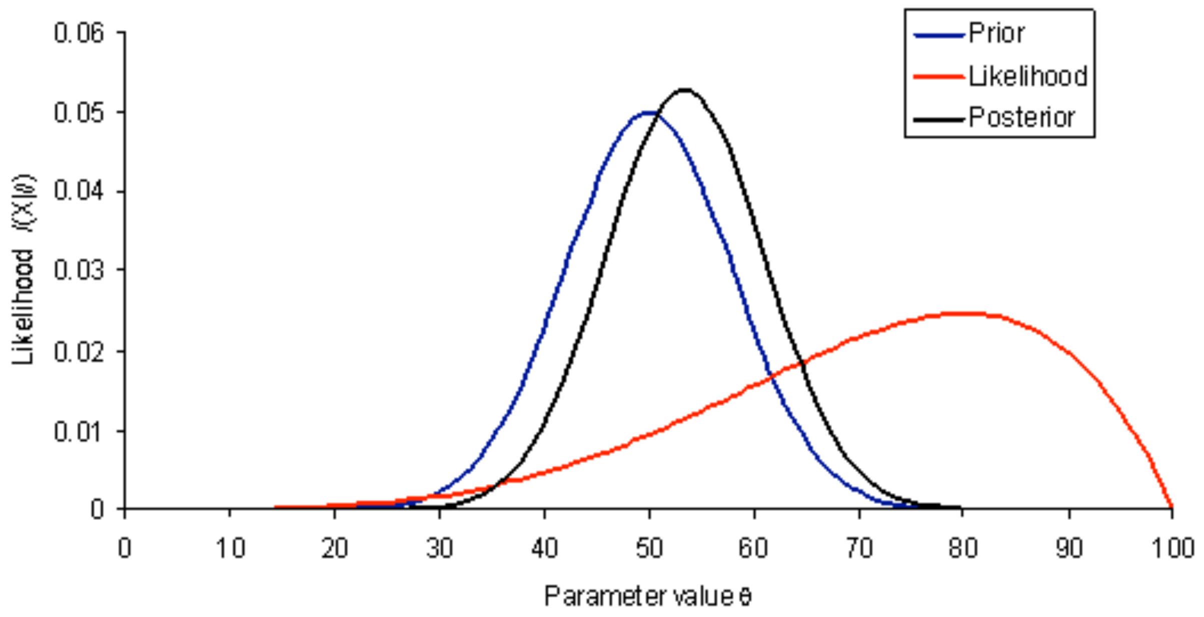
- Can think of Bayesian estimation as a map, where we update the prior to a new posterior based on data

$$P(p) \longrightarrow \boxed{\times \frac{P(z|p)}{P(z)}} \longrightarrow P(p|z)$$

Prior

Likelihood/P(z)

Posterior



Denominator term - P(z)

- The denominator term:

$$P(z) = \int_p P(z,p)dp$$

- Probability of seeing the data z from the model, over all parameter space

- Often doesn't have a closed form solution—evaluating numerically can also be difficult

    - E.g. if p is a three dimensional, then if we took 1000 grid points in each direction, the grid representing the function to be integrated has $1000^3 = 10^9$ points

Maximum *a posteriori* (MAP) estimation

- Instead of working with the full term, just use the numerator:

$$P(p|z) = \frac{P(z|p) \cdot P(p)}{P(z)}$$

- The denominator is a constant, so the numerator is proportional to the posterior we are trying to estimate

- Then the ***p*** which yields max($P(z|p) \cdot P(p)$) is the same ***p*** that maximizes $P(p|z)$

- If we only need a point estimate, MAP gets around needing to estimate $P(z)$

Conjugate Priors

- For some likelihood distributions, there can be a distribution family for our prior that makes the posterior and prior come from the same type of distribution

- This is called a **conjugate prior** for that likelihood

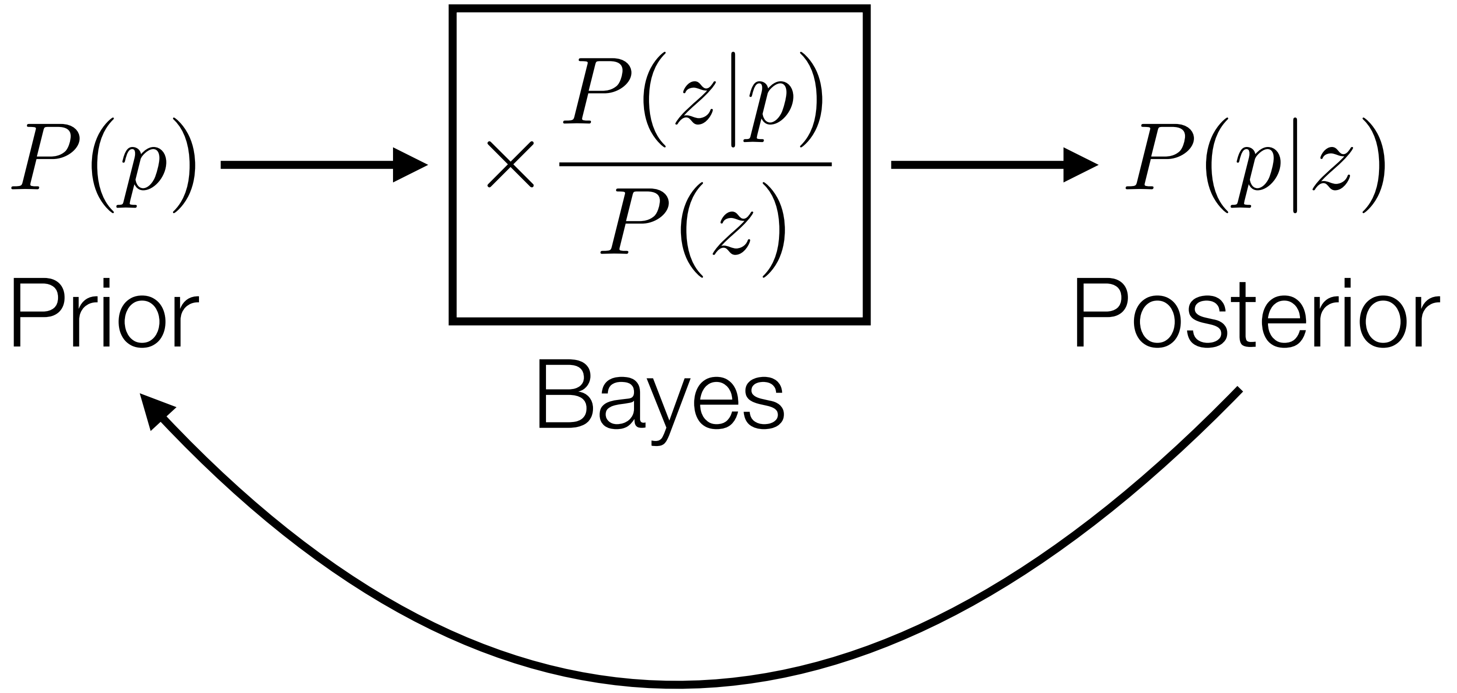
- For example, a gamma distribution is the conjugate prior for a Poisson likelihood.

$$\begin{tikzcd}
P(p) \arrow[r] & \boxed{\times \dfrac{P(z|p)}{P(z)}} \arrow[r] & P(p|z) \\
\text{Gamma} & \text{Poisson} & \text{Gamma}
\end{tikzcd}$$

Why conjugate priors?

- If we have a conjugate prior, we can calculate the posterior directly from the likelihood and the prior—handles the issue with calculating the denominator P(z)

- Also makes it easier to repeat Bayesian estimation—making the posterior the prior and updating as new data comes in



Conjugate prior example: coin flip

- Let z be the data—i.e. the coin flip outcome, z = 1 if it’s heads, z = 0 if it’s tails

- Let θ be the probability the coin shows heads

- Likelihood: Bernoulli distribution

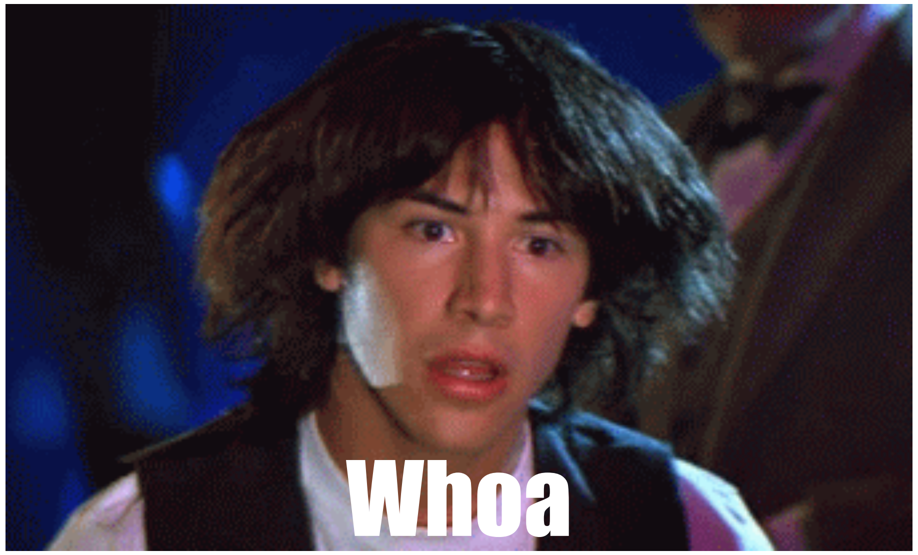
$$P(z|\theta) = \theta^z(1-\theta)^{1-z}$$

Conjugate prior example: coin flip

- **Conjugate prior**: beta distribution

$$P(\theta|\alpha,\beta) = \frac{\theta^{\alpha-1}(1-\theta)^{\beta-1}}{\int_0^1 \theta^{\alpha-1}(1-\theta)^{\beta-1}d\theta}$$

- α and β are **hyperparameters** - shape parameters that describe the distribution of the model parameters



How does the posterior work out to be a beta distribution as well?

$$P(\theta|z) = \frac{P(z|\theta)P(\theta|\alpha,\beta)}{P(z)}$$

$$= \frac{\theta^z(1-\theta)^{1-z}\frac{\theta^{\alpha-1}(1-\theta)^{\beta-1}}{\int_0^1 \theta^{\alpha-1}(1-\theta)^{\beta-1}d\theta}}{P(z)}$$

$$= \frac{\theta^z(1-\theta)^{1-z}\frac{\theta^{\alpha-1}(1-\theta)^{\beta-1}}{\int_0^1 \theta^{\alpha-1}(1-\theta)^{\beta-1}d\theta}}{\int_0^1 P(z,\theta)d\theta}$$

$$= \frac{\theta^z(1-\theta)^{1-z}\frac{\theta^{\alpha-1}(1-\theta)^{\beta-1}}{\int_0^1 \theta^{\alpha-1}(1-\theta)^{\beta-1}d\theta}}{\int_0^1 \theta^z(1-\theta)^{1-z}d\theta}$$

Etc.—but you can see it will work out to be beta distributed

Coin flip example - Posterior

- **Beta distributed** with posterior hyperparameters:

$$\alpha_{post} = \alpha + z \qquad \beta_{post} = \beta + 1 - z$$

- If we take multiple data points, this works out to be:

$$\alpha_{post} = \alpha + \sum_{i=1}^{n} z_i \qquad \beta_{post} = \beta + n - \sum_{i=1}^{n} z_i$$

Sampling methods: approximating a distribution

- What if we want priors that aren’t conjugate? Or what if our likelihood is more complicated and it isn’t clear what the conjugate prior is?

- Now we need some way to get the posterior, even though the denominator term is annoying

- How to approximate the distribution?

Markov Chain Monte Carlo (MCMC)

- Sampling-based methods—in particular, **Markov chain Monte Carlo (MCMC)**

- Also used for many other things! Can approximate distributions more generally—used in cryptography, calculating neutron diffusion, all sorts of things

Markov Chain Monte Carlo (MCMC)

- **MCMC** is a method for sampling from a distribution
- **Markov chain**: a type of (discrete) Markov process
  - Markov: memoryless, i.e. what happens at the next step only depends on the current step
- **Monte Carlo methods** are a class of algorithms that use sampling/randomness—often used to solve deterministic problems (such as approximating an integral)

Markov Chain Monte Carlo (MCMC)

- **Main idea**: make a Markov chain that converges to the distribution we’re trying to sample from—in this case, the posterior distribtuion!
  - The Markov chain will have some transient dynamics (burn-in), and then reach an equilibrium distribution which is the one we’re trying to approximate

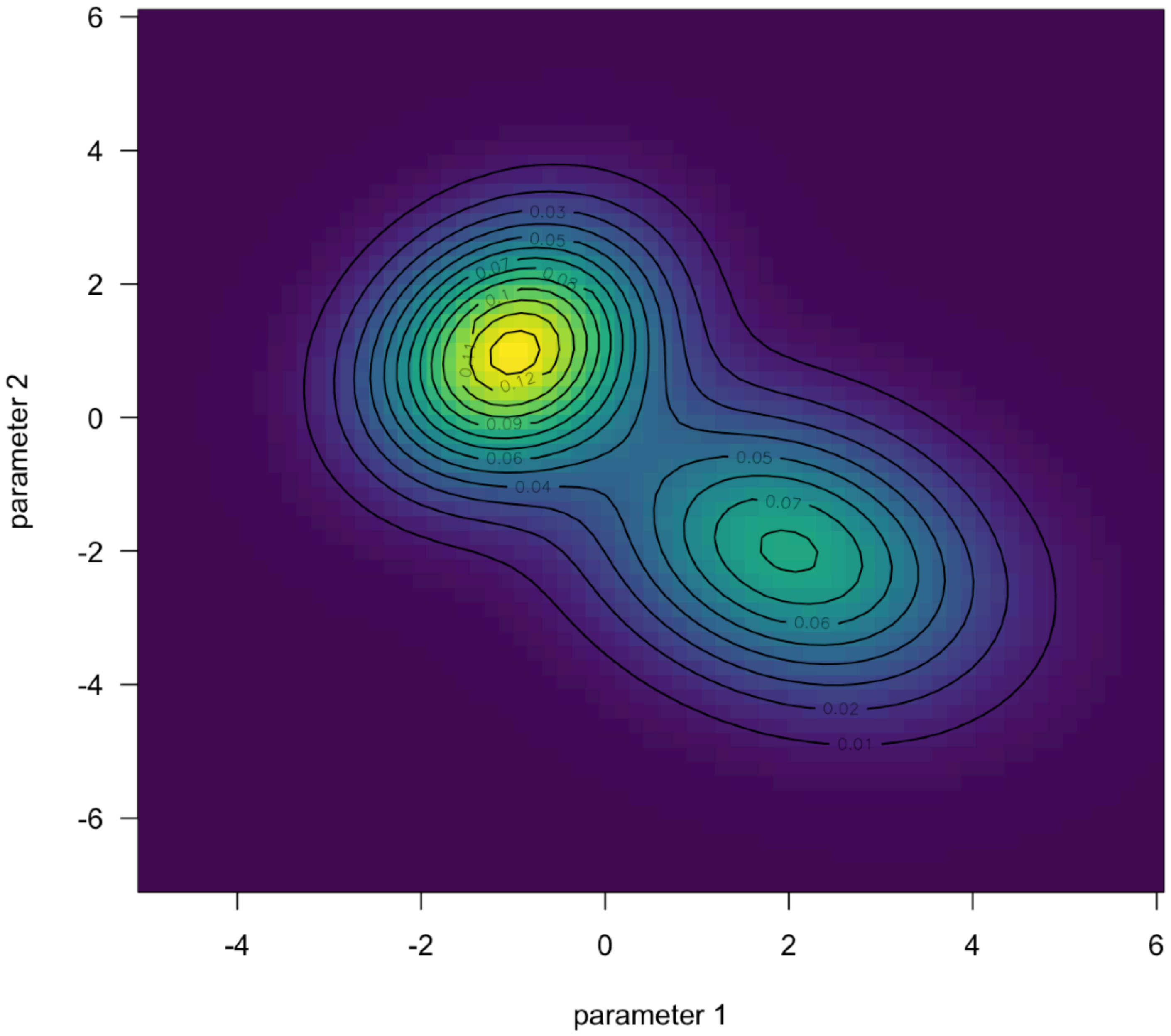
Markov Chain Monte Carlo (MCMC)

- Many MCMC methods are based on random walks
  - Set up walk to spend more time in higher probability regions
- Typically don't need the actual distribution for this, just something proportional—so we can get the relative probability density at two points
  - So we don't need to calculate P(z)! We can just use the numerator

Example adapted from https://nicercode.github.io/guides/mcmc/

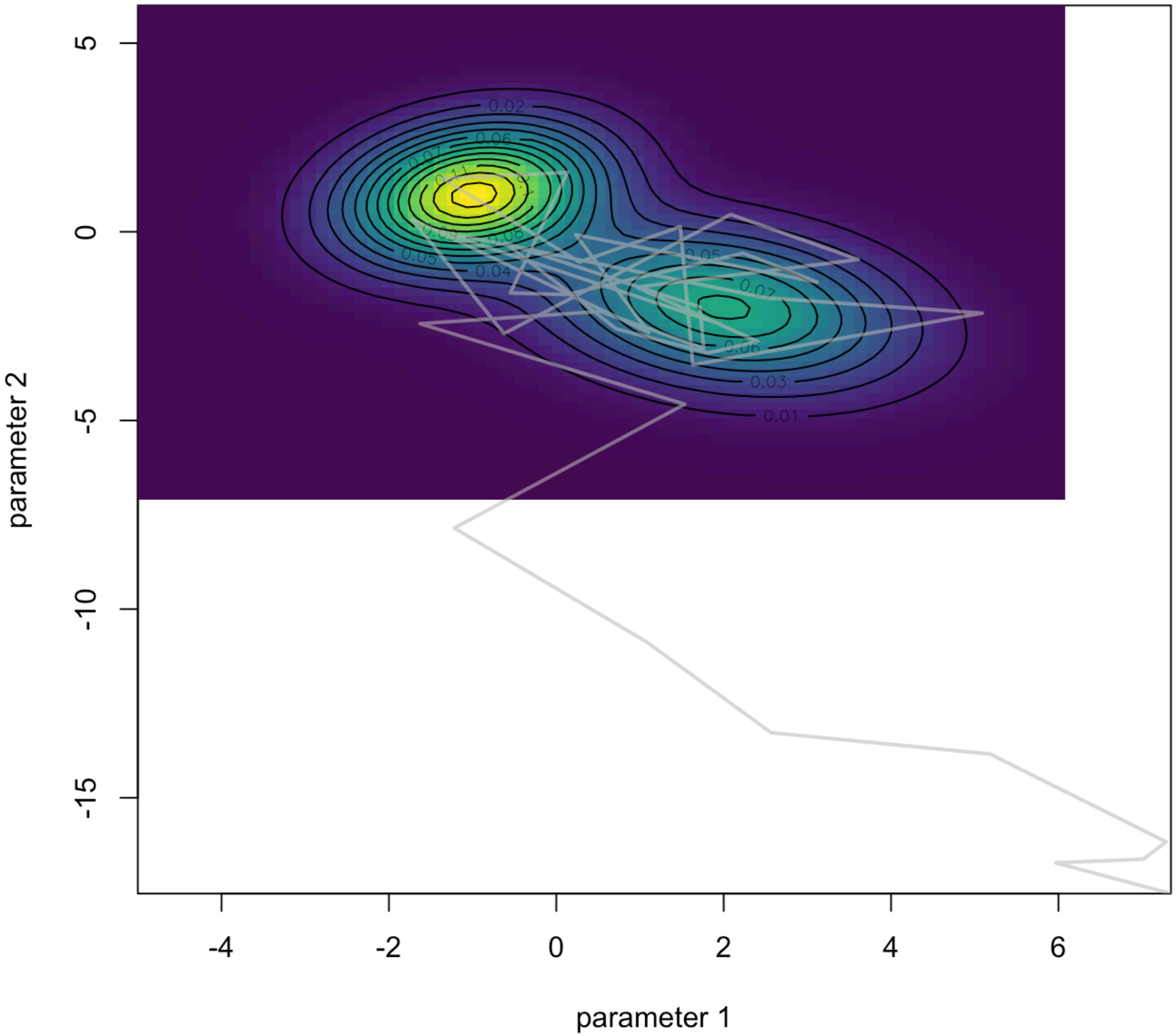
Example

- Suppose two parameters, with likelihood x prior:

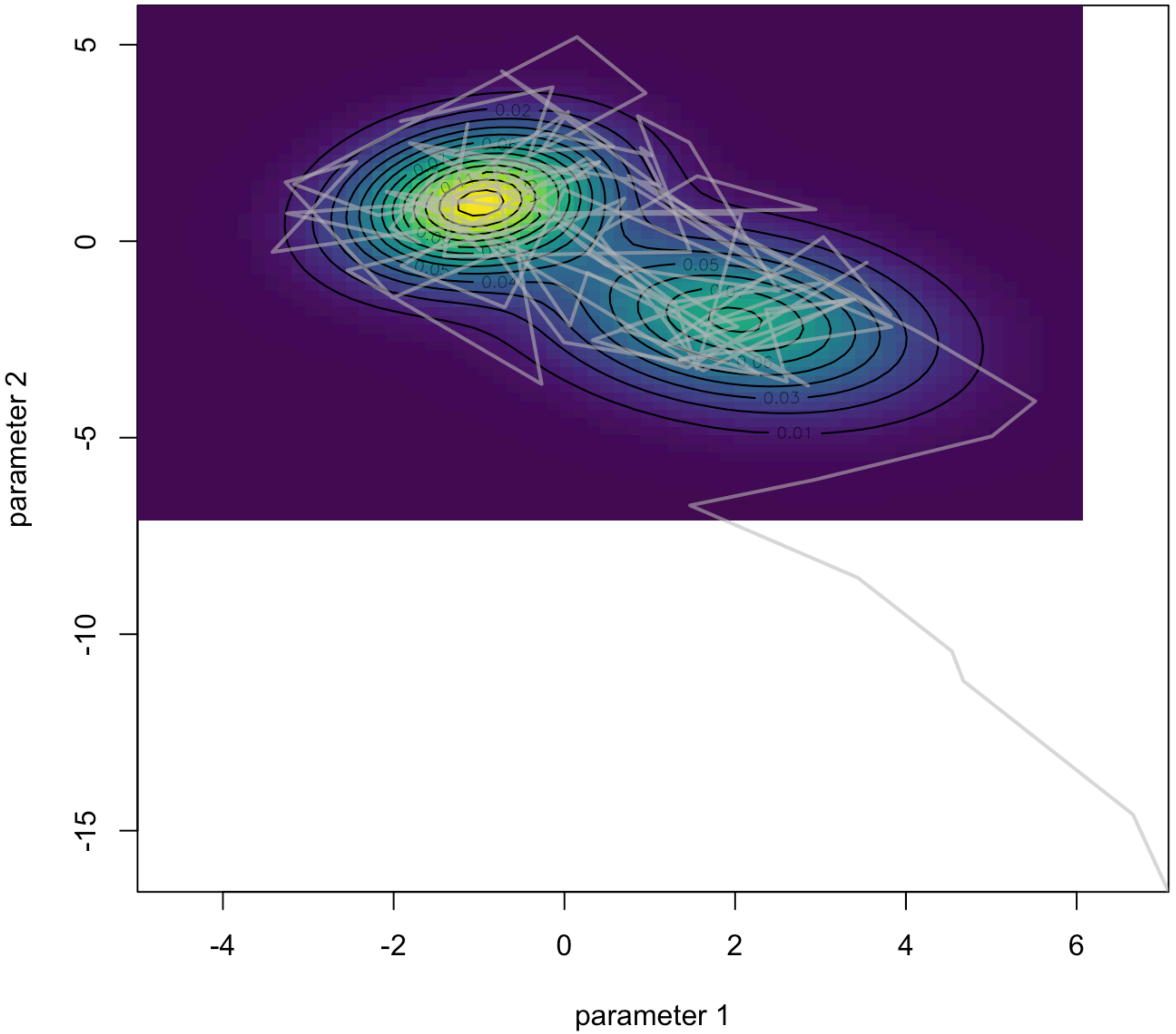


Sample path

We start our parameter values at a random guess
The random walk the MCMC traverses is shown as the grey line

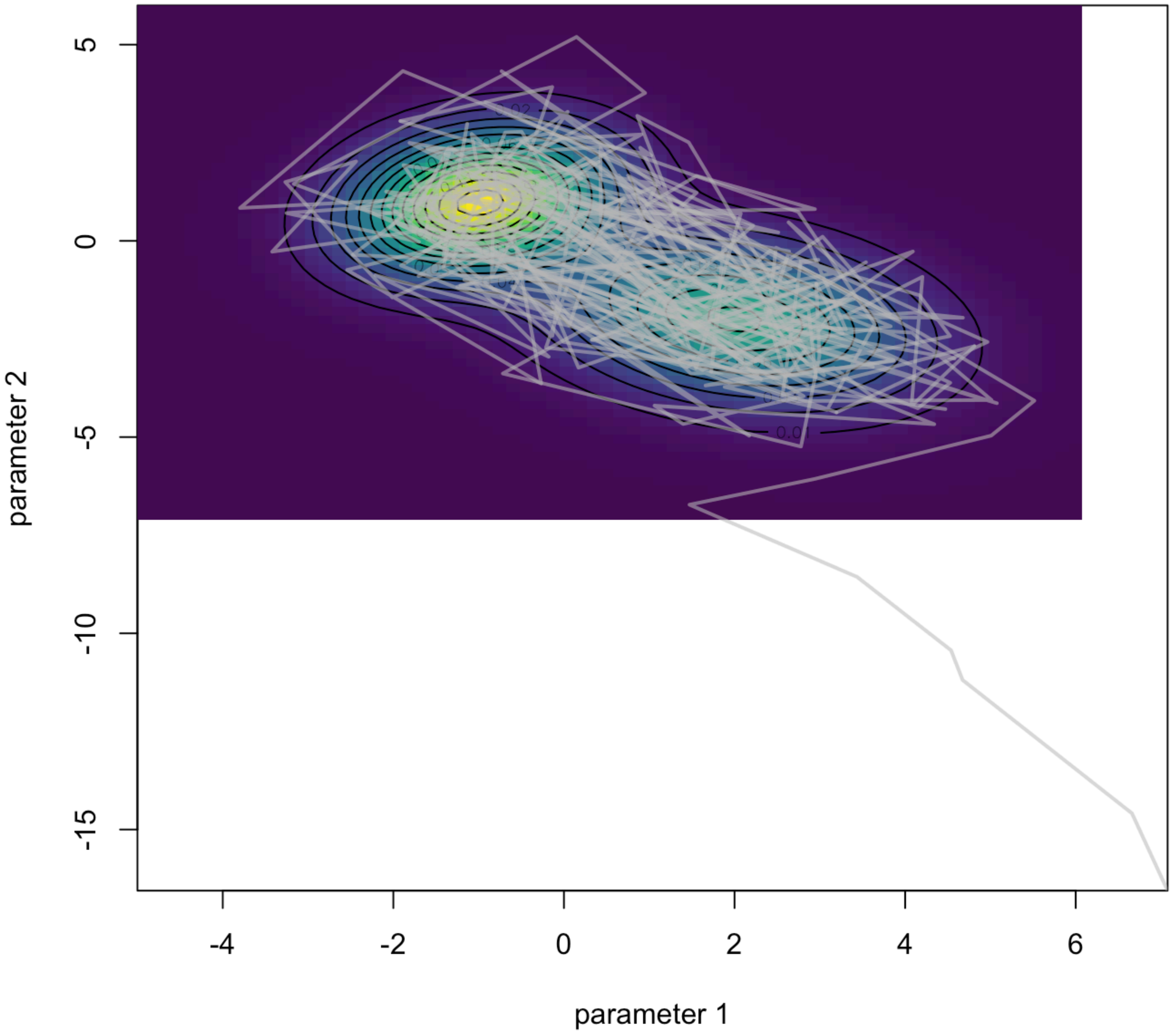


Sample path


parameter 2
5
0
-5
-10
-15
parameter 1
-4
-2
0
2
4
6
0.02
0.05
0.04
0.05
0.03
0.01

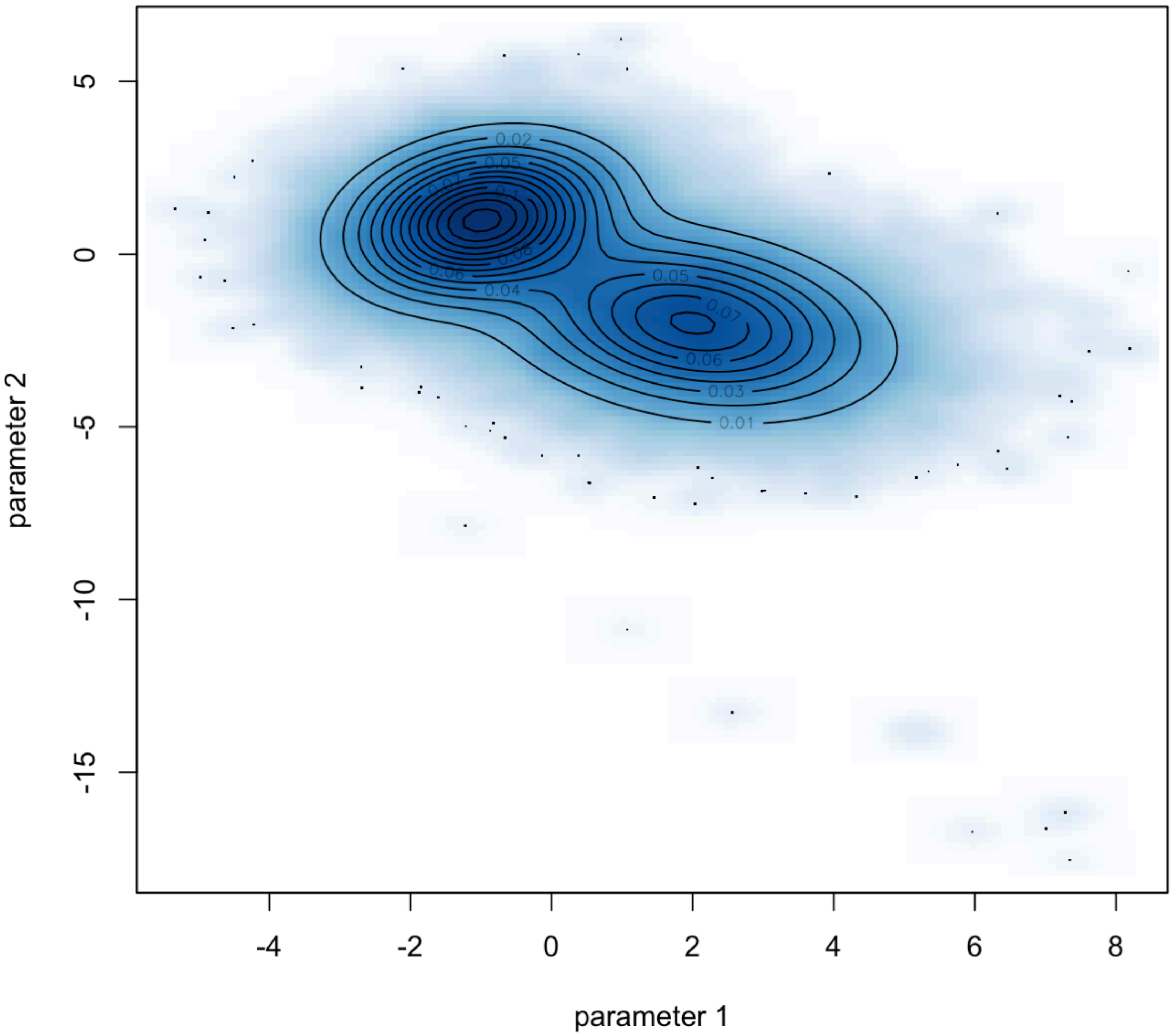
Sample path

Over time, the random walk accrues samples of the posterior distribution, proportional to the probability of those values. In other words, we get more samples from higher probability regions

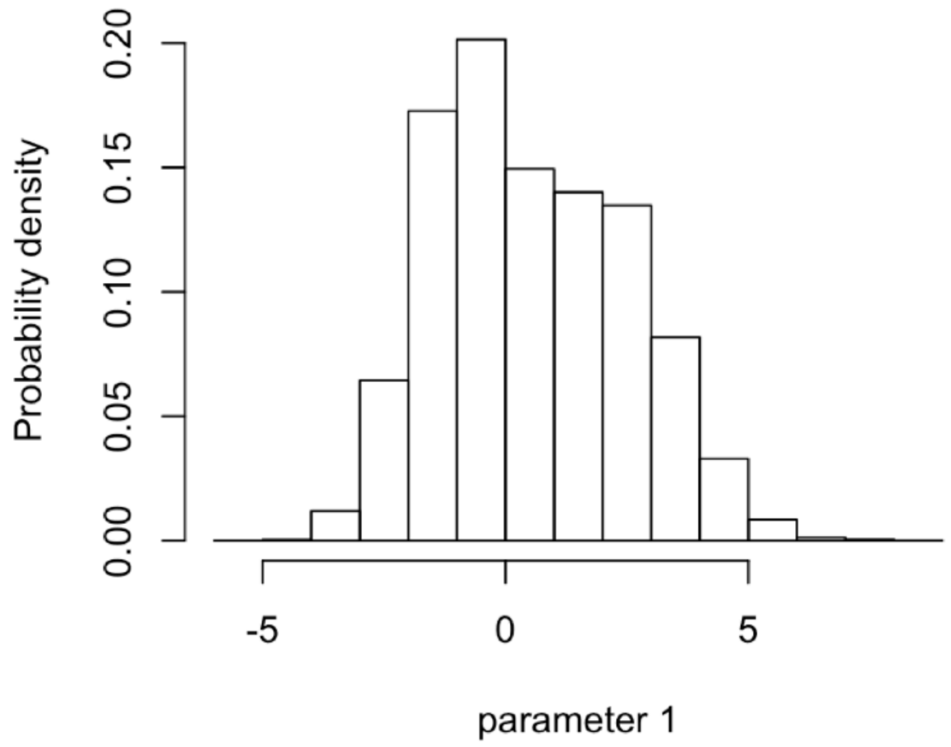


Sampled density

Eventually, the sampled values recreate the posterior distribution! And we didn't need the denominator term, only the numerator term, so these are relatively easy to calculate



Can also get marginals:



Example: Metropolis Algorithm

- Idea is to ‘walk’ randomly through parameter space, spending more time in places that are higher probability—that way, the overall distribution draws more from higher probability spots

- Setup—we need

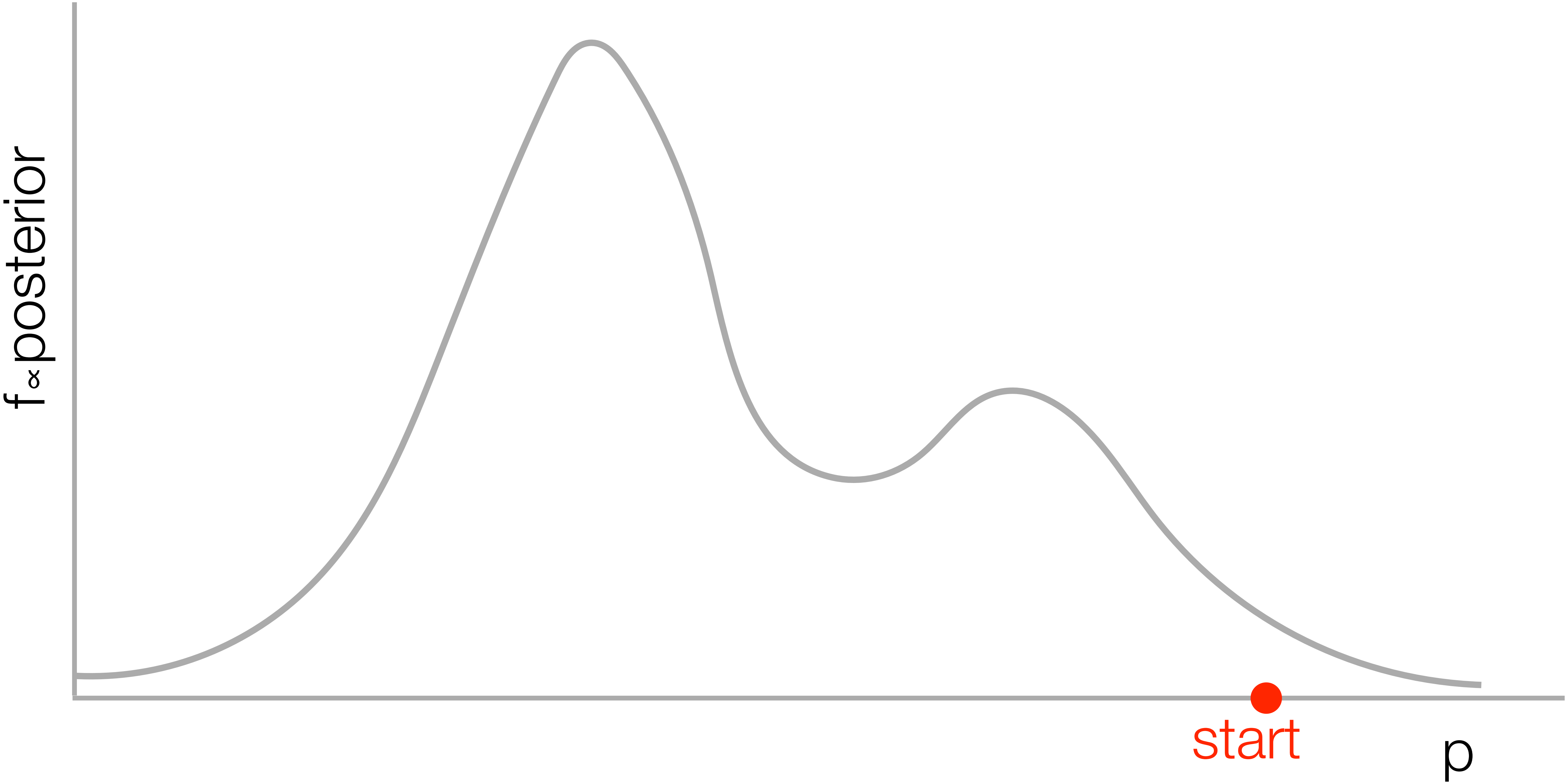
  - A function $f(p)$ proportional to the distribution we want to sample, in our case $f(p) = P(z|p) \cdot P(p)$

  - A proposal distribution (how we choose the next point from the current one) - more on this in a minute

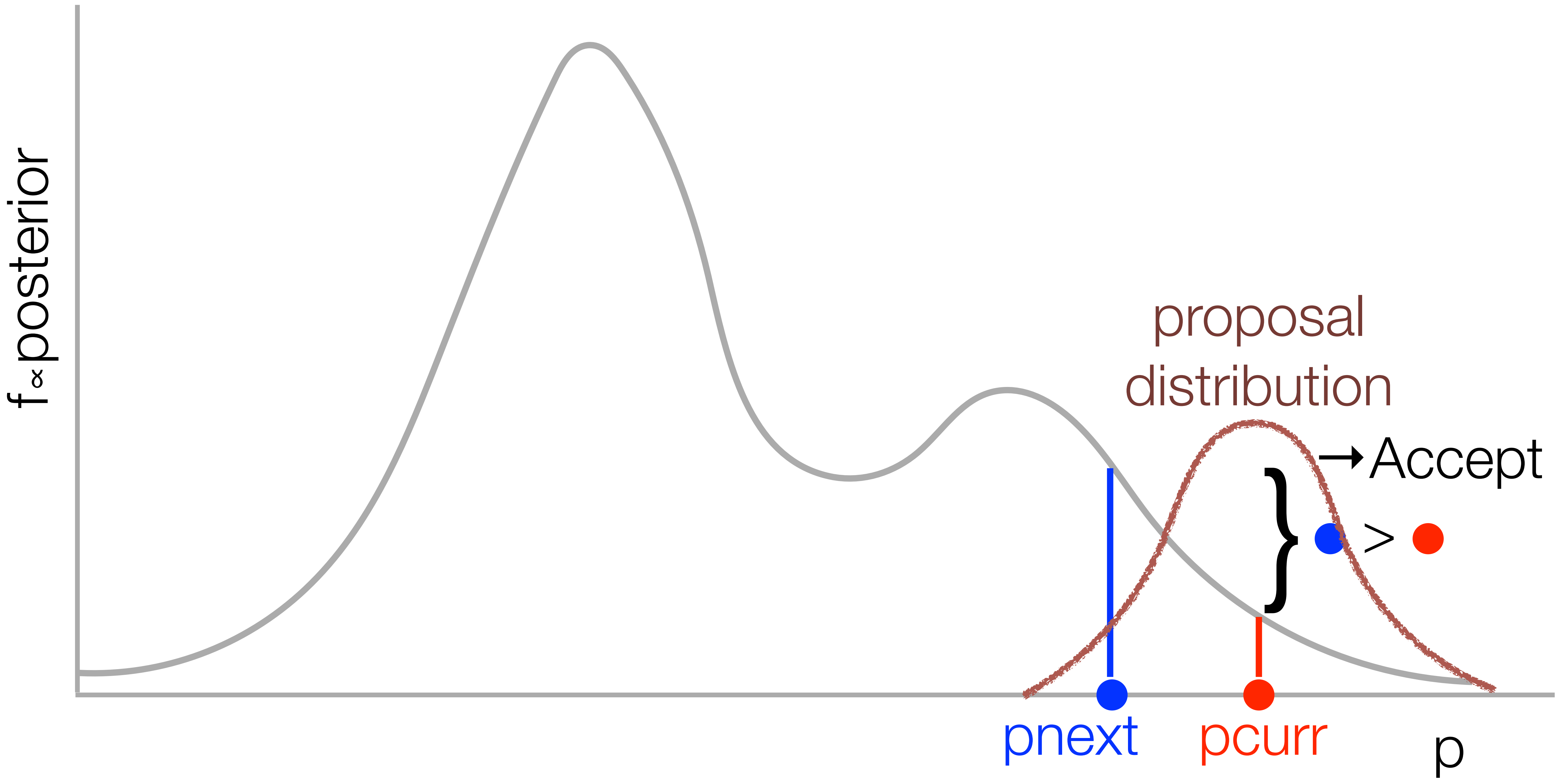
Metropolis Algorithm

- Start at some point in parameter space
- For each iteration
  - Propose a new random point p_{next} based on the current point p_{curr} (using the proposal distribution)
  - Calculate the **acceptance ratio**, $\alpha = f(p_{next})/f(p_{curr})$
    - If $\alpha \geq 1$, the new point is as good or better—accept
    - If $\alpha < 1$, accept with probability α

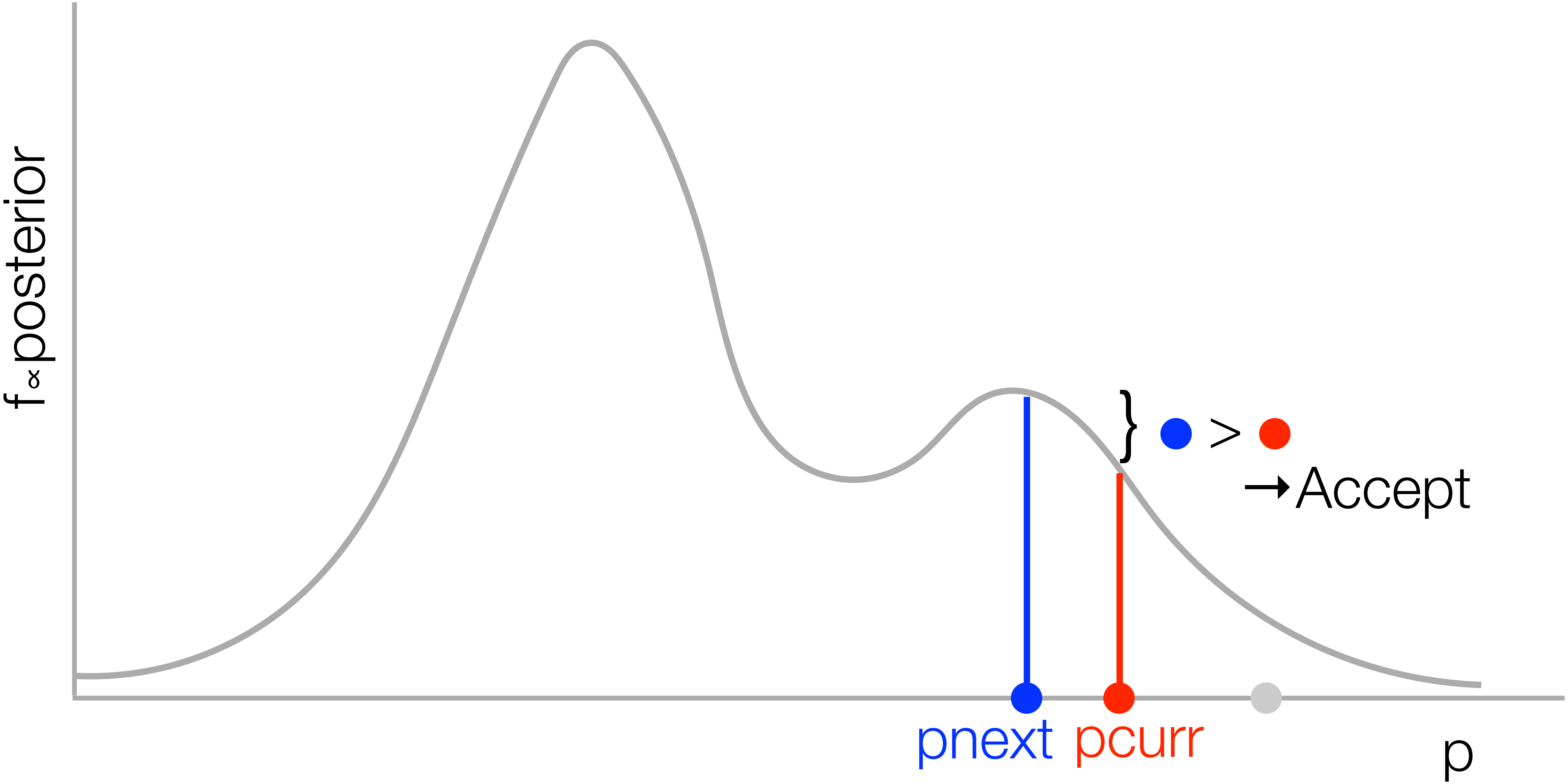
What does the metropolis algorithm do?



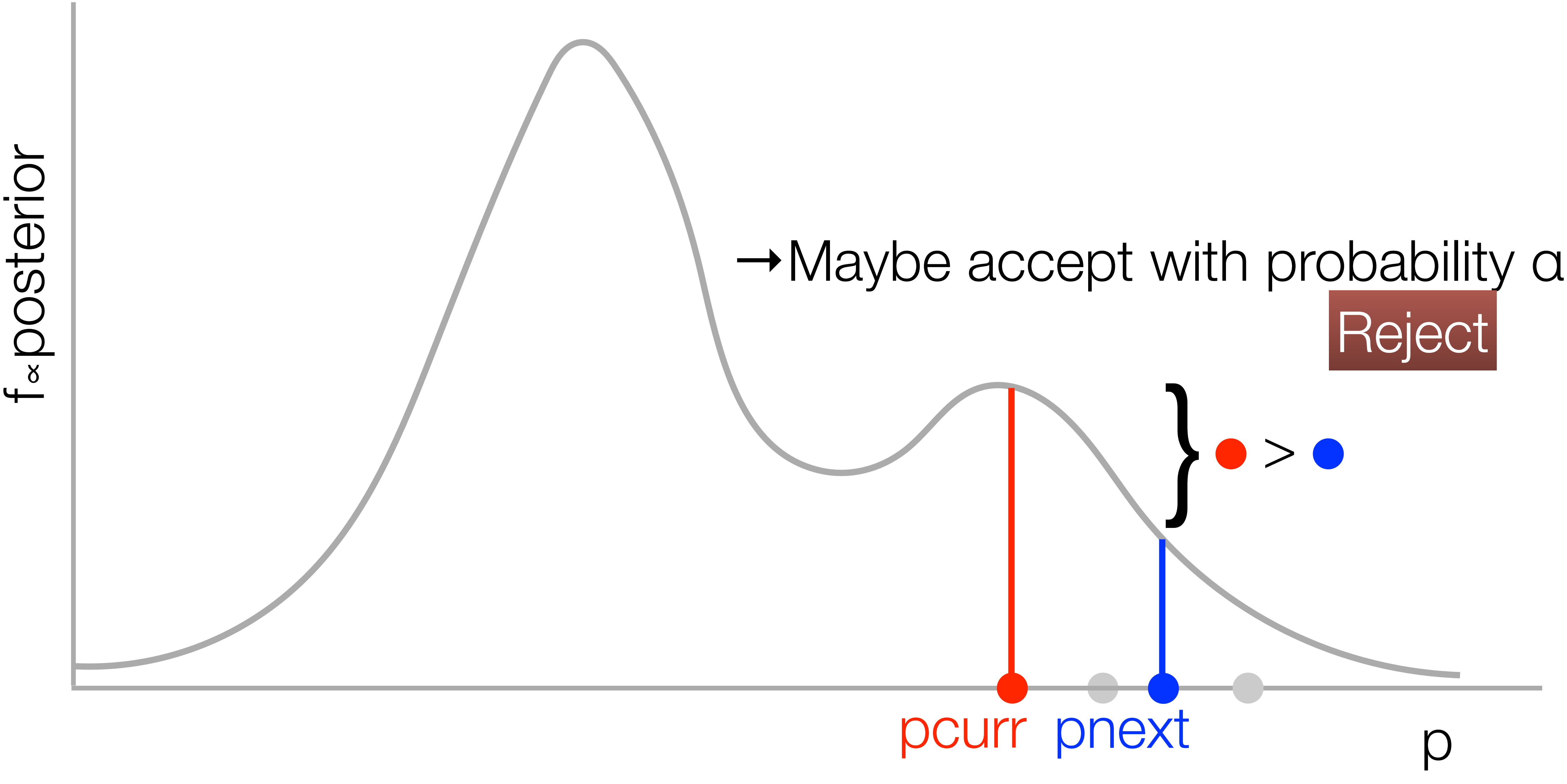
What does the metropolis algorithm do?



What does the metropolis algorithm do?



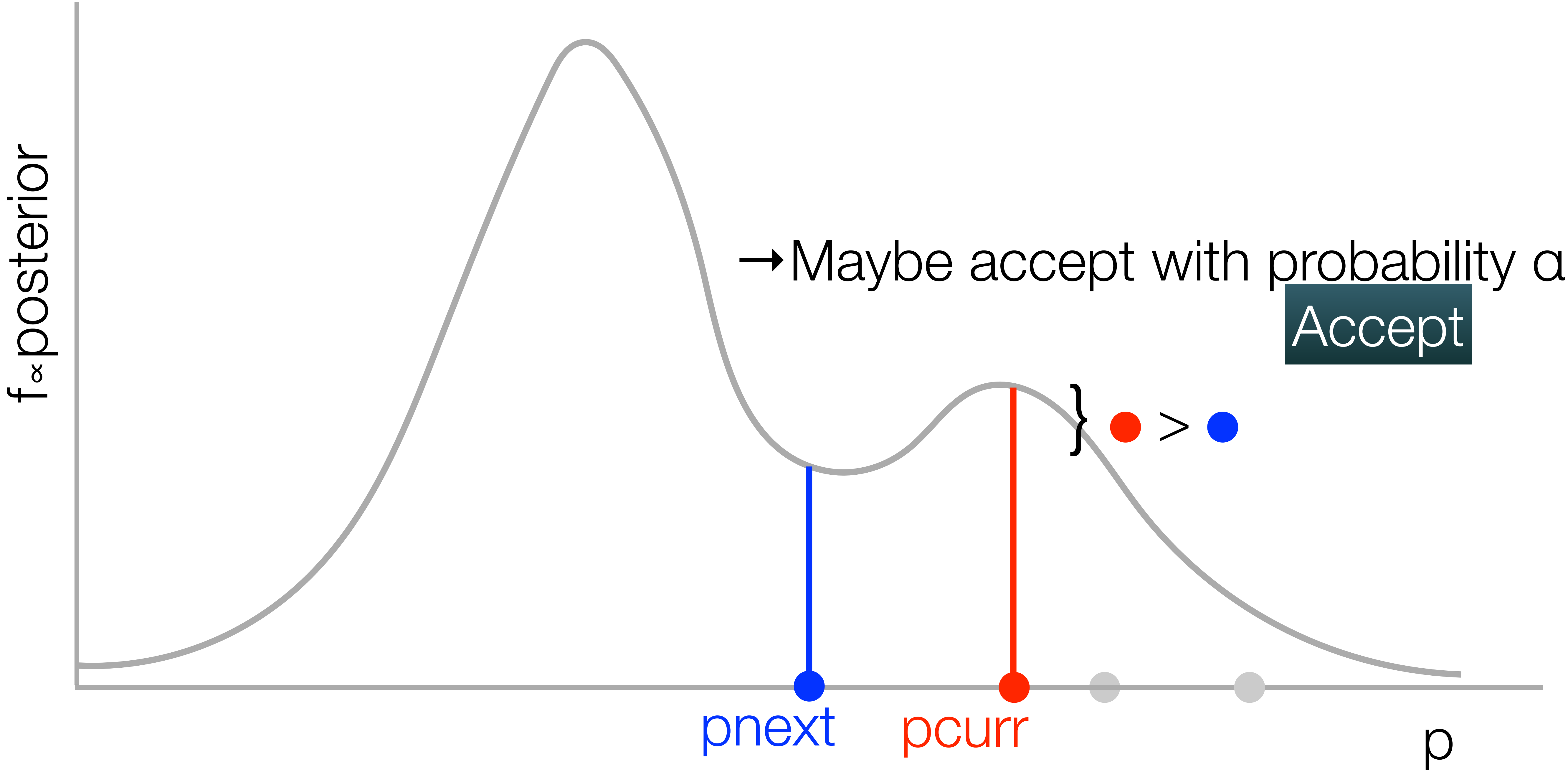
What does the metropolis algorithm do?



→Maybe accept with probability α

Reject

What does the metropolis algorithm do?

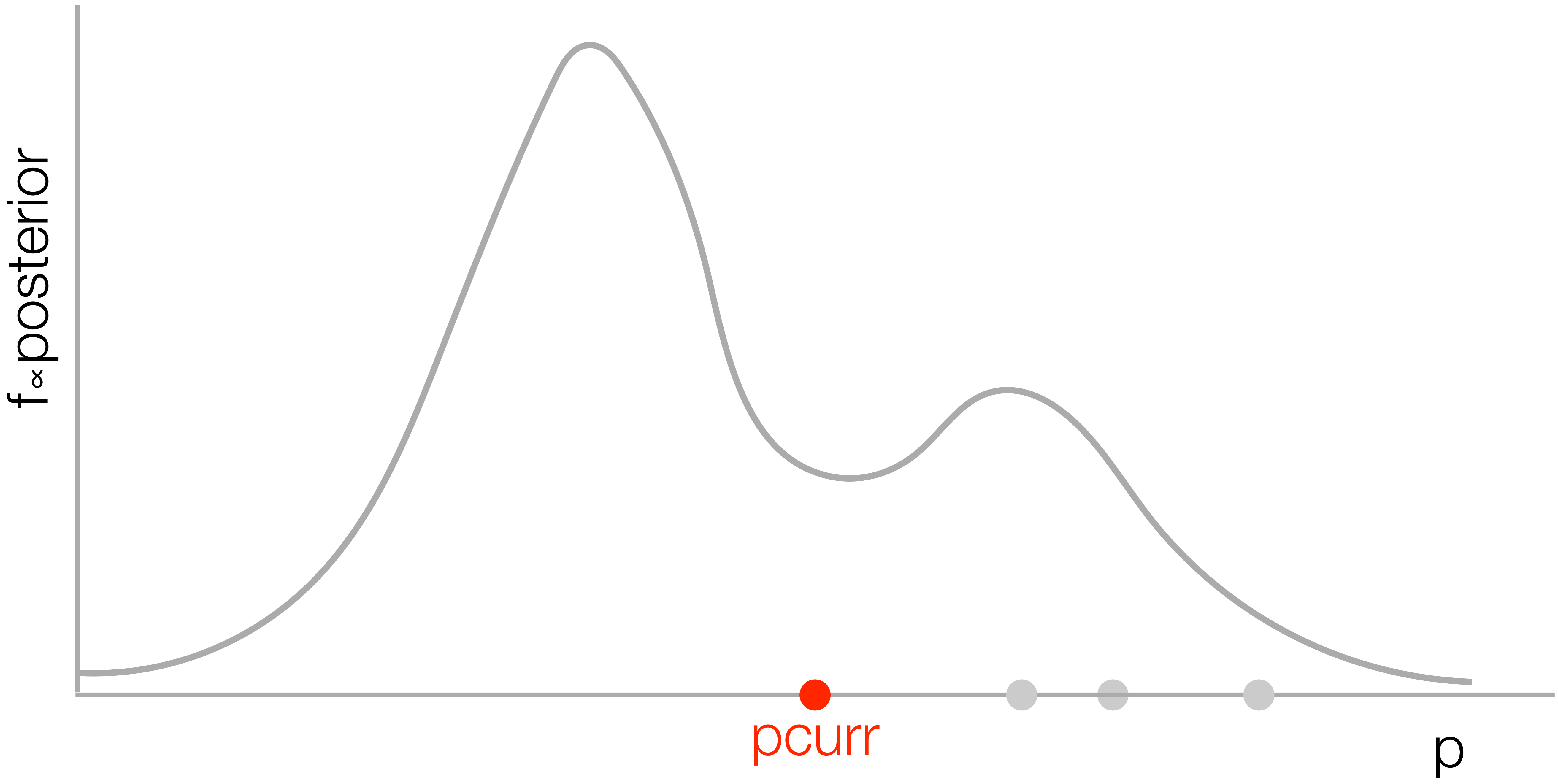


→Maybe accept with probability α

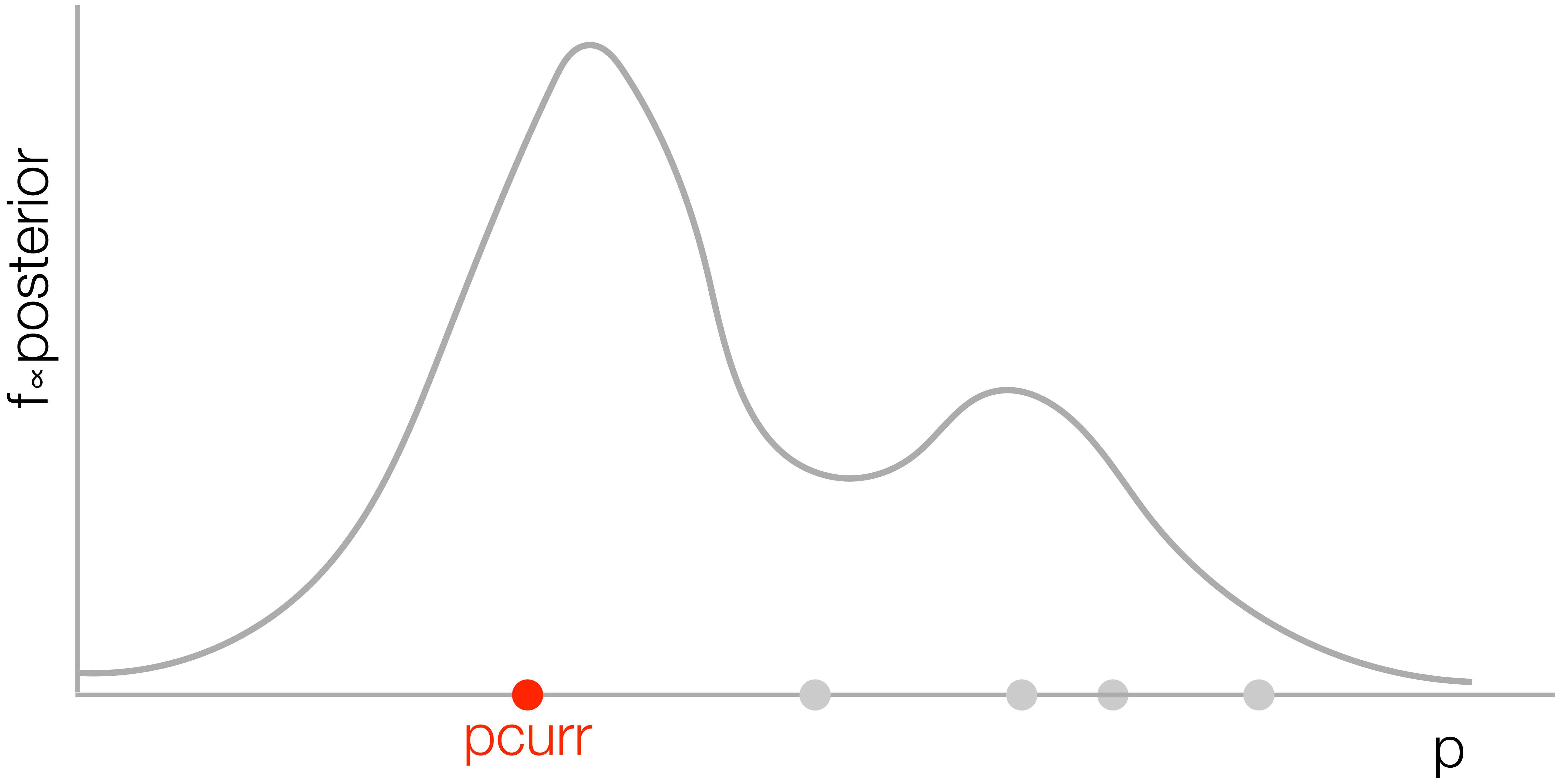
Accept

} ● > ●

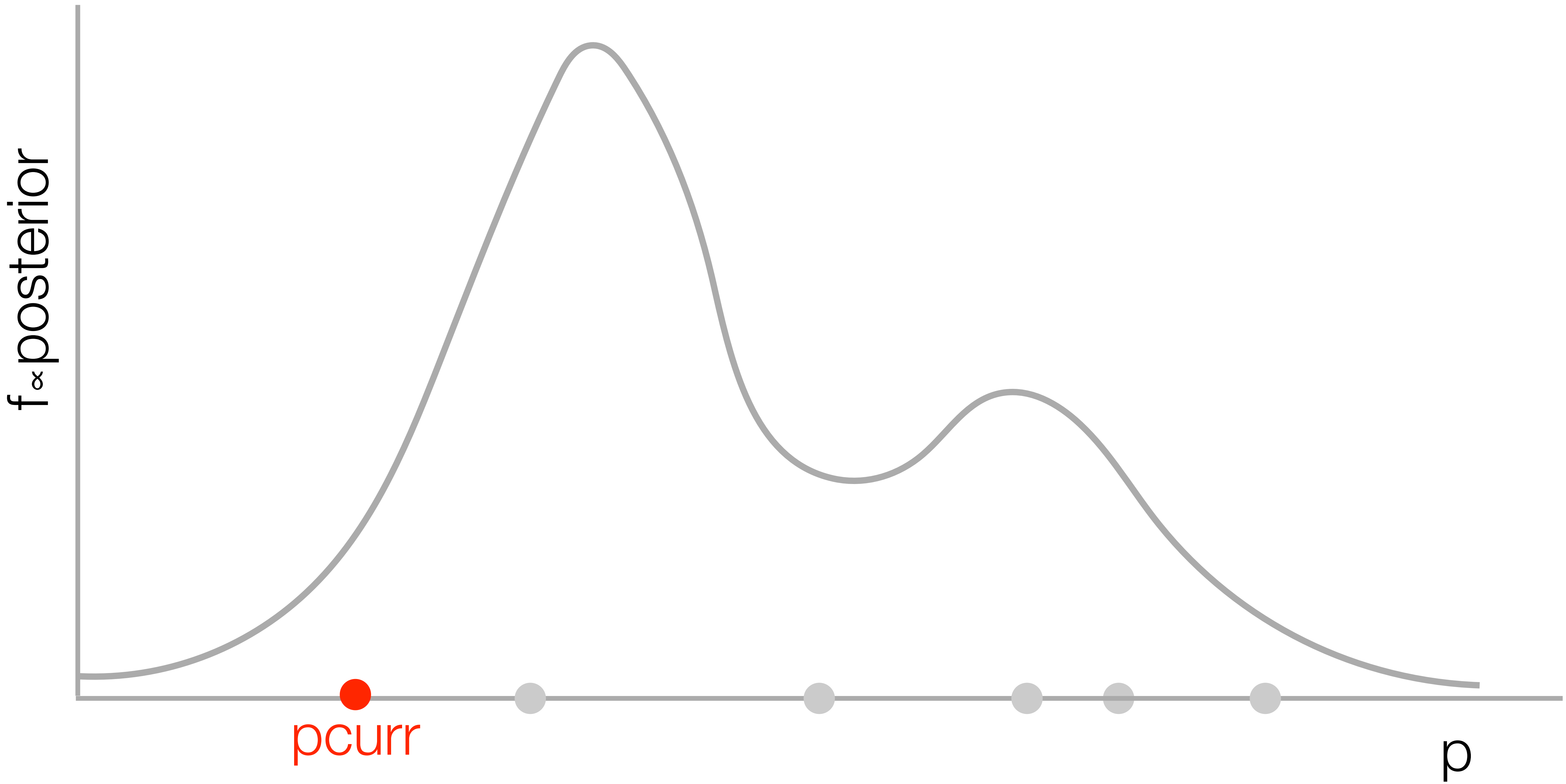
What does the metropolis algorithm do?



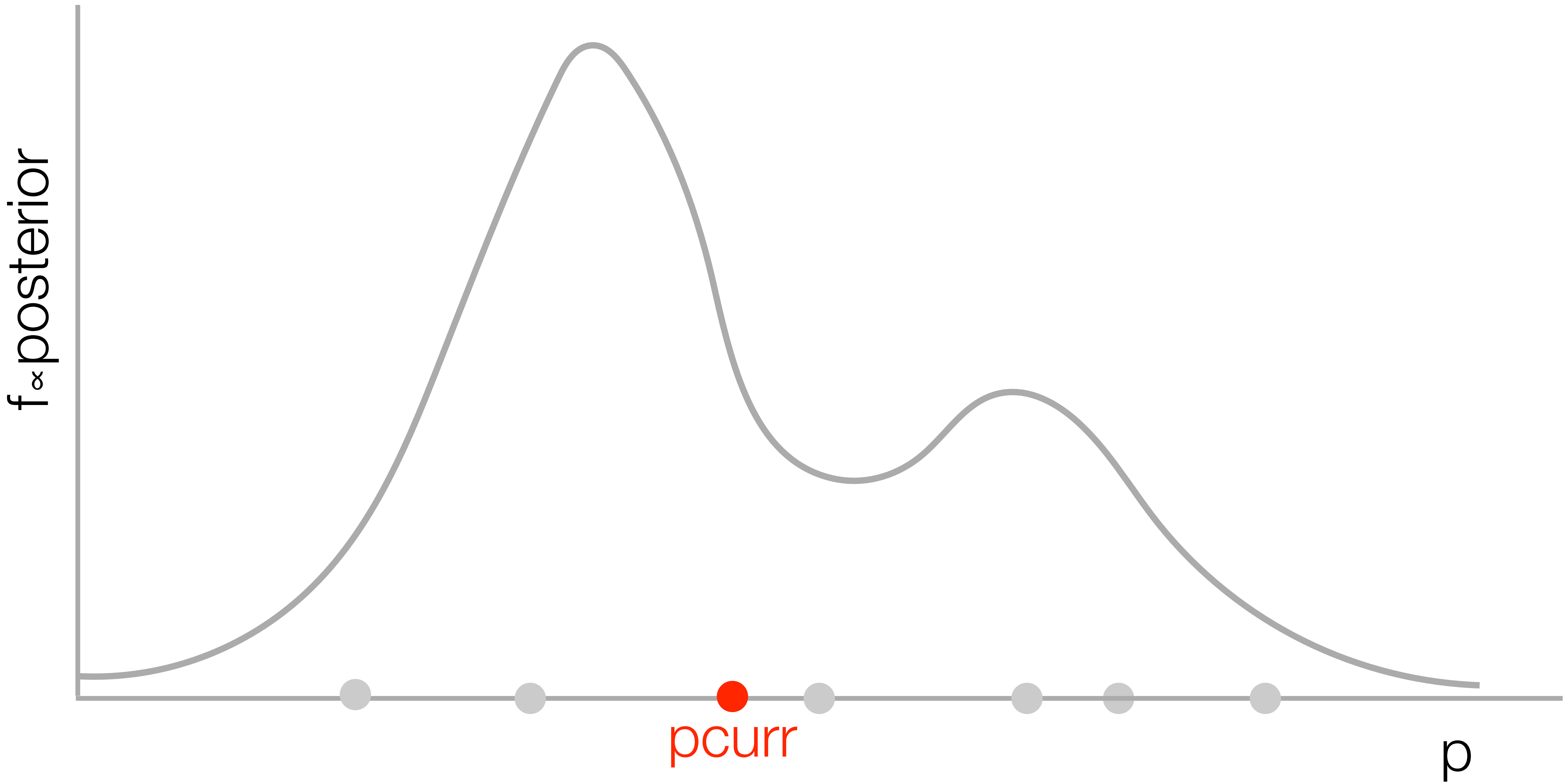
What does the metropolis algorithm do?



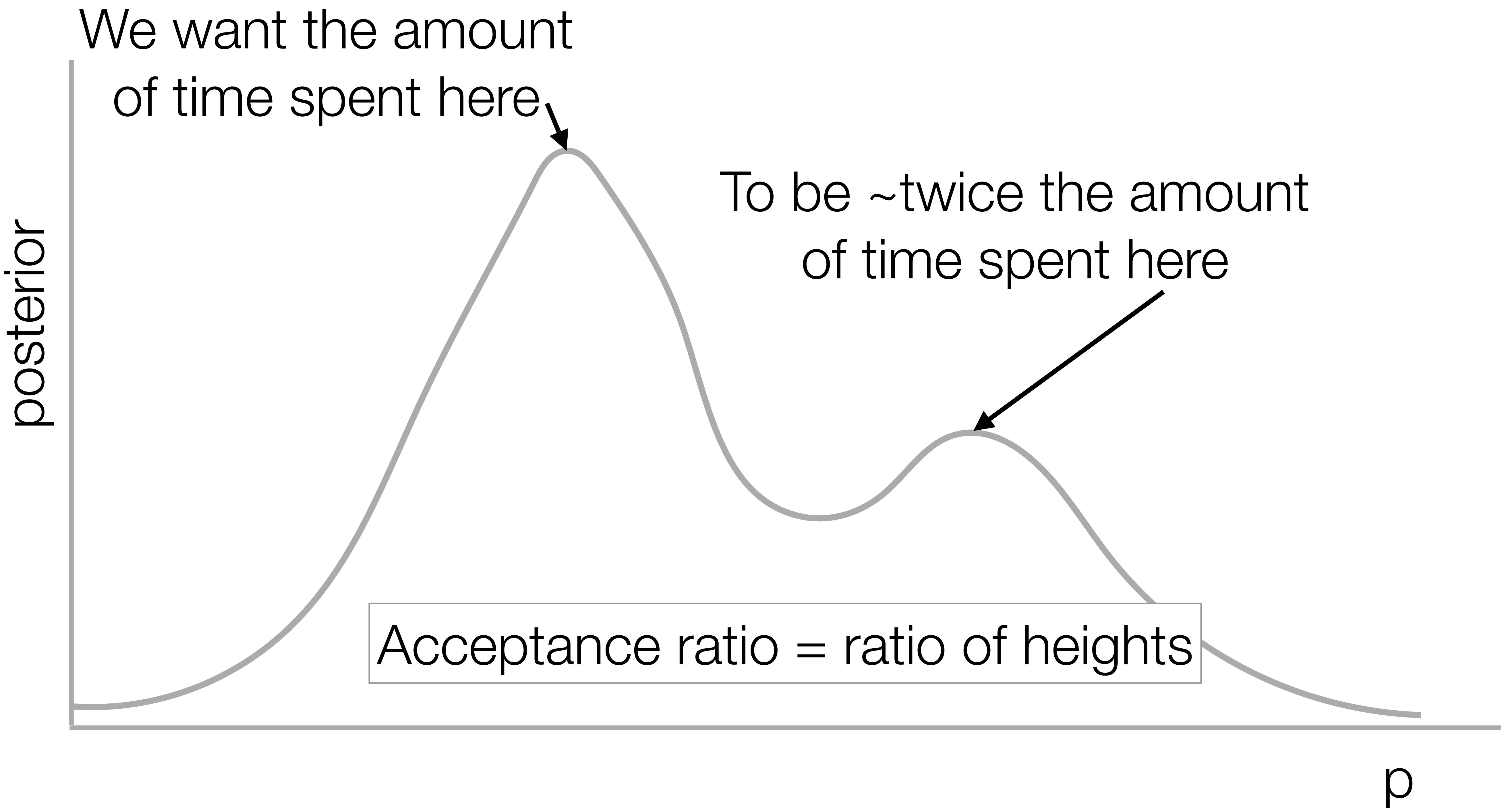
What does the metropolis algorithm do?



What does the metropolis algorithm do?



Why does this recover the posterior distribution? Key is the acceptance ratio α



We want the amount of time spent here

To be ~twice the amount of time spent here

Acceptance ratio = ratio of heights

Why does this recover the posterior distribution?

- The acceptance ratio $\alpha = f(p_{next})/f(p_{curr})$

- Note it is equal to $P(p_{next}|z)/P(p_{curr}|z)$ since the denominators cancel

- Suppose we're at the peak

  - If f(p_{curr}) = 2 f(p_{next}), then $\alpha = 1/2$, i.e. we accept with 1/2 probability

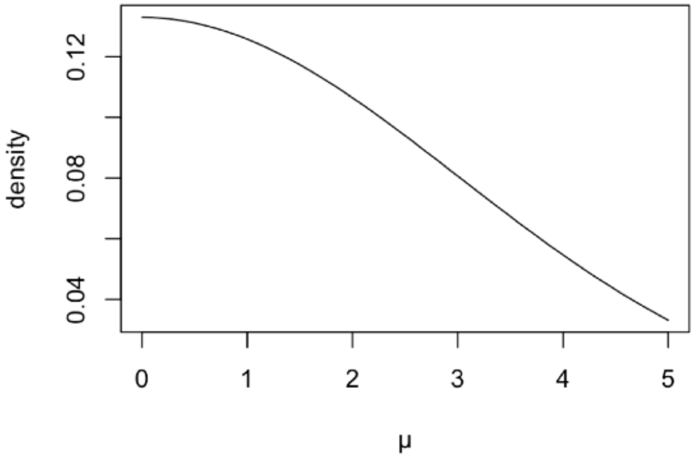
- Overall, will mean the number of samples we take from a region will be proportional to the height of the distribution

Proposal Distribution

- A distribution that lets us choose our next point randomly from our current one
- For Metropolis algorithm, must be symmetric
- Common to choose a normal distribution centered on current point
- Width (SD) of normal = proposal width
  - Choice of proposal width can strongly affect how the Markov chain behaves, how well it converges, mixes, etc.

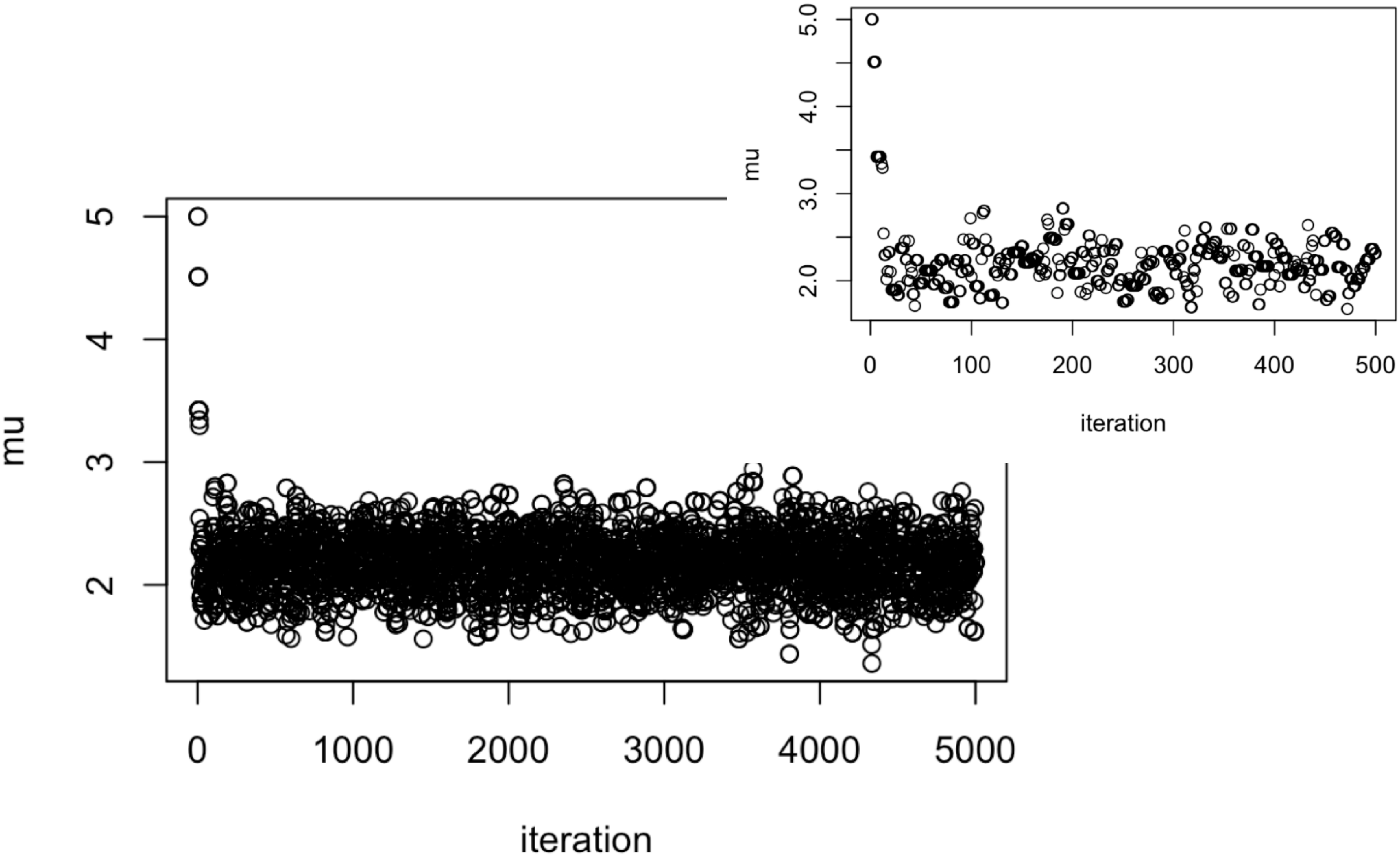
Example (apologies in advance for picking normal distributions for everything!)

- Model: normal distribution $\mathcal{N}(\mu, \sigma)$
  - Suppose σ is known, μ to be estimated
  - Likelihood: $P(z_i \mid \mu, 1) = f(z_i \mid \mu, 1) = \frac{1}{\sqrt{2\pi}} e^{-\frac{(z-\mu)^2}{2}}$ $\qquad P(z \mid \mu) = \prod_{i=1}^{n} f(z_i \mid \mu, 1)$
- Prior: $\mu \sim \mathcal{N}(0, 3)$



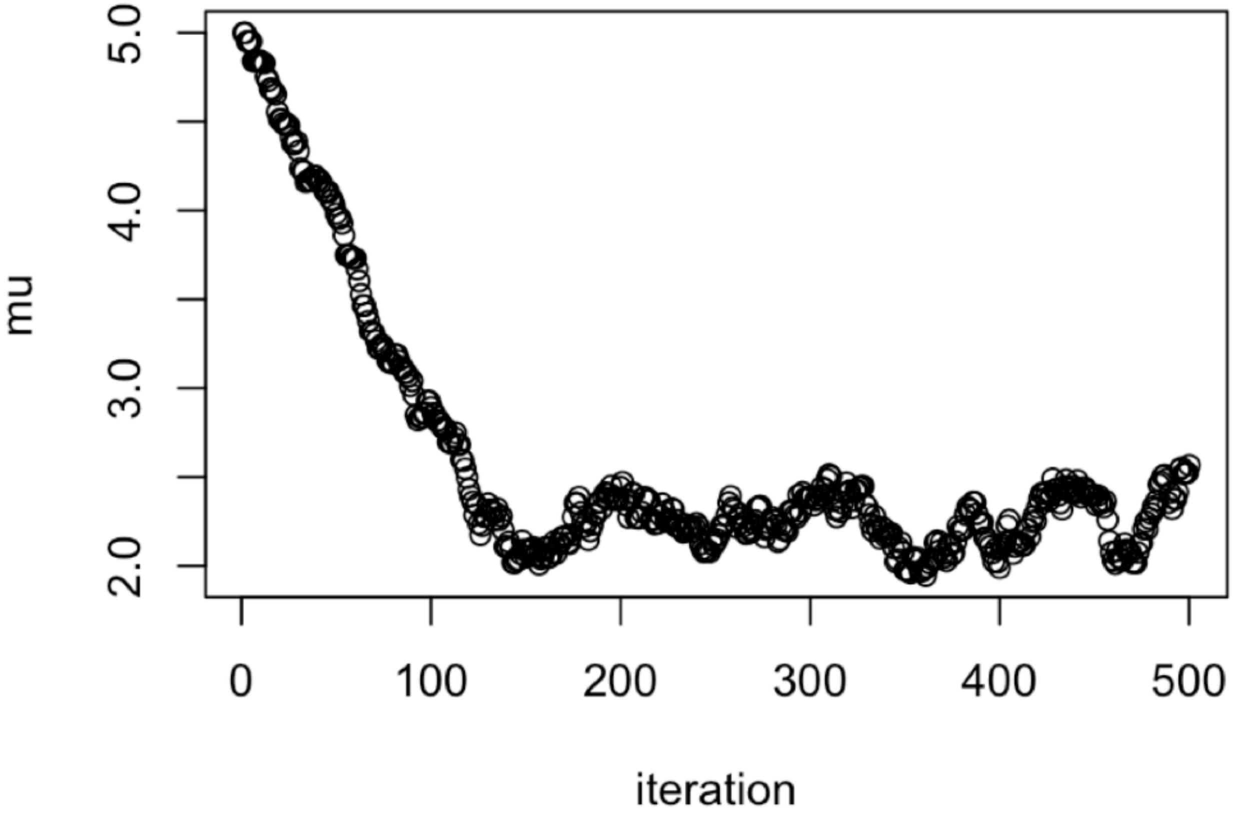
- Suppose we have 20 data points

Example - proposal width: SD = 0.5

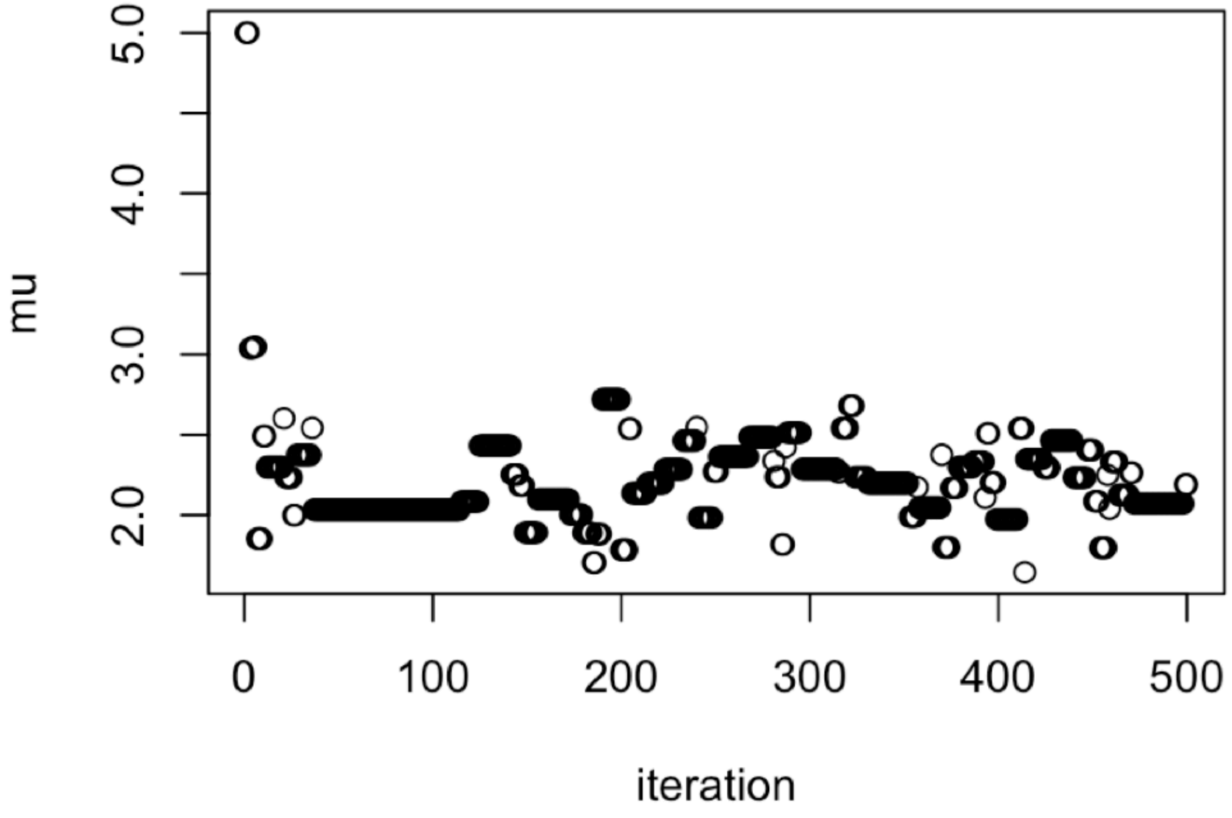


Goldilocks problem:
What happens if we change the proposal width?

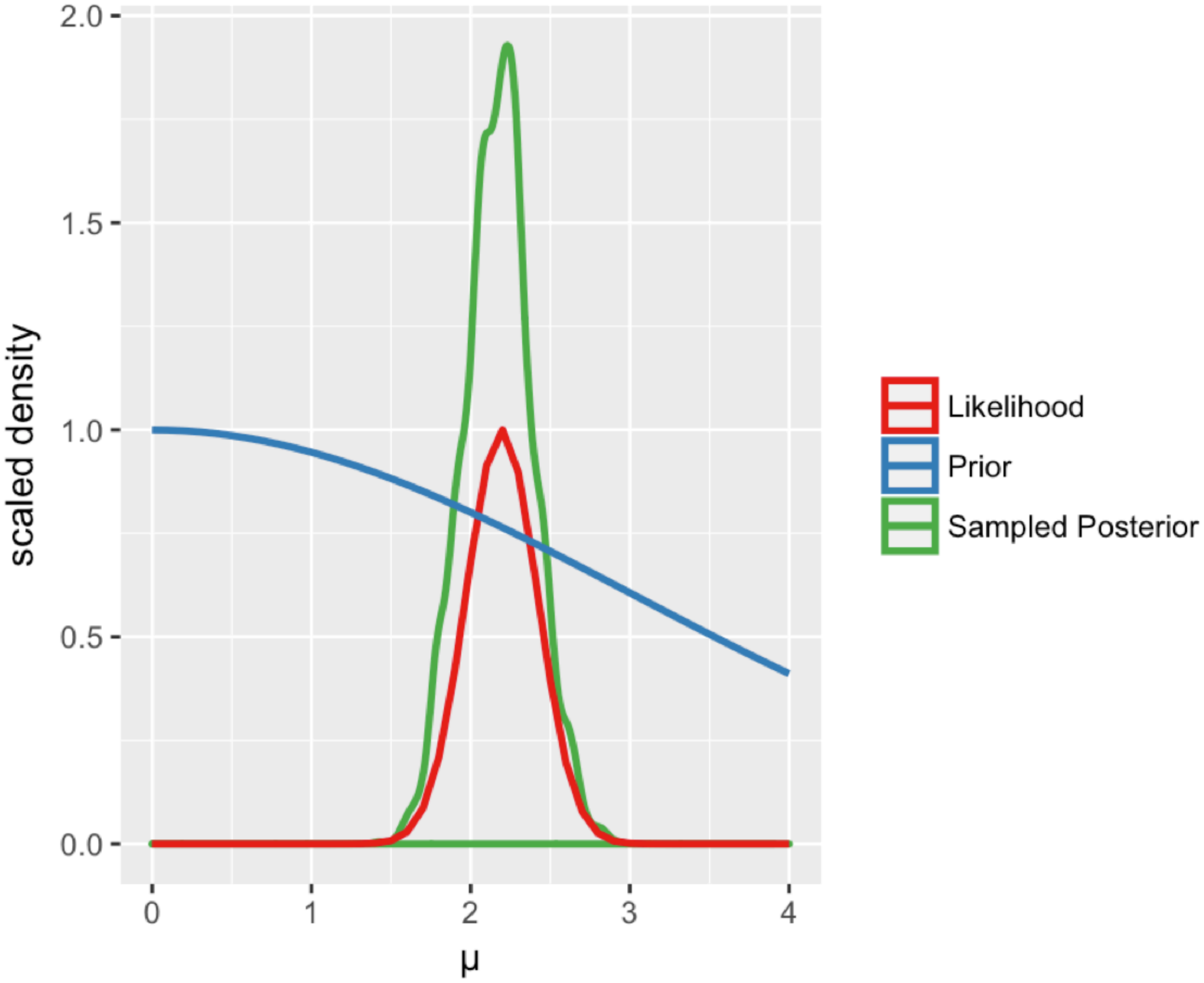
proposal SD = 0.05



proposal SD = 2



Example: prior, likelihood, and posterior (all scaled)



MCMC

- MCMC improves many of the problems that other optimization methods face (getting trapped in local minima, etc.)

- However, those issues can still cause problems for MCMC too

- How to know when you’ve run the MCMC long enough and collected enough samples to reflect the distribution?

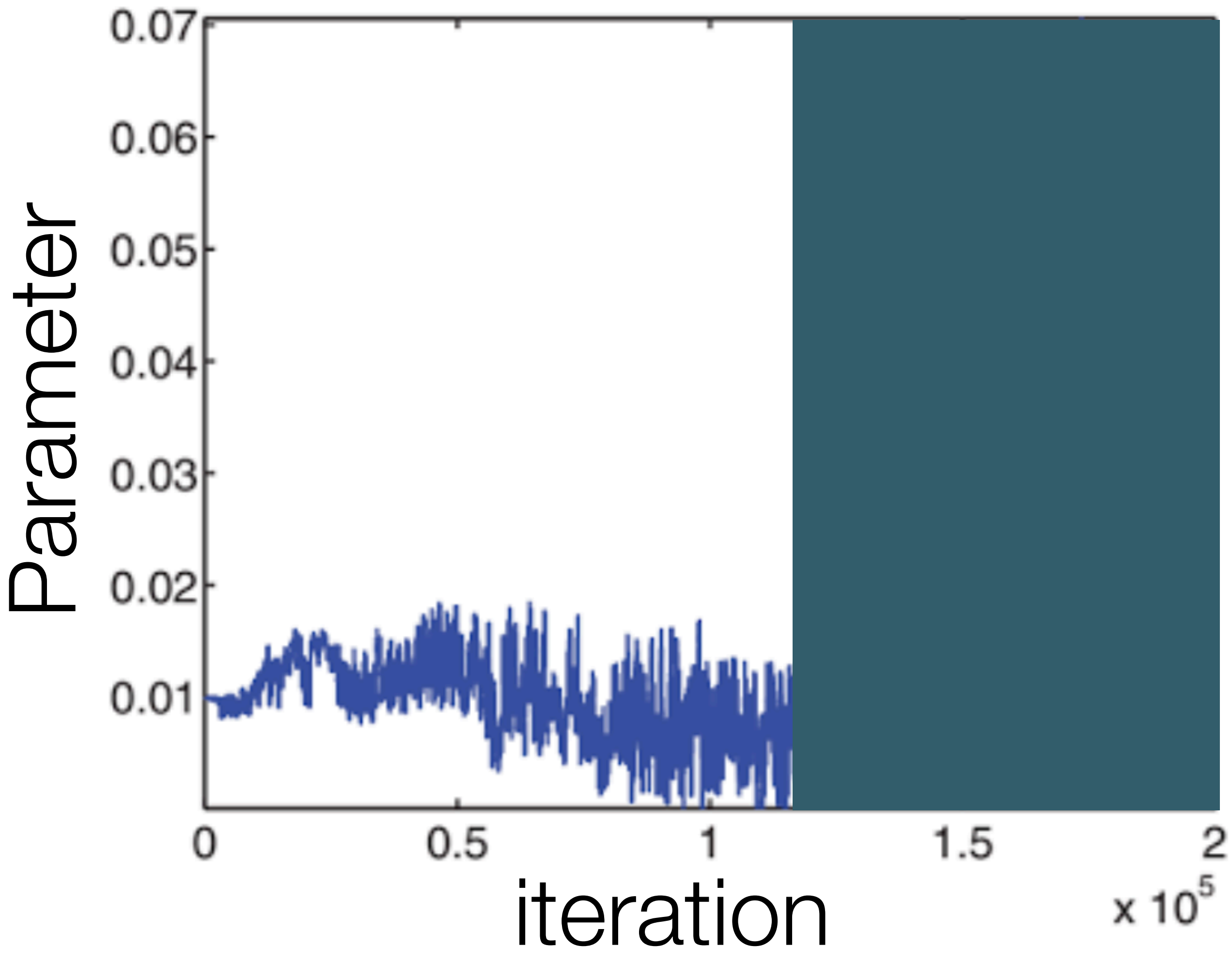
- How to know if you have explored the space sufficiently?

Assessing convergence

- MCMC methods will let us sample the posterior once they've converged to their equilibrium distribution
- How to know once we've reached equilibrium?
  - Visual evaluation of **burn-in**
  - Autocorrelation of elements in chain *k* iterations apart
- Also approaches to use in combination with/instead of burn-in: start with MAP estimation, multiple chains, etc.

Assessing convergence

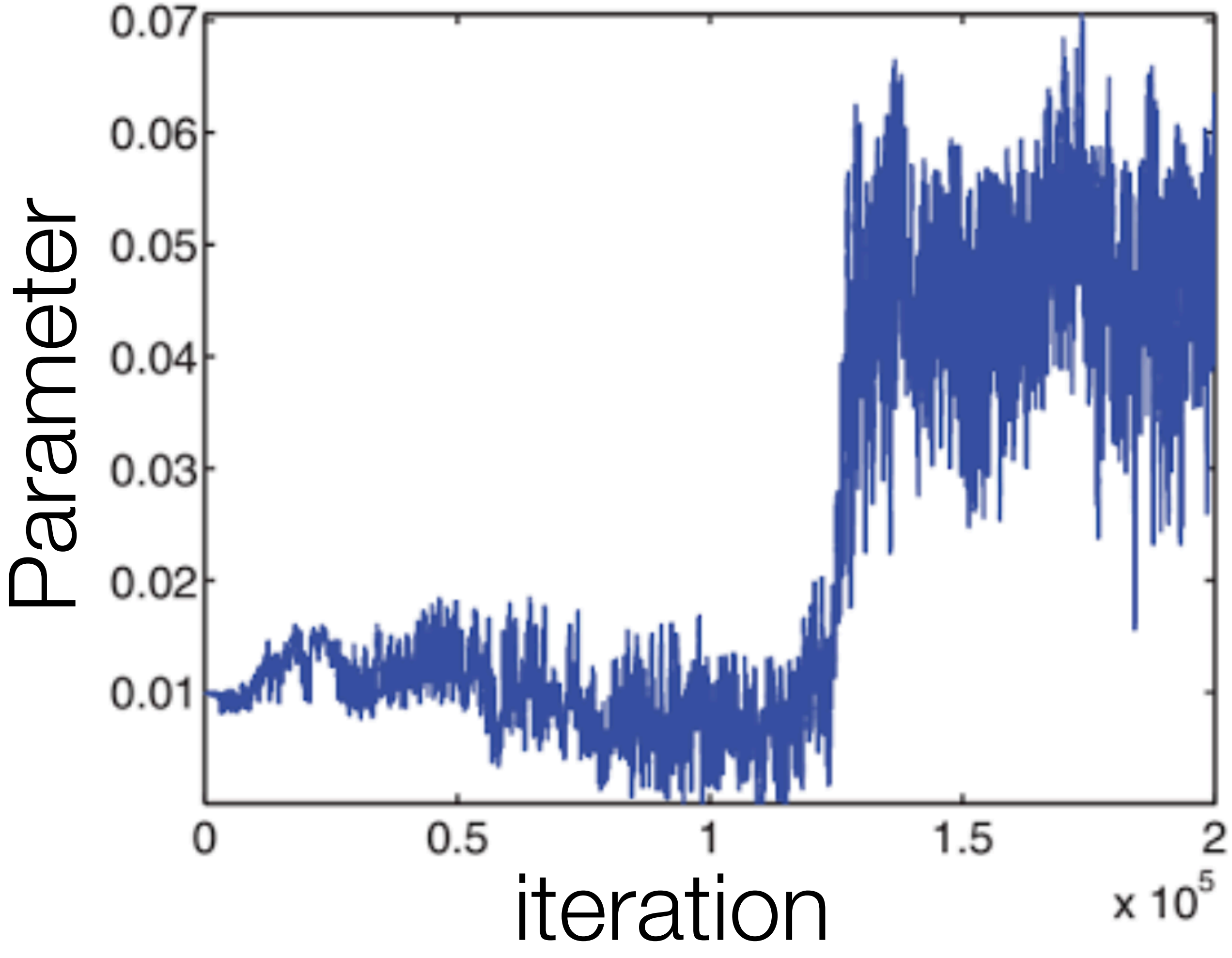
- Often done visually
- Although, this can be misleading:



Chain shifts after 130,000 iterations due to a local min in sum of squares (Example from R. Smith, *Uncertainty Quantification*)

Assessing convergence

- Often done visually
- Although, this can be misleading:



Chain shifts after 130,000 iterations due to a local min in sum of squares (Example from R. Smith, *Uncertainty Quantification*)

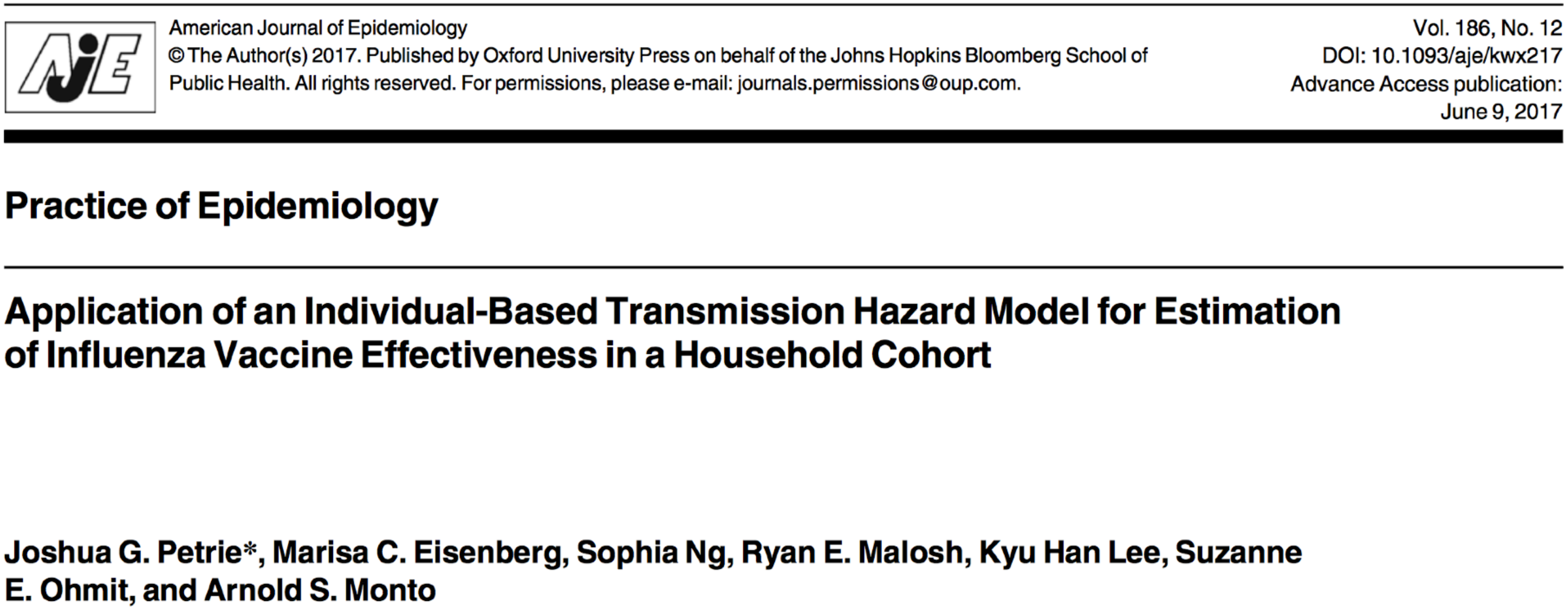
Metropolis & Metropolis-Hastings Caveats

- Assessing convergence—how long is burn-in?
  - What about when you have unidentifiability or multiple minima?
- Correlated samples
- How to choose a proposal width? (~size of next jump)

Wide range of methods

- Metropolis–Hastings
- Gibbs sampling
- Variations of the above: prior optimization, multi-start, adaptive methods, delayed rejection
    - DRAM (Delayed Rejection Adaptive Metropolis-Hastings)
- Many more!

Examples



American Journal of Epidemiology
© The Author(s) 2017. Published by Oxford University Press on behalf of the Johns Hopkins Bloomberg School of Public Health. All rights reserved. For permissions, please e-mail: journals.permissions@oup.com.

Vol. 186, No. 12
DOI: 10.1093/aje/kwx217
Advance Access publication:
June 9, 2017

Practice of Epidemiology

Application of an Individual-Based Transmission Hazard Model for Estimation of Influenza Vaccine Effectiveness in a Household Cohort

Joshua G. Petrie*, Marisa C. Eisenberg, Sophia Ng, Ryan E. Malosh, Kyu Han Lee, Suzanne E. Ohmit, and Arnold S. Monto

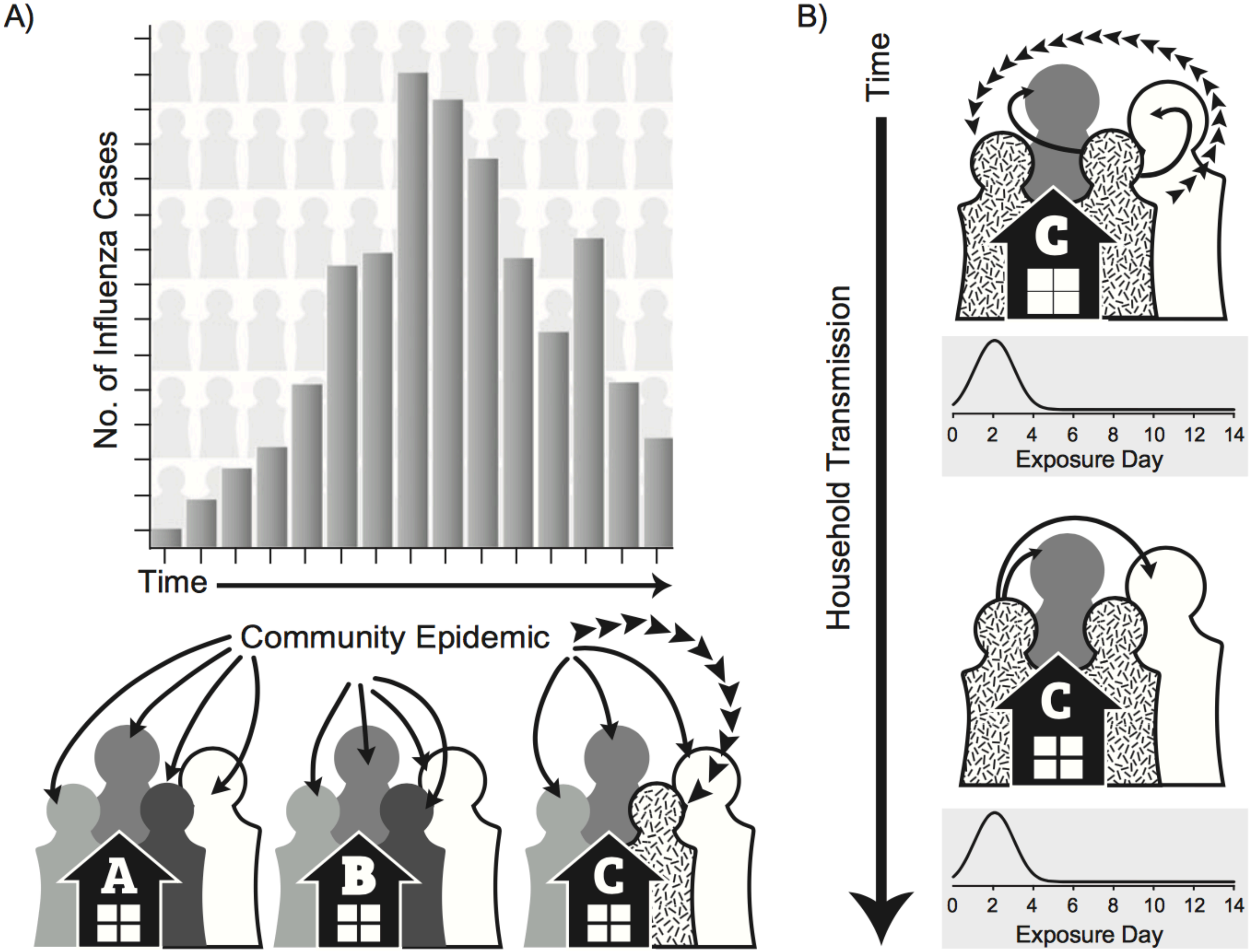

A)
No. of Influenza Cases
Time
Community Epidemic
A
B
C
B)
Time
Household Transmission
C
0
2
4
6
8
10
12
14
Exposure Day
C
0
2
4
6
8
10
12
14
Exposure Day

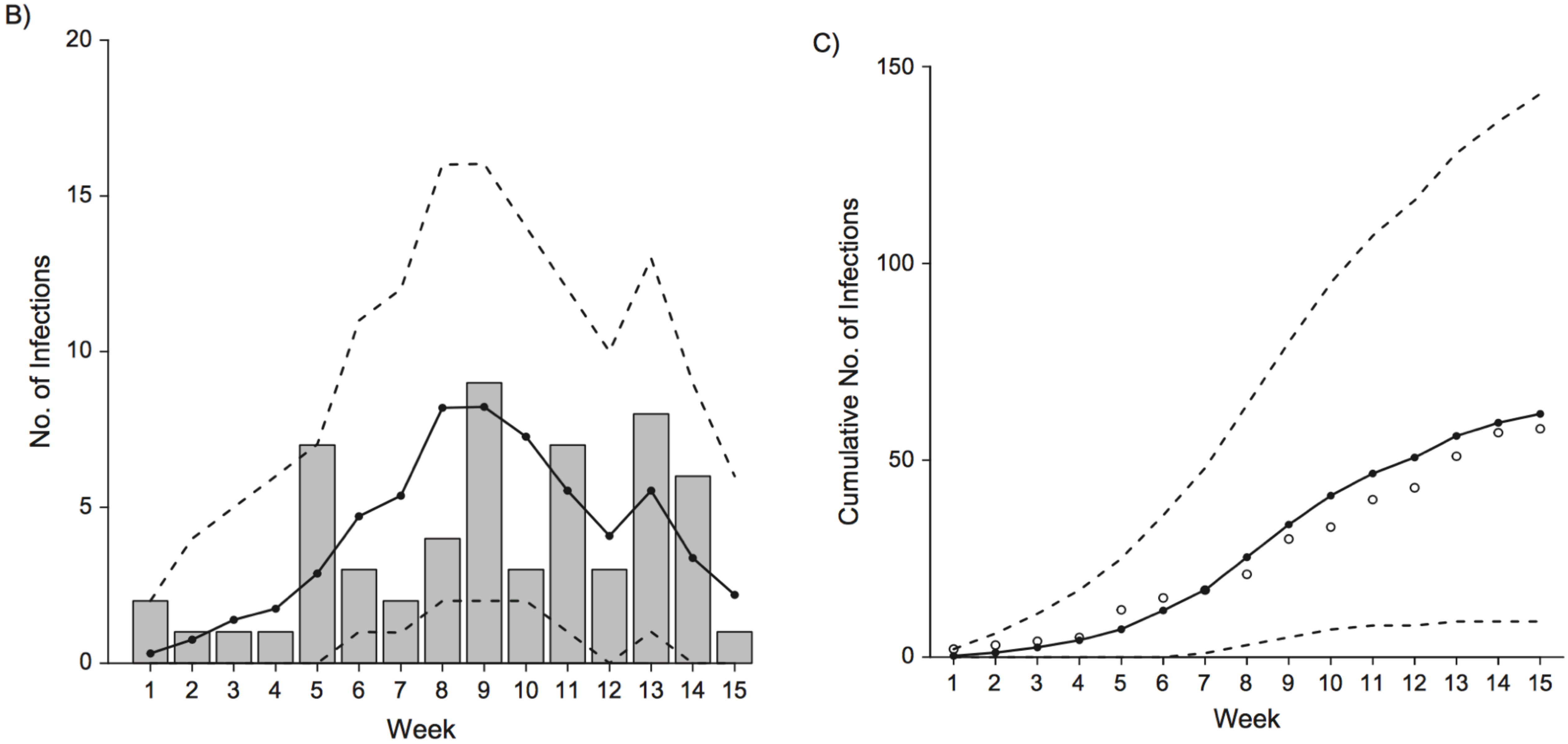
Table 2. Observed and Individual-Based Transmission Hazard Model–Predicted Influenza A(H3N2) Infections According to Infection Source, Age, Presence of High-Risk Health Condition, and Influenza Vaccination Status, Household Influenza Vaccine Effectiveness Study, Ann Arbor, Michigan, 2010–2011

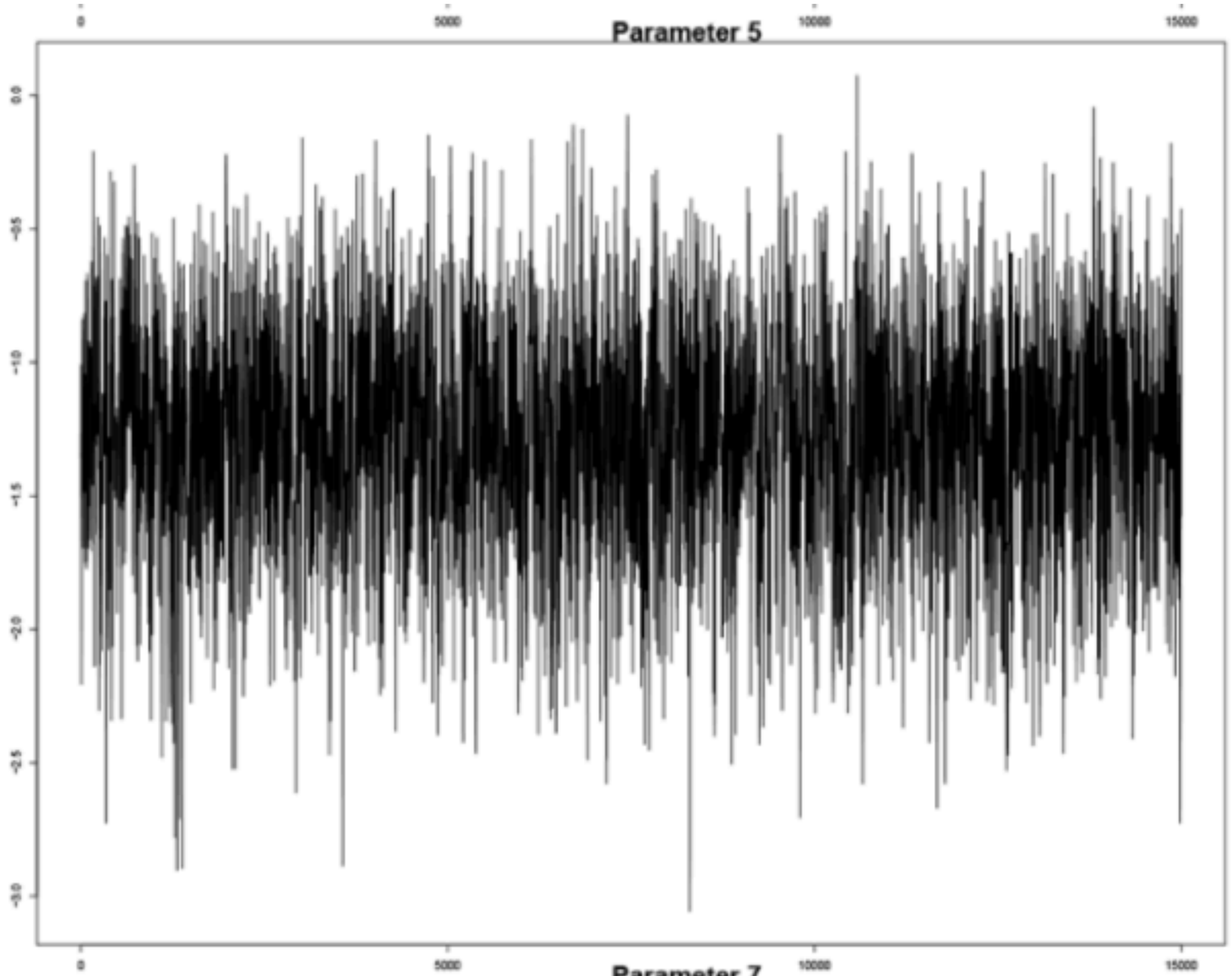
Characteristic	Observed Data			TH Model Predictions				
	No. of Cases (*n* = 58)	Total No. Exposed (*n* = 1,441)	% Positive	Median No. of Cases	95% CrI	% Positive	95% CrI	*P* Value[a]
Community-acquired	41	1,441	2.8	43	31, 55	3.0	2.2, 3.8	0.70
Household-acquired	17	111	15.3	18	9, 30	13.2	6.6, 20.5	
Secondary	N/O	N/O		15	7, 24			
Tertiary	N/O	N/O		3	0, 9			
Quaternary	N/O	N/O		0	0, 0			
Age category, years								0.80
<9	32	468	6.8	36	22, 50	7.7	4.7, 10.7	
9–17	8	371	2.2	8	3, 14	2.2	0.8, 3.8	
≥18	18	602	3.0	18	9, 27	3.0	1.5, 4.5	
Documented high-risk health condition								0.49
Any	6	162	3.7	5	1, 11	3.1	0.6, 6.8	
None	52	1,279	4.1	56	38, 76	4.4	3.0, 5.9	
Documented influenza vaccination[b]								0.45
Yes	33	864	3.8	32	19, 48	3.7	2.2, 5.6	
No	25	577	4.3	29	16, 44	5.0	2.8, 7.6	
Overall model predictions				62	42, 82	4.3	2.9, 5.7	

Abbreviations: CrI, credible interval; N/O, not observed; TH, transmission hazard.

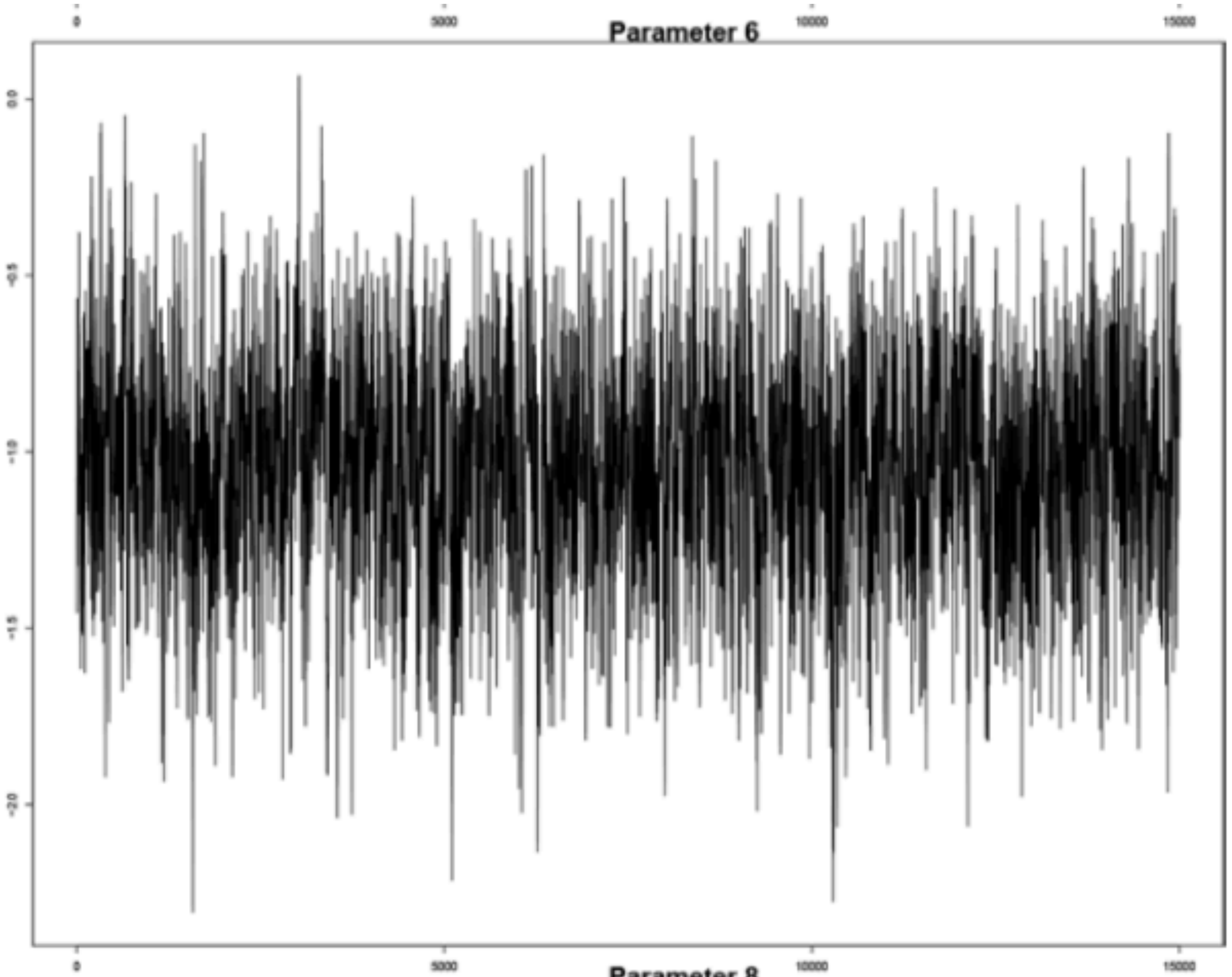
[a] Simulation-based χ^2 test.

[b] At least 1 dose of 2010–2011 influenza vaccine documented in the electronic medical record or state registry; vaccination must have occurred ≥14 days prior to illness onset for influenza A(H3N2) infected subjects.

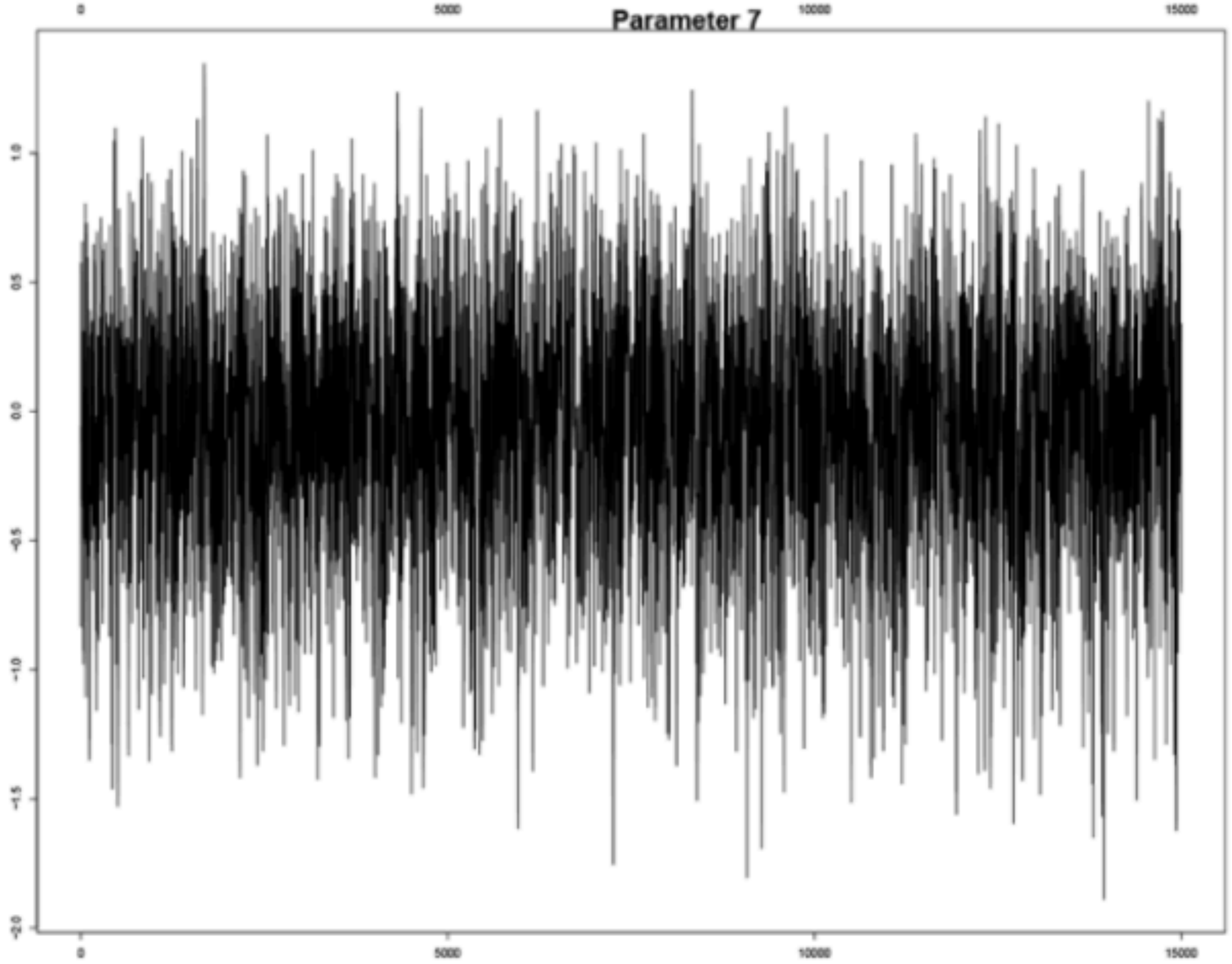

B)
20
15
10
5
0
No. of Infections
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
Week
C)
150
100
50
0
Cumulative No. of Infections
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
Week



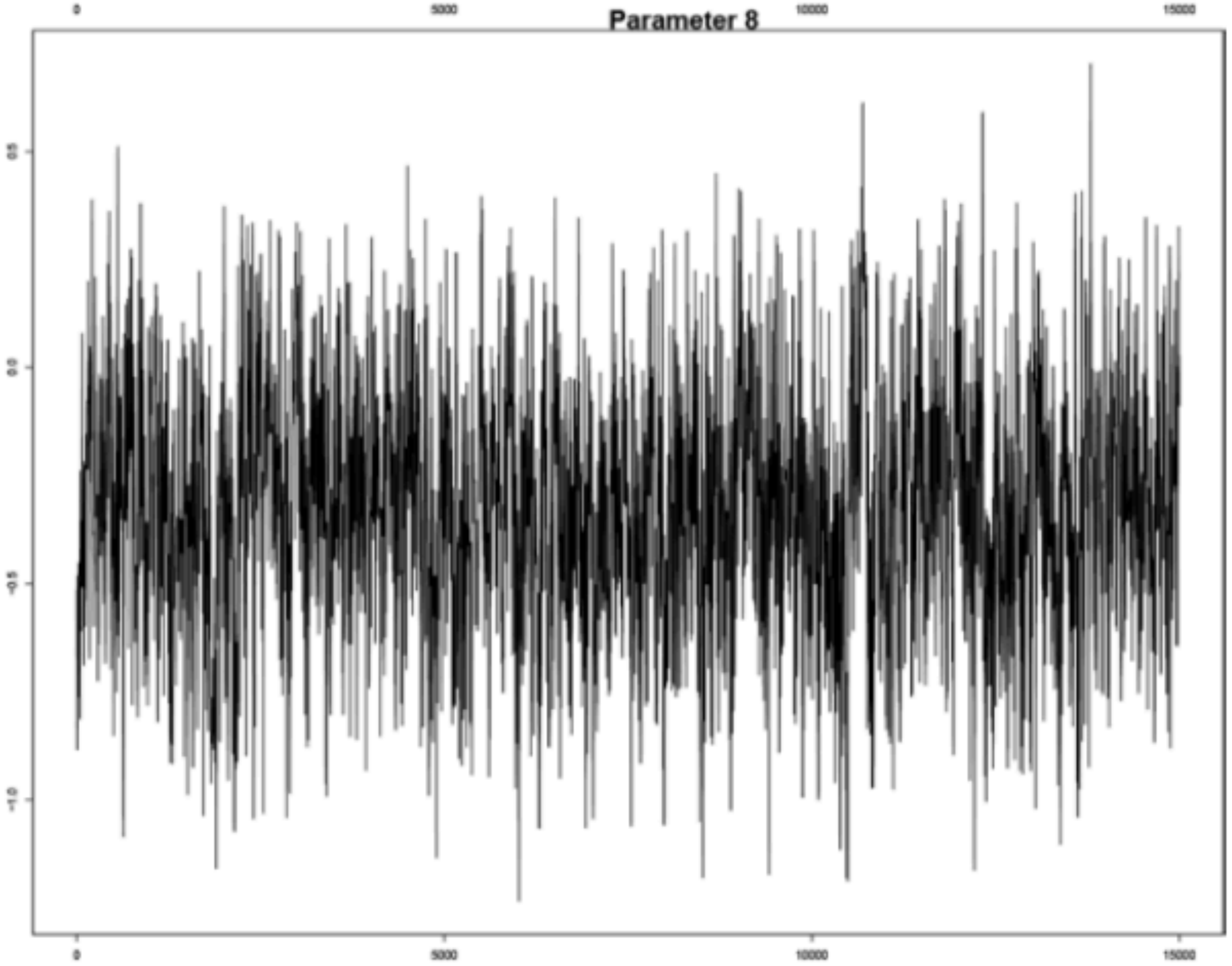
Parameter 5



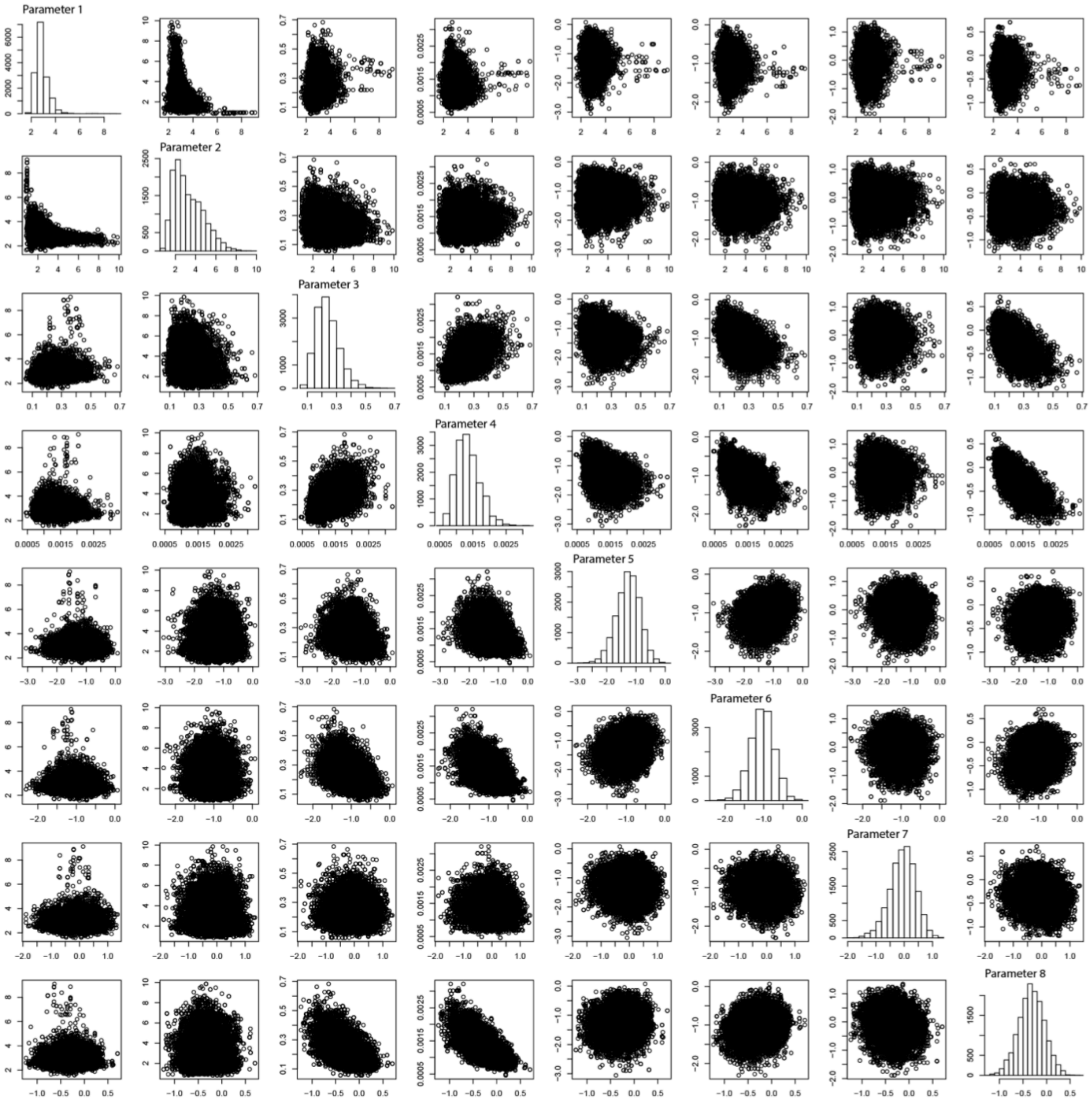
Parameter 6



Parameter 7



Parameter 8


Parameter 1
0
2000
4000
6000
2
4
6
8
Parameter 2
0
500
1500
2500
2
4
6
8
10
Parameter 3
0
1000
3000
0.1
0.3
0.5
0.7
Parameter 4
0
1000
2000
3000
0.0005
0.0015
0.0025
Parameter 5
0
1000
2000
3000
-3.0
-2.0
-1.0
0.0
Parameter 6
0
1000
3000
-2.0
-1.0
0.0
Parameter 7
0
500
1500
2500
-2.0
-1.0
0.0
1.0
Parameter 8
0
500
1500
-1.0
-0.5
0.0
0.5

MCMC code examples

- Basic normal distribution model with MCMC
- Mean-field SIR model MCMC
    - Recap model structure
    - Go through code
    - Illustrate identifiability issues

Sample Importance Resampling

- MCMC can be slow—another approach to getting a posterior sample is sample importance resampling
    - Can be used with the true likelihood
    - Or with an approximating function (e.g. approximate Bayesian computation, more on this in a bit)
- One of a bunch of related approaches in importance sampling/approximate Bayesian computation/etc)

Using reweighting to convert between distributions

- Starts with a sample of values drawn from probability distribution A, and suppose probability distribution B is our target distribution

- Set the weight of each element x to be $PDF_B(x)/PDF_A(x)$

- Sample from the list using the above weights (normalized to sum to 1 say)

- This results in a draw from distribution B, even though distribution A was used to generate the sample! (With caveats that you need to make your initial sample big enough etc.)

Sample Importance Resampling

- Draw a sample of parameters from your prior (either drawing at random or with LHS/Sobol/etc. sampling)

- Run the model for each sample

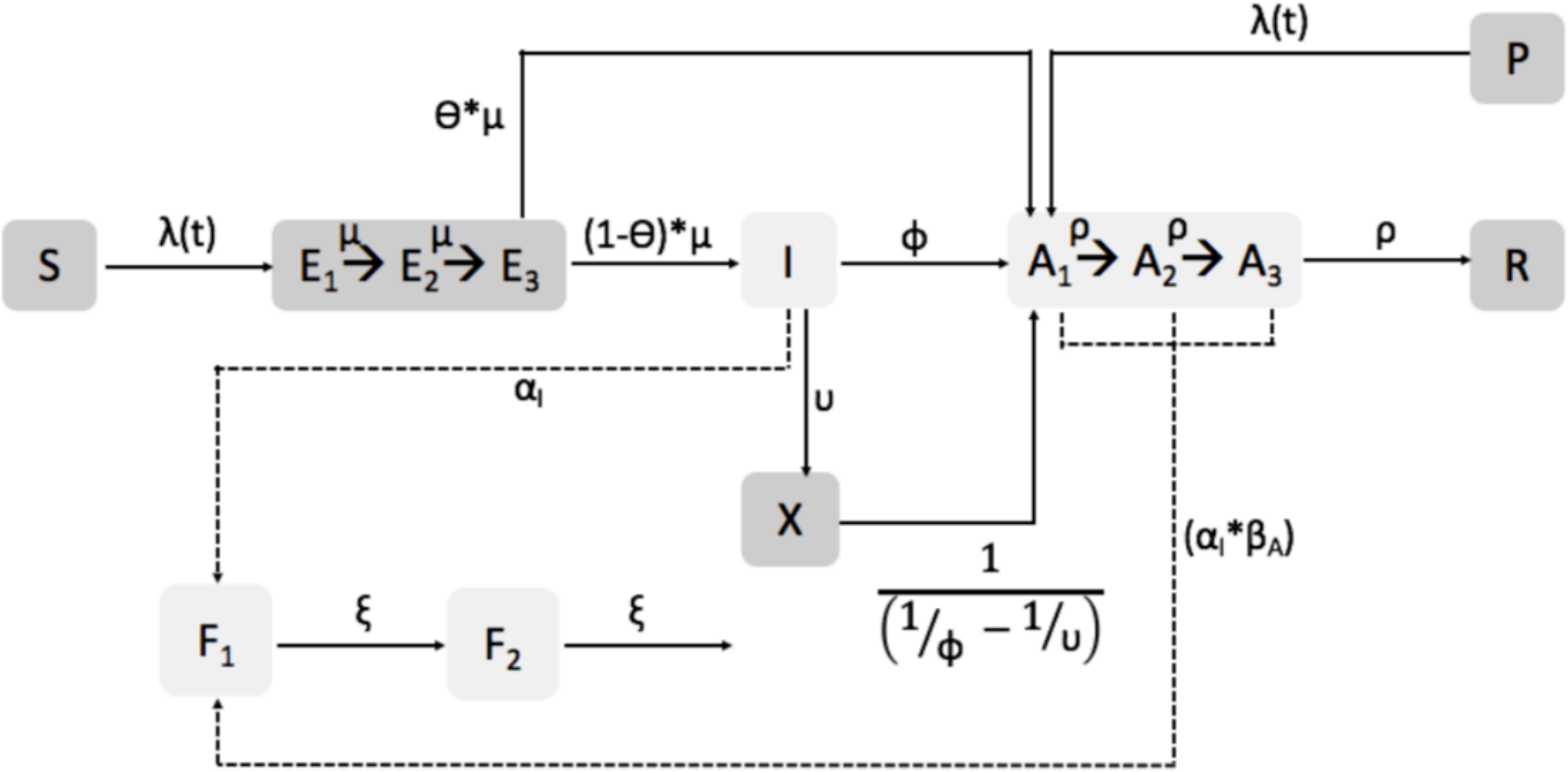
- Calculate the likelihood value for each sample

- Weight the samples based on the likelihood

- Resample to get the final set of samples

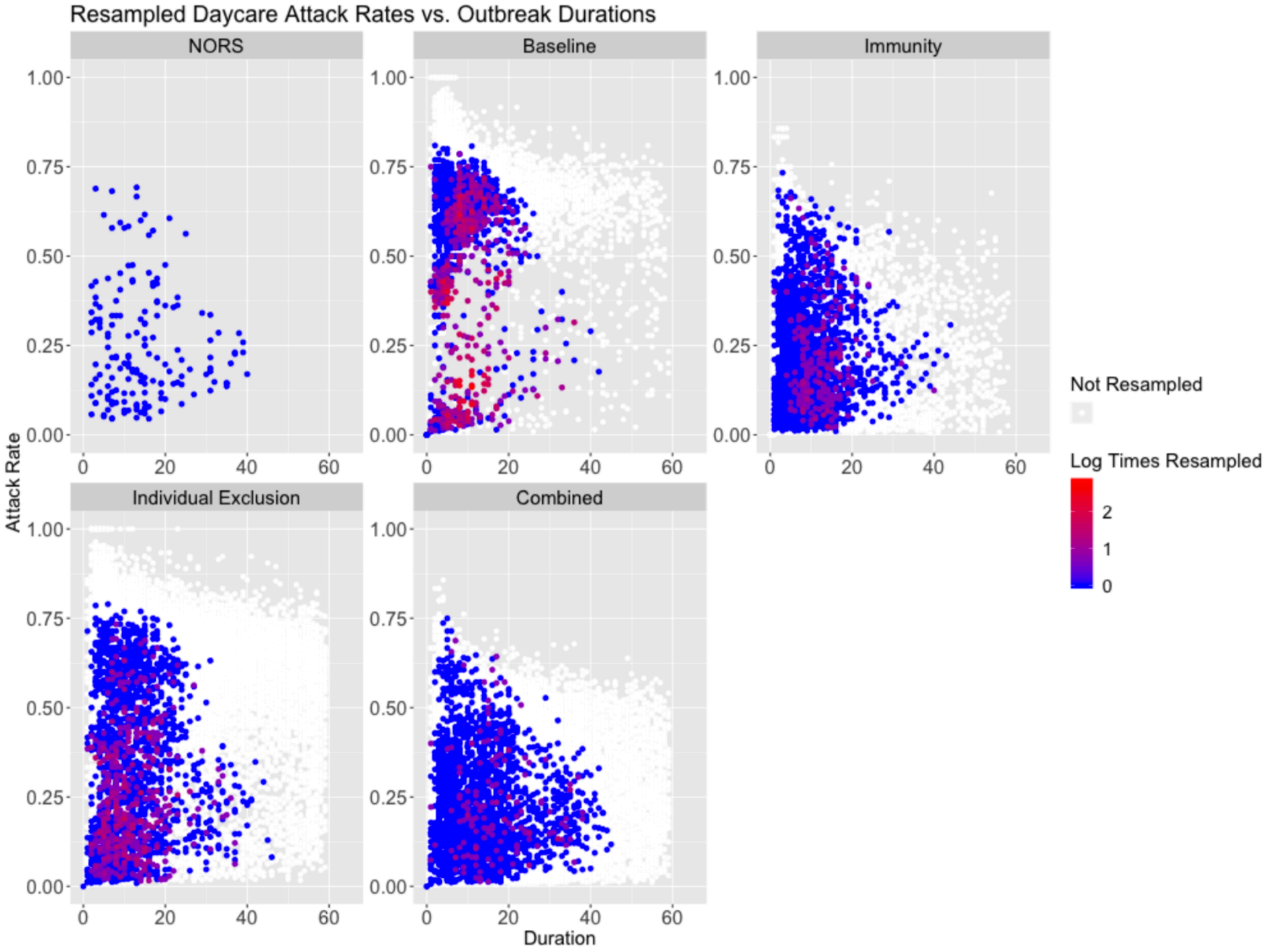
- The result follows the posterior distribution because you sample from the prior, and then weight with the likelihood (and we don’t divide by the prior in our weighting like in the previous example)

Example: Norovirus model



λ(t)
P
θ*μ
S
λ(t)
$E_1 \xrightarrow{\mu} E_2 \xrightarrow{\mu} E_3$
(1-θ)*μ
I
ϕ
$A_1 \xrightarrow{\rho} A_2 \xrightarrow{\rho} A_3$
ρ
R
α_I
υ
X
$(\alpha_I * \beta_A)$
$\frac{1}{(1/\phi - 1/\upsilon)}$
F_1
ξ
F_2
ξ

Havumaki et al. 2020


Resampled Daycare Attack Rates vs. Outbreak Durations
NORS
Baseline
Immunity
Individual Exclusion
Combined
Attack Rate
Duration
1.00
0.75
0.50
0.25
0.00
0
20
40
60
Not Resampled
Log Times Resampled
2
1
0

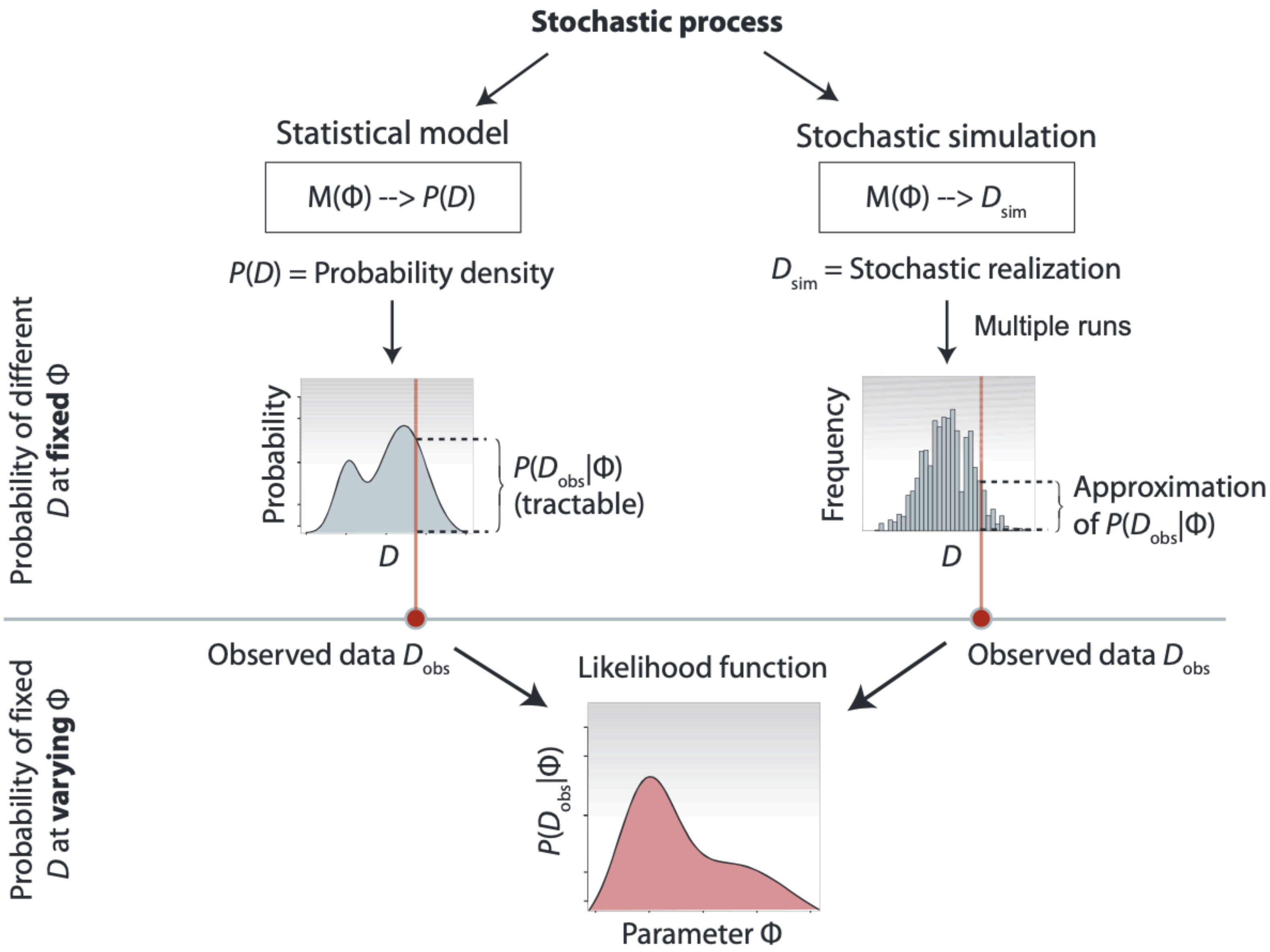
Likelihoods can be challenging to calculate for agent based and network models

- Do a little mini example where the network is known
- What if the network structure is unknown?
    - Some done for Erdos-Renyi graphs, other special networks
    - Exponential random graph models (e.g. see this review for SIR dynamics)
    - Depends on what data you observe too
- But often quite difficult, especially for more general ABMs or networks

So what to do?

- Could numerically approximate the likelihood via sampling (i.e. figure out the probability of observing the data for a given set of parameters by sampling many times from those parameter values, and then do this across parameter space)

- Very computationally intensive



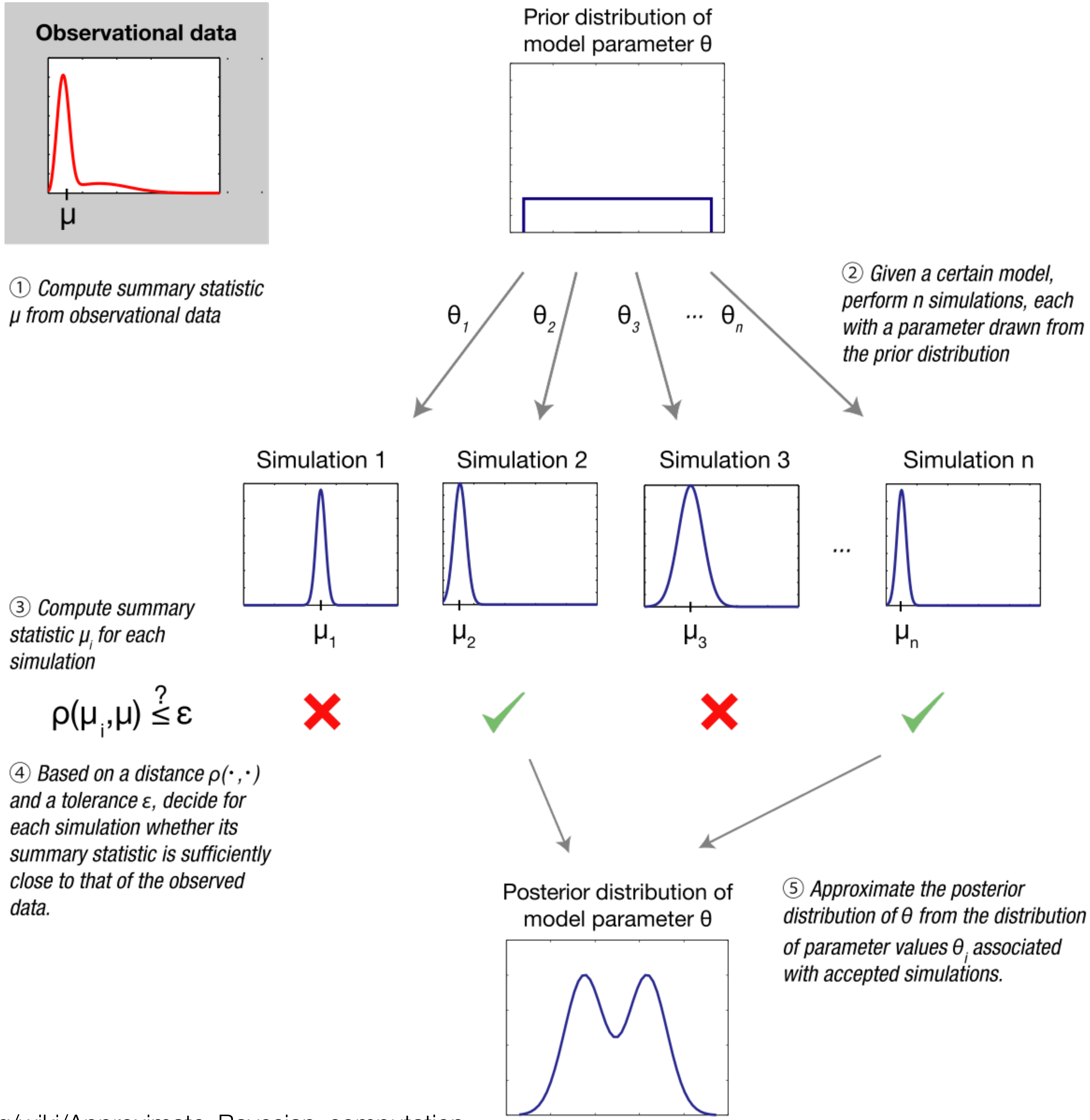
https://onlinelibrary.wiley.com/doi/epdf/10.1111/j.1461-0248.2011.01640.x

So what to do?

- Another alternative is **approximate Bayesian computation (ABC)**
- Similar to rejection sampling or sample-importance resampling
- Approximate the posterior by sampling from the prior and then selecting only those samples that match the data closely (within some threshold for however we define ‘close’)

ABC: Approximate Bayesian Computation

- Basic idea
  - Choose a function to measure the distance between model and data (goodness of fit), typically based on some sort of **summary statistic** of the model fit
  - Sample from the prior
  - Keep only those samples that fall within a threshold based on the summary statistic
  - Resulting distribution of parameter samples should approximate the posterior (**if** you choose a good summary statistic!)



① *Compute summary statistic μ from observational data*

② *Given a certain model, perform n simulations, each with a parameter drawn from the prior distribution*

③ *Compute summary statistic μ_i for each simulation*

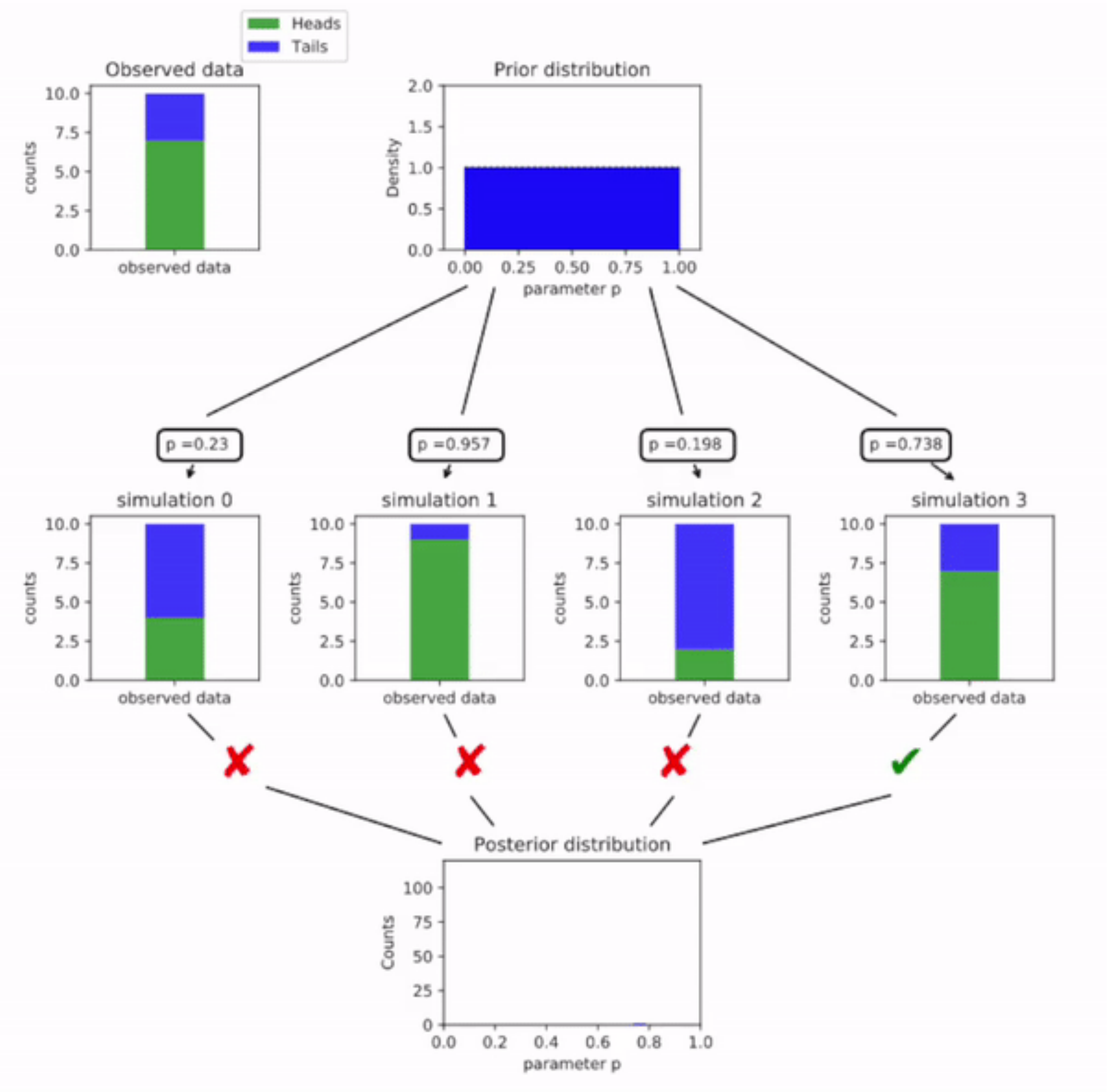
$$\rho(\mu_i,\mu) \overset{?}{\leq} \varepsilon$$

④ *Based on a distance $\rho(\cdot,\cdot)$ and a tolerance ε, decide for each simulation whether its summary statistic is sufficiently close to that of the observed data.*

⑤ *Approximate the posterior distribution of θ from the distribution of parameter values θ_i associated with accepted simulations.*

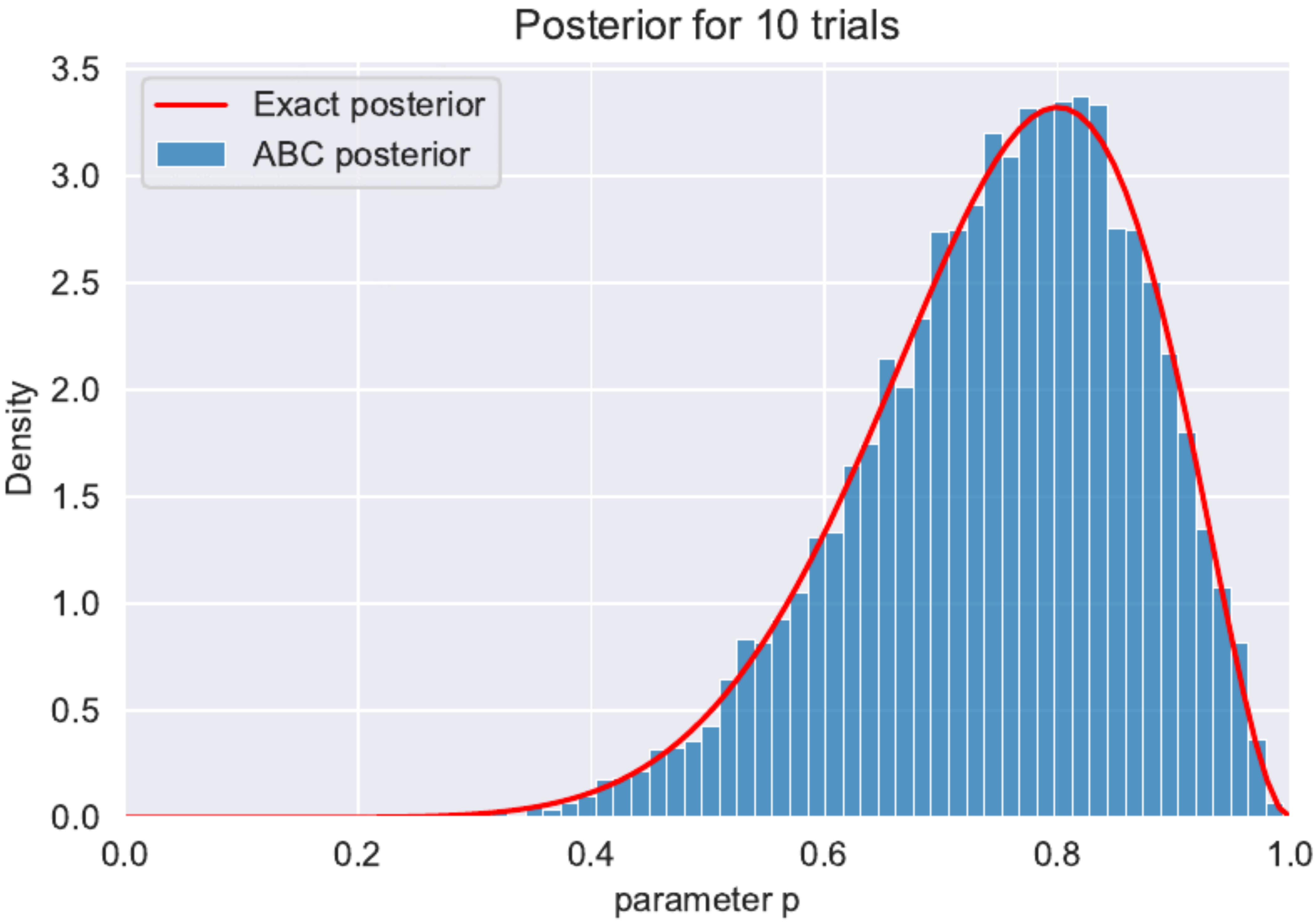
https://en.wikipedia.org/wiki/Approximate_Bayesian_computation

Coin flip example: p = probability of heads distance function = |observed H - simulated H|



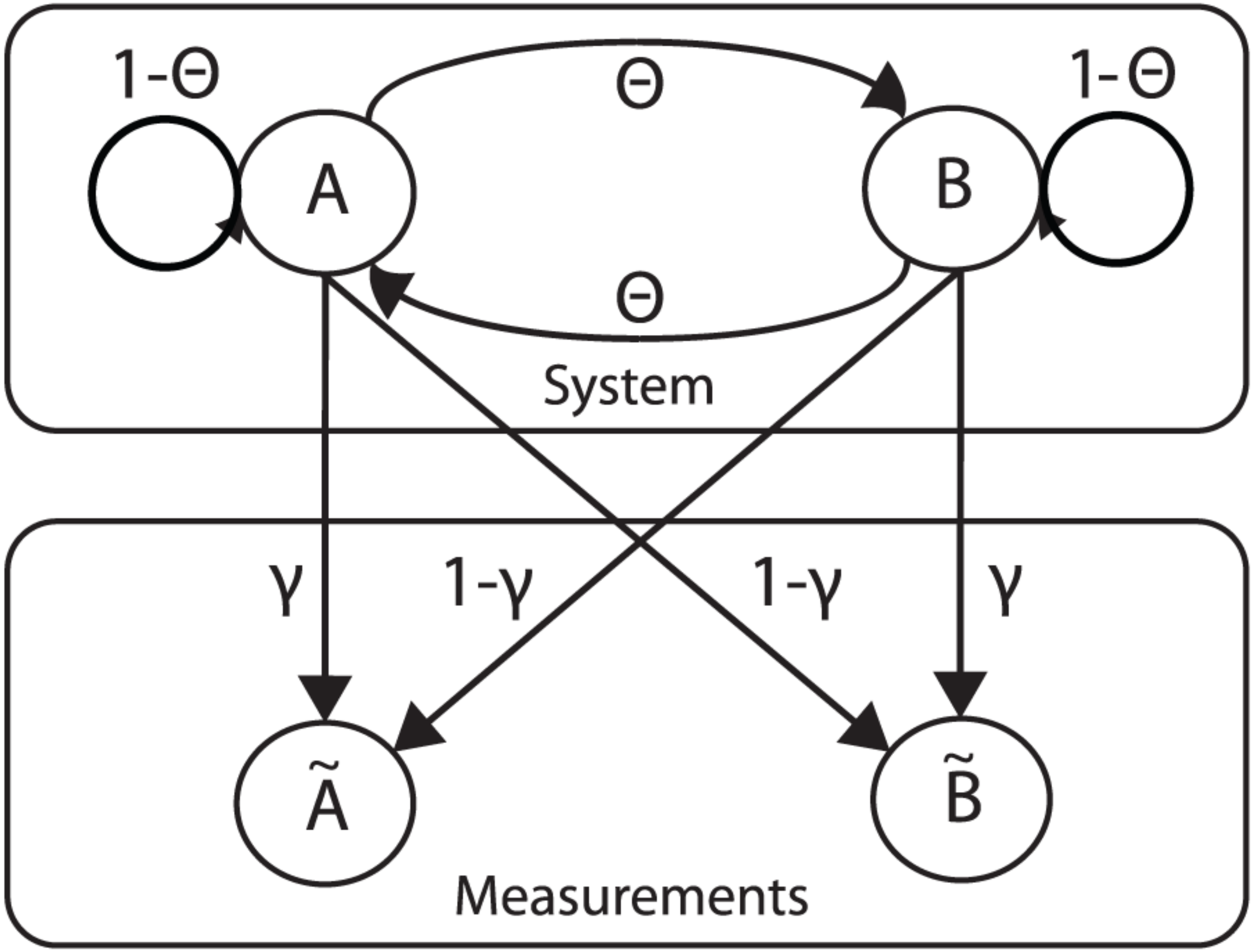
https://towardsdatascience.com/the-abcs-of-approximate-bayesian-computation-bfe11b8ca341

How well do we do?



https://towardsdatascience.com/the-abcs-of-approximate-bayesian-computation-bfe11b8ca341

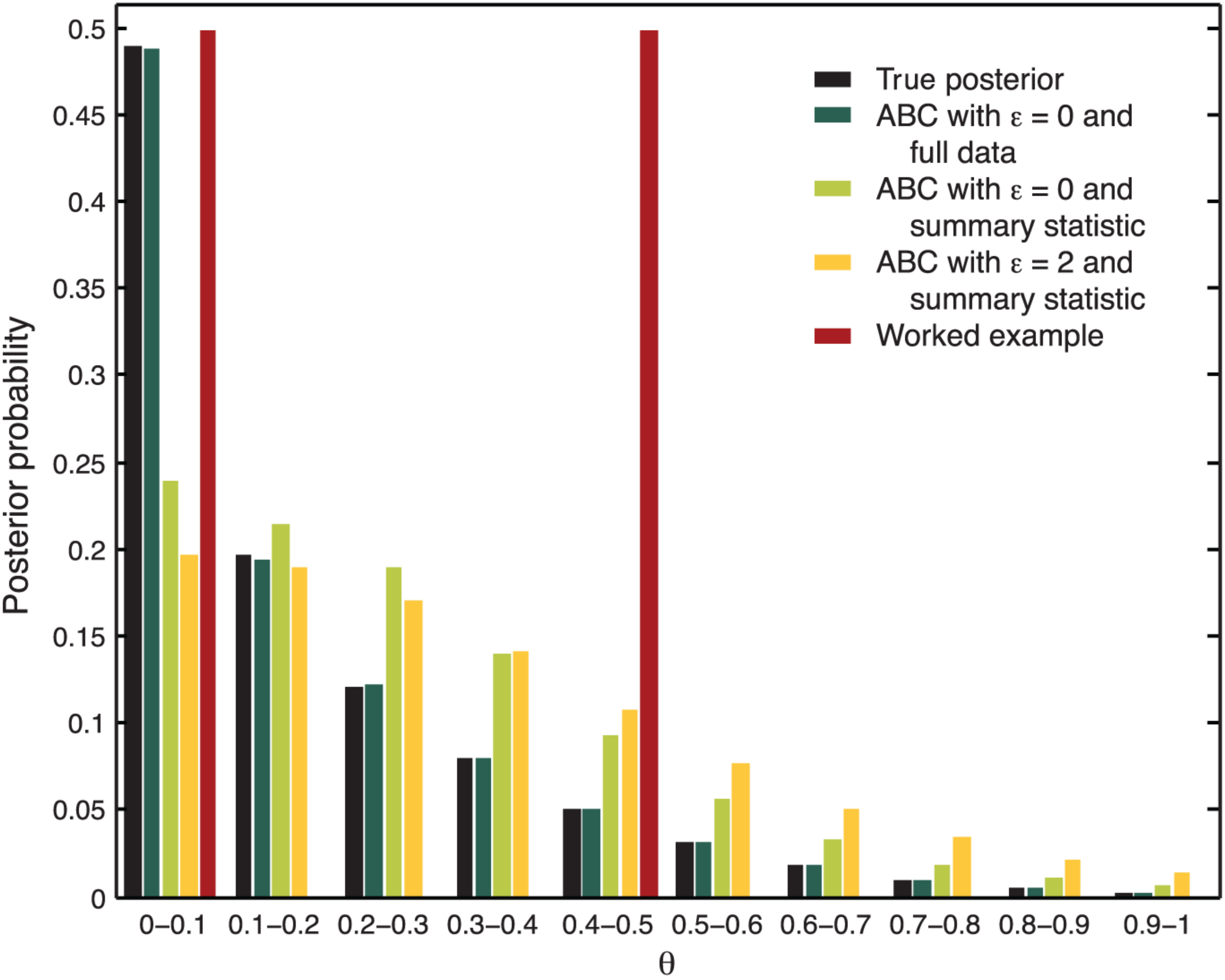
Another example: Markov process



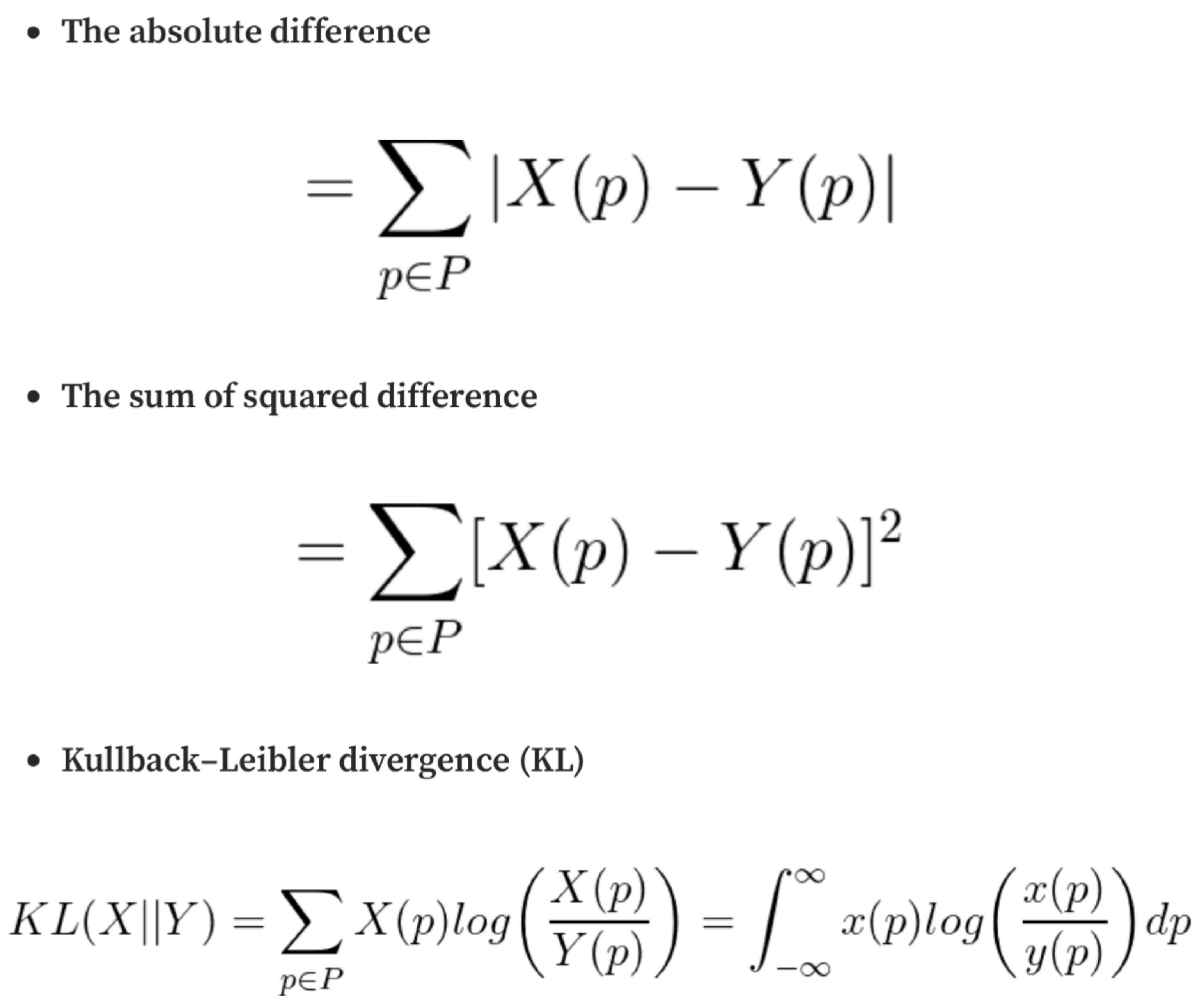
https://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1002803

i	θ_i	Simulated Datasets (Step 2)	Summary Statistic $\omega_{S,i}$ (Step 3)	Distance ρ $(\omega_{S,i},\omega_E)$ (Step 4)	Outcome (Step 4)
1	0.08	AABAAAABAABAAABAAAAA	8	2	accepted
2	0.68	AABBABABAAABBABABBAB	13	7	rejected
3	0.87	BBBABBABBBBABABBBBBA	9	3	rejected
4	0.43	AABAAAAABBABBBBBBBBA	6	0	accepted
5	0.53	ABBBBBAABBABBABAABBB	9	3	rejected

doi:10.1371/journal.pcbi.1002803.t001


⬛ True posterior
🟩 ABC with ε = 0 and full data
🟩 ABC with ε = 0 and summary statistic
🟨 ABC with ε = 2 and summary statistic
🟥 Worked example
Posterior probability
0
0.05
0.1
0.15
0.2
0.25
0.3
0.35
0.4
0.45
0.5
0–0.1
0.1–0.2
0.2–0.3
0.3–0.4
0.4–0.5
0.5–0.6
0.6–0.7
0.7–0.8
0.8–0.9
0.9–1
θ

Summary statistics and threshold



- **The absolute difference**

$$= \sum_{p \in P} |X(p) - Y(p)|$$

- **The sum of squared difference**

$$= \sum_{p \in P} [X(p) - Y(p)]^2$$

- **Kullback–Leibler divergence (KL)**

$$KL(X||Y) = \sum_{p \in P} X(p) log\left(\frac{X(p)}{Y(p)}\right) = \int_{-\infty}^{\infty} x(p) log\left(\frac{x(p)}{y(p)}\right) dp$$

Summary statistic and threshold

- These are very tricky to choose!
- Often use an adaptive threshold
- Sufficient statistic: one that gives the same distribution (often in the large N limit) as the true likelihood
- Sufficient statistics are tough to sort out—often test several candidate statistics (e.g. subset selection and projection methods), and potentially use a simpler model for testing (one where the likelihood is more tractable)

Sufficient summary statistics

- Often people pick things like comparing statistical moments (mean, variance, etc) of the data vs. model simulation as a summary statistic in hopes that this will be a sufficient statistic

- Or many people fit a descriptive model (e.g. a distribution or regression model) to the data, and then also fit the same model to the model simulation, and compare the values

Sufficient summary statistics

- Hard to prove sufficiency!

- Typically test with simulated data (i.e. simulate the data and then try to estimate the parameters using a given summary statistic and see how you do)

- Or with simpler examples of the model where sufficiency can be tested/proven more easily

Choosing the threshold for ABC

- Ideally would sort of want a threshold of 0 (so that you're always sampling from points in parameter space that can generate the exact data you observed)

- But realistically this will often lead to no accepted samples

- Alternatively, if the threshold is too high, all samples will be accepted and you just get back a sample of the prior

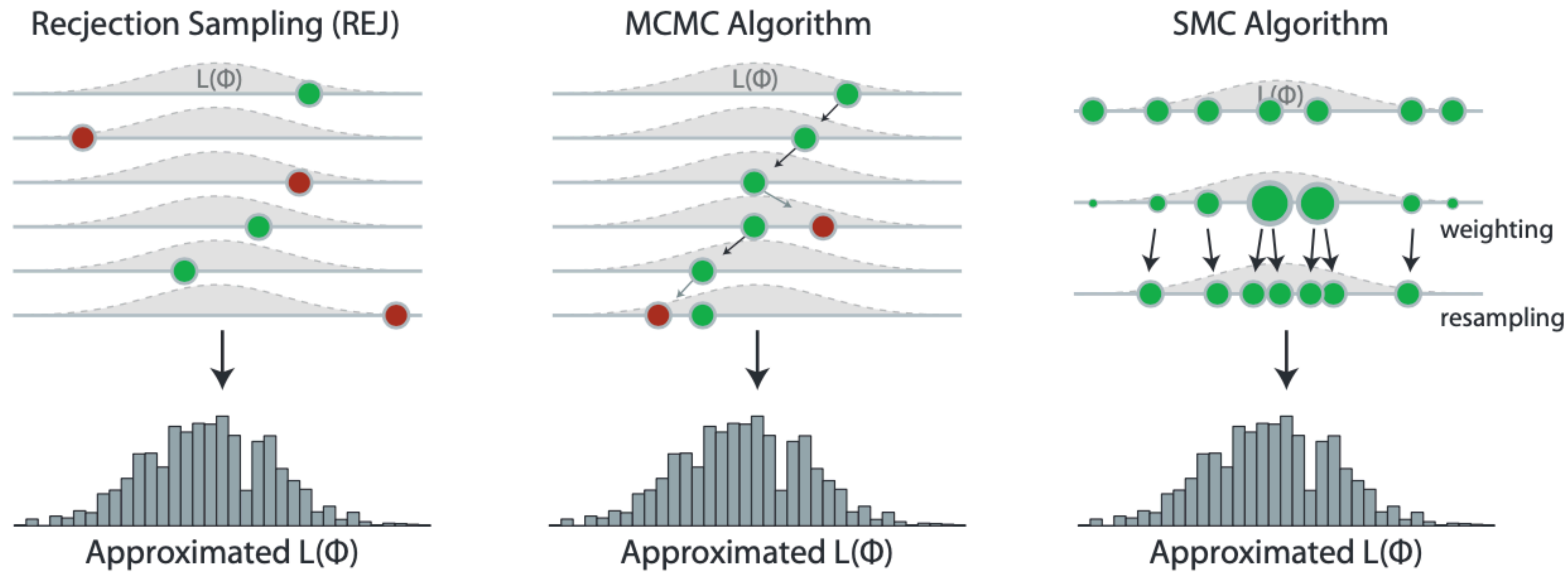
- In general try to balance small threshold with computational constraints

Error Source	Potential Issue	Solution	Subsection
Nonzero tolerance ε	The inexactness introduces a bias in the computed posterior distribution.	Theoretical/practical studies of the sensitivity of the posterior distribution to the tolerance. Noisy ABC.	Approximation of the posterior
Nonsufficient summary statistics	The information loss causes inflated credible intervals.	Automatic selection/semi-automatic identification of sufficient statistics. Model validation checks (e.g., Templeton 2009 [19]).	Choice and sufficiency of summary statistics
Small number of models/mis-specified models	The investigated models are not representative/lack predictive power.	Careful selection of models. Evaluation of the predictive power.	Small number of models
Priors and parameter ranges	Conclusions may be sensitive to the choice of priors. Model choice may be meaningless.	Check sensitivity of Bayes factors to the choice of priors. Some theoretical results regarding choice of priors are available. Use alternative methods for model validation.	Prior distribution and parameter ranges
Curse-of-dimensionality	Low parameter acceptance rates. Model errors cannot be distinguished from an insufficient exploration of the parameter space. Risk of overfitting.	Methods for model reduction if applicable. Methods to speed up the parameter exploration. Quality controls to detect overfitting.	Curse-of-dimensionality
Model ranking with summary statistics	The computation of Bayes factors on summary statistics may not be related to the Bayes factors on the original data, which may therefore render the results meaningless.	Only use summary statistics that fulfill the necessary and sufficient conditions to produce a consistent Bayesian model choice. Use alternative methods for model validation.	Bayes factor with ABC and summary statistics
Implementation	Low protection to common assumptions in the simulation and the inference process.	Sanity checks of results. Standardization of software.	Indispensable quality controls

doi:10.1371/journal.pcbi.1002803.t002

ABC: Approximate Bayesian Computation

- The rejection sampling method is very common for ABC
- But you can also do MCMC, sample importance resampling, etc, but use the summary statistic instead of the likelihood (e.g. ABC-MCMC is a thing people use a lot!)
- See this review for more (and this post for associated code)



1) Draw a parameter Φ
2) Calculate L(Φ)
3) Accept proportional to L(Φ)

1) Draw new parameter Φ′ close to the old Φ
2) Calculate L(Φ′)
3) Jump proportional to L(Φ′)/L(Φ)

1) Last set of parameters $\{\Phi_i\}$
2) Assign weight ω_i proportional to $L(\Phi_i)$
3) Draw new $\{\Phi_i\}$ based on the ω_i

https://onlinelibrary.wiley.com/doi/epdf/10.1111/j.1461-0248.2011.01640.x

Readings

- Menzies NA, Soeteman DI, Pandya A, Kim JJ. Bayesian methods for calibrating health policy models: a tutorial. PharmacoEconomics. 2017 Jun 1;35(6):613-24.