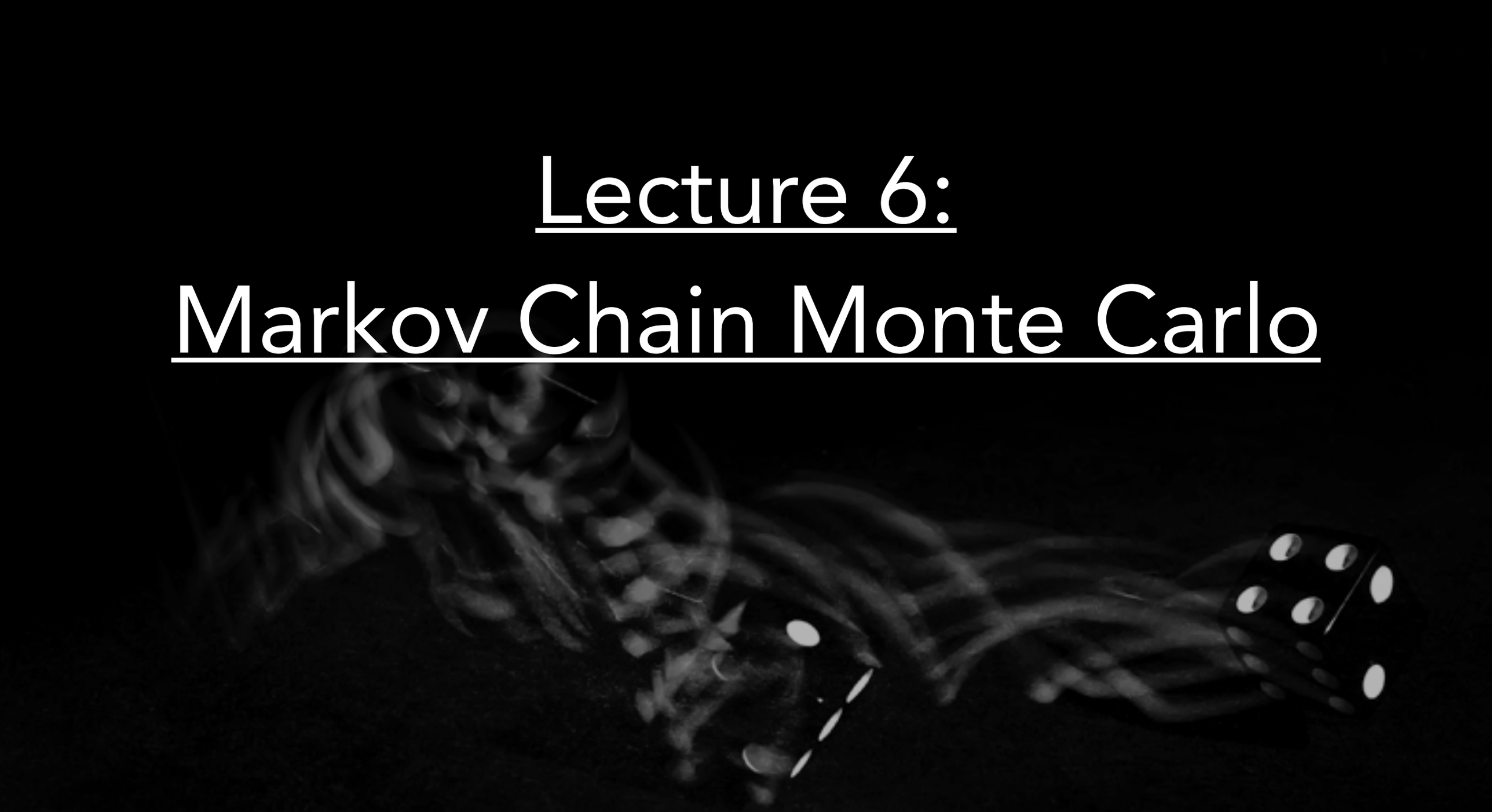
Lecture 6:
Markov Chain Monte Carlo



D. Jason Koskinen
koskinen@nbi.ku.dk

Advanced Methods in Applied Statistics
Feb - Apr 2019

Photo by Howard Jackman

Outline

- Bayes Recap
- Markov Chain
- Markov Chain Monte Carlo

*Material drawn from R. M. Neal, C. Chan, and wikipedia

Bayes Theorem

- We have Bayes' theorem

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

- or sometimes

(Discrete)

$$P(A|B) = \frac{P(B|A)P(A)}{\sum_i P(B|A_i)P(A_i)}$$

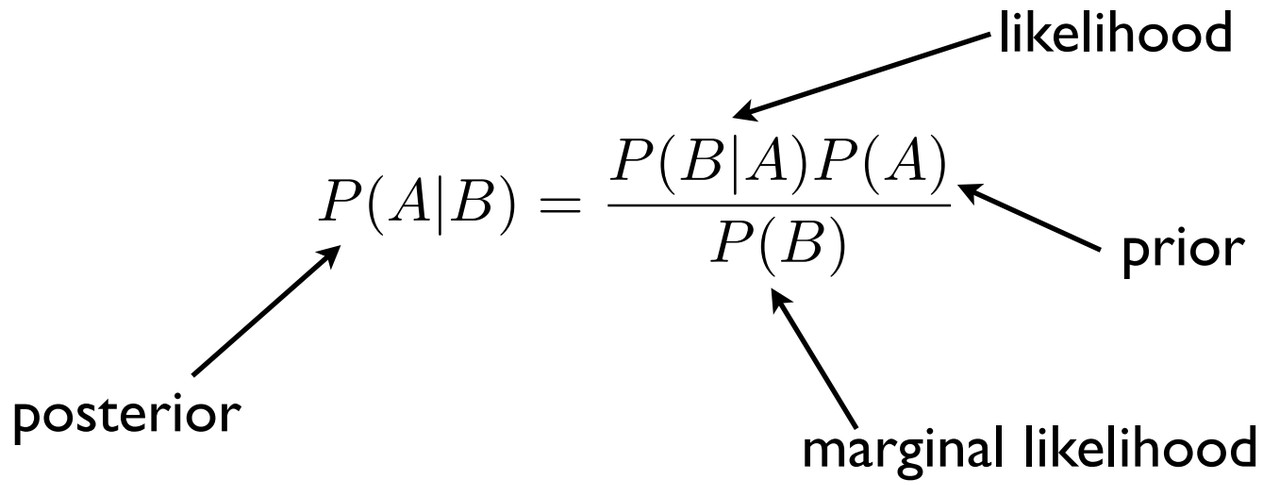
(Continuous)

$$\frac{P(B|A)P(A)}{\int P(B|A)P(A)dA}$$

- Let B be the observed data and A be the model/theory parameters, then we often want the P(A|B); the posterior probability distribution conditional on having observed B.

Bayes' Theorem

- One can solve the respective conditional probability equations for P(A and B) and P(B and A), setting them equal to give Bayes' theorem:



$$\text{posterior} \propto \text{prior} \times \text{likelihood}$$

- In the previous lecture we avoided dealing with the marginal likelihood, i.e. the normalizing constant, because it does not depend on the parameter(s) A. But it is an important value in order to get an accurate posterior distribution which is a useable probability.

Common PDF (Beta Distribution)

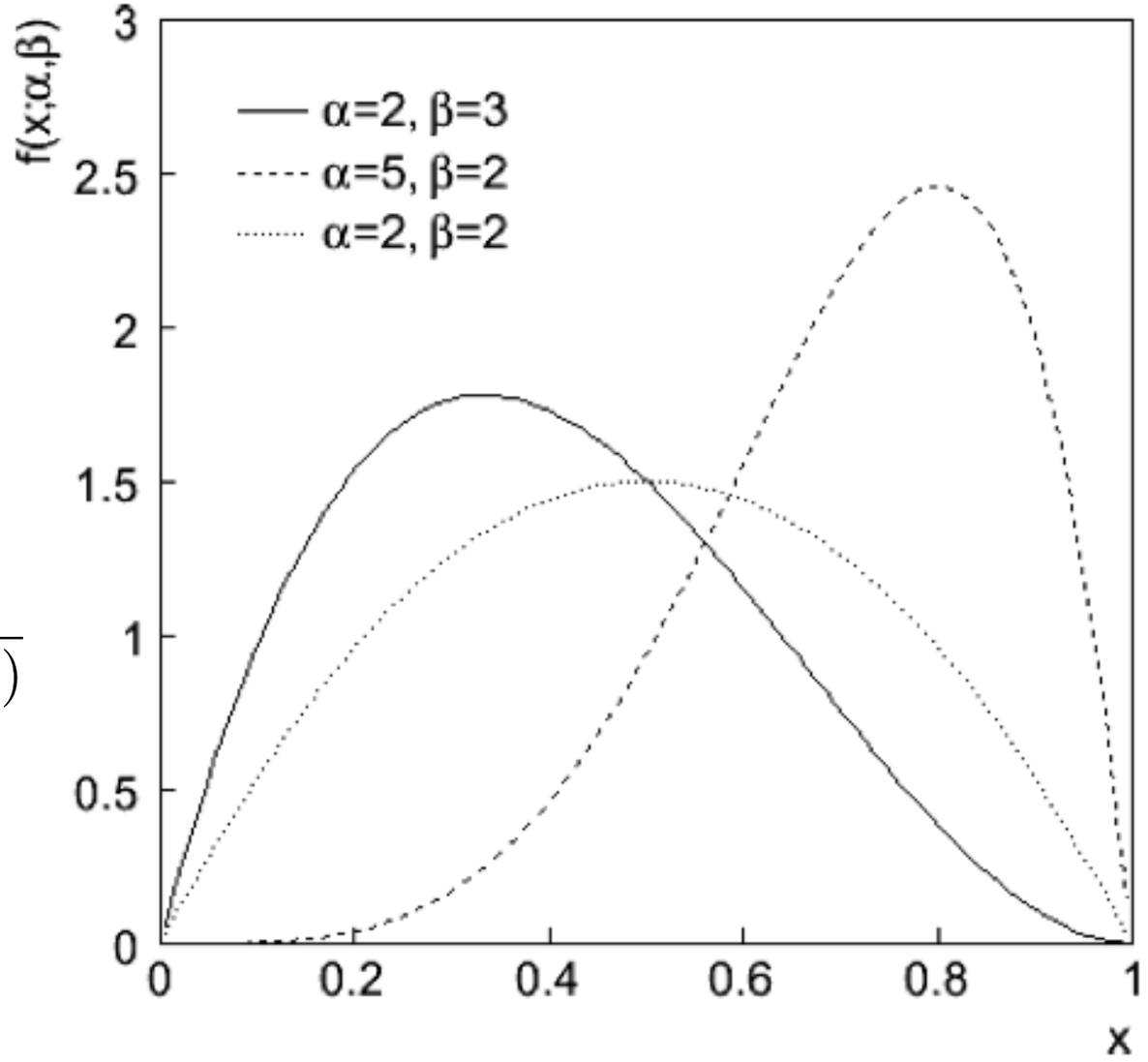
- Beta Distribution
  - for a continuous random variable (x)

$$f(x;\alpha,\beta) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} x^{\alpha-1}(1-x)^{\beta-1}$$

  - Expectation and Variance:

$$E[x] = \frac{\alpha}{\alpha+\beta} \qquad V[x] = \frac{\alpha\beta}{(\alpha+\beta)^2(\alpha+\beta+1)}$$

  - Often used to represent a PDF of cc
    non-zero only between finite limits



Exercise #1

- Coin flipping bias with **n** throws/flips, but now in Bayesian where we want the prob. of coming up heads (θ)
  - Likelihood is the standard binomial
  - Prior here is the beta-distribution with $\boldsymbol{\alpha}$=5 and $\boldsymbol{\beta}$=17
  - Plot the priors, likelihood, and posteriors for n=100 coin flips with heads=66
- This is the repeat of a previous lectures exercise, but now with different priors, likelihoods, and posteriors
- Normalize them to be on the same scale for plotting
  - Does not have to be normalized to 1
  - Normalizations for a binomial likelihood (and posterior distributions using a binomial likelihood) are summations instead of integrals

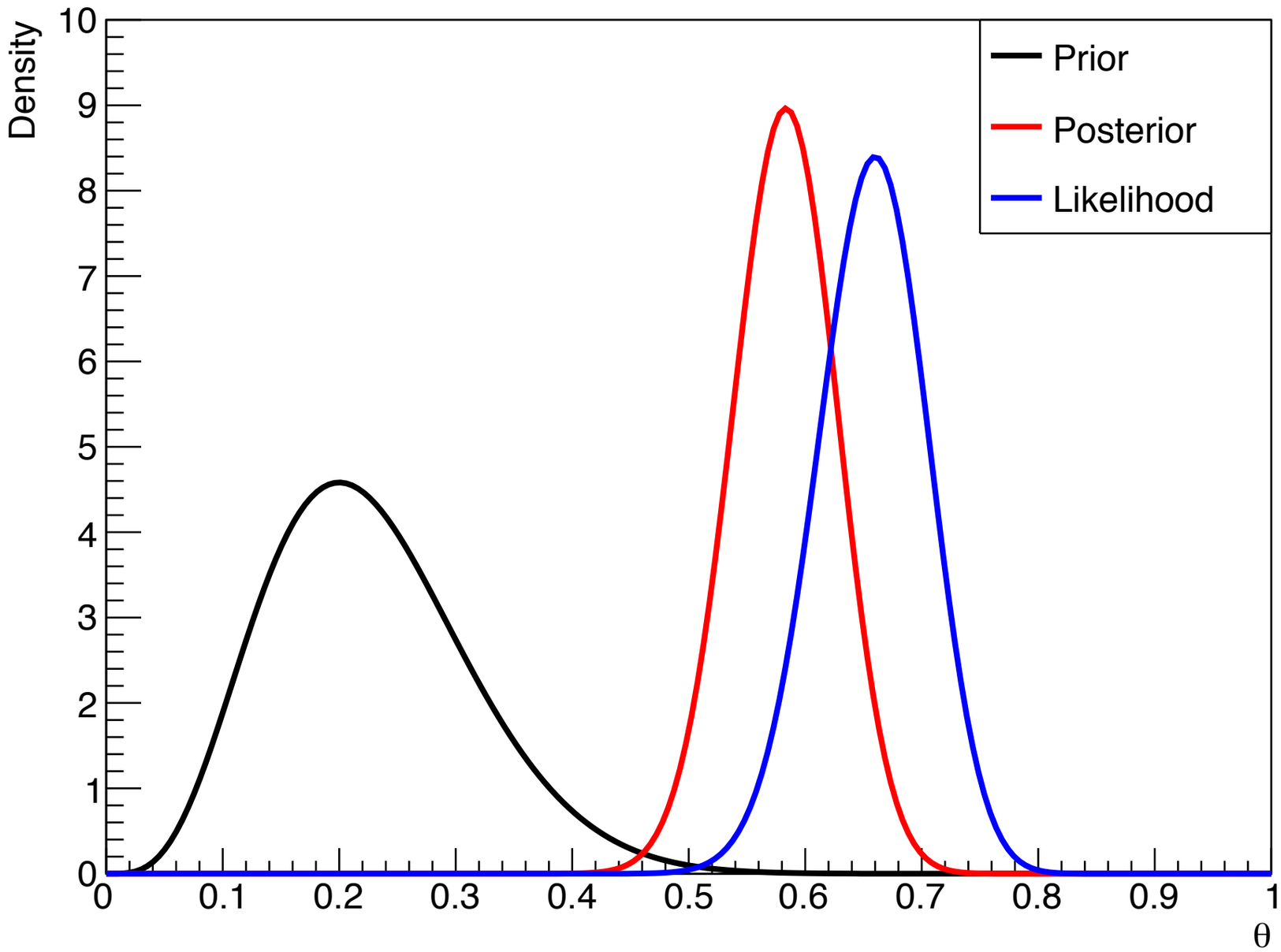
(continued on next slide)

Exercise #1 (cont.)

- Remember that for Bayesian analyses we include all possible values of the parameter, i.e. θ
  - This means for the PDF, it will not be calculated at a single value of θ, but over a suitable range of 0-1
  - The likelihood PDF is now technically a 'probability mass function' because it is discrete
- Many packages have the binomial PMF (scipy.stats.binom.pmf(k, n, p))
  - I'll point out again that the above scipy function is a calculation for a **single** probability p, but we want to scan θ over a range from 0-1

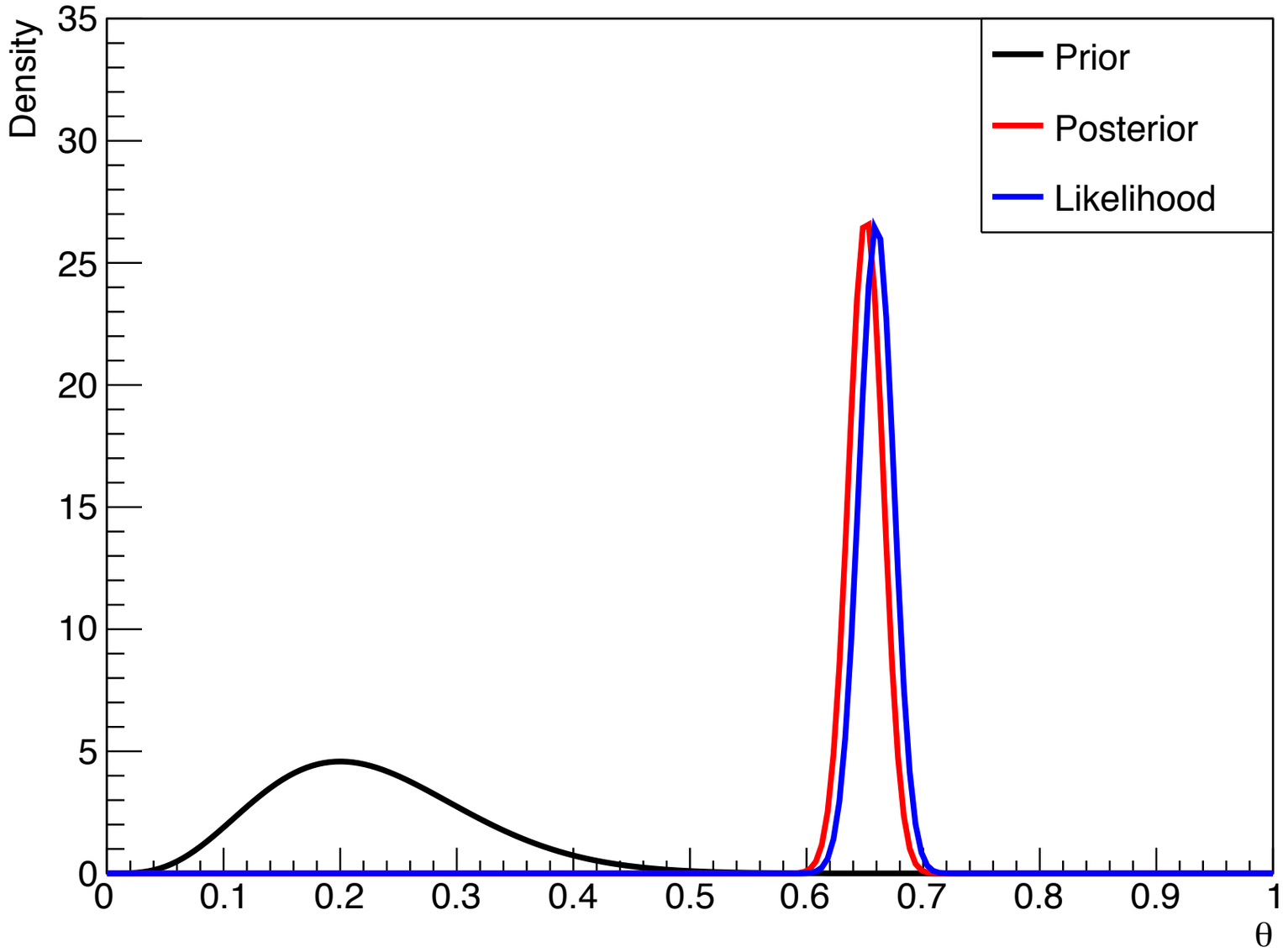
Exercise #1 plot

- Compare to a situation with 10x more data (scale n and ‘heads’ by 10)



Exercise #1 (cont.)

- With 10x more statistics, an obvious feature pops up, i.e. that as n→infinity the maximum a posterior (MAP) approaches the maximum likelihood estimator (MLE), irrespective of the prior



Numerical Limitations

- The previous example had only 1 parameter (θ) and 1 prior, so it is not computationally intensive to produce the marginal likelihood, prior, or likelihood for all relevant values by summation/integration. But, when dealing with more parameters, the computational load approx. increases exponentially with the number of parameters.
  - For summation — or integration via Monte Carlo sampling — the number of points (n) grows as $\mathcal{O}(n^d)$ if n points are used to cover each parameter (d)
  - It's possible to tune the number of scan or Monte Carlo points, but then the number of points necessary for calculation is the product of the number of points: $\prod_{i=1}^{d} n_i$

Posterior Distribution Sampling

- In order to estimate joint likelihood probability functions, e.g. likelihoods with multiple parameters, we can use Monte Carlo integration and sampling to estimate the distribution
- In order to estimate Bayesian posterior distributions, it would be nice to use the same strategy when confronted by higher dimension integrals, e.g. for the marginal likelihood. But, it is often difficult — sometimes impossible — to sample from the posterior distribution effectively using the same approach as what we can do for likelihood distributions
- Solution is to use Markov chains which converge to the posterior distribution

Markov Chain

- A Markov chain is a stochastic process (random walk) of steps or state transitions that is only conditional on the current step/state. The transition from the current state to the next state is often, but not always, governed by a probability

$$new\ state = f(current\ state, transition\ probability)$$

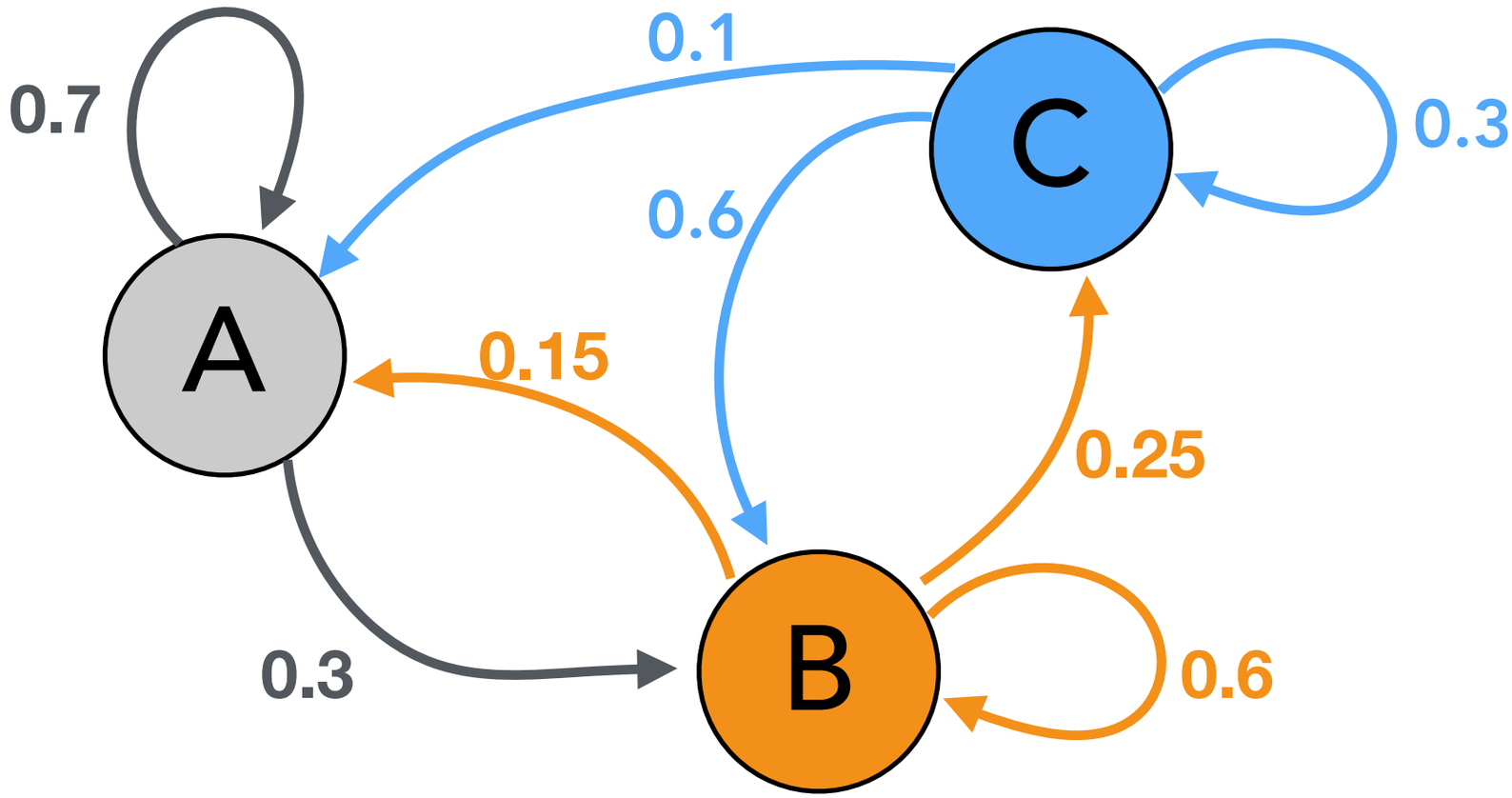
- More math-like: the conditional transition of X_{t+1} **only** depends on X_t and not X_1, X_2,…, X_{t-1}

$$P(X_{t+1}|X_1, X_2, ..., X_t) = P(X_{t+1}|X_t)$$

- In this way Markov chains have no memory or dependence on previous transitions

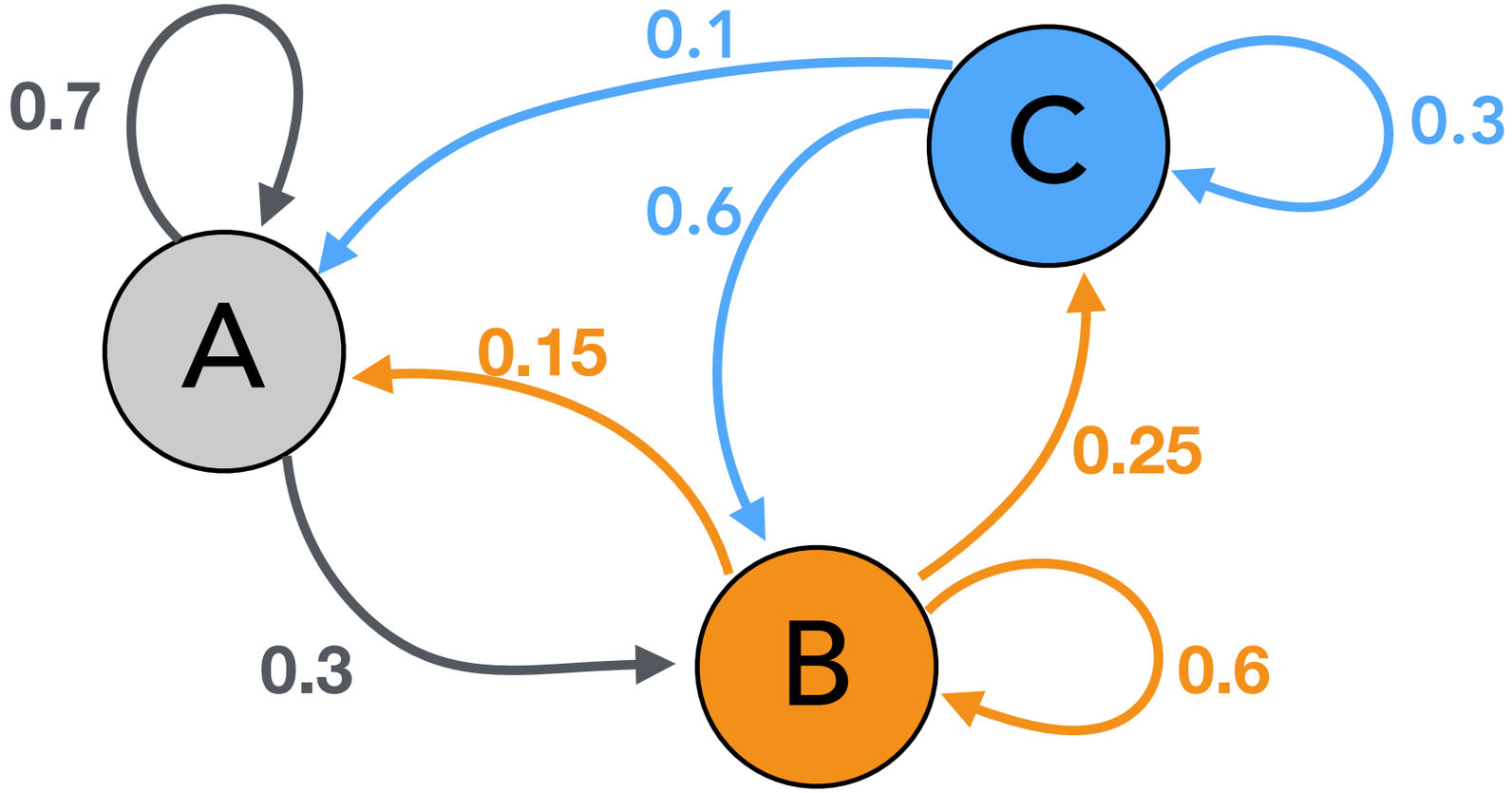
Markov Chain

- For 3 discrete states, there are transition probabilities to move to other states, as well as a probability to stay in the same state



Markov Chain

- Transition probabilities can also be shown in matrix notation
- Visually, it should be clear that the next transition is conditional only on the current state



$$P(A \rightarrow A) = 0.7$$
$$P(A \rightarrow B) = 0.3$$
$$P(A \rightarrow C) = 0.0$$

$$P(B \rightarrow A) = 0.15$$
$$P(B \rightarrow B) = 0.60$$
$$P(B \rightarrow C) = 0.25$$

$$P(C \rightarrow A) = 0.1$$
$$P(C \rightarrow B) = 0.3$$
$$P(C \rightarrow C) = 0.6$$

Example of Building a Chain

$$P = \begin{bmatrix} 0.9 & 0.075 & 0.025 \\ 0.15 & 0.8 & 0.05 \\ 0.25 & 0.25 & 0.5 \end{bmatrix}$$

at time n the system is in state 2 = bear, then 3 time periods later at time n+3 the distribution is...

$$x^{(n+3)} = x^{(n+2)}P = \left(x^{(n+1)}P\right)P$$

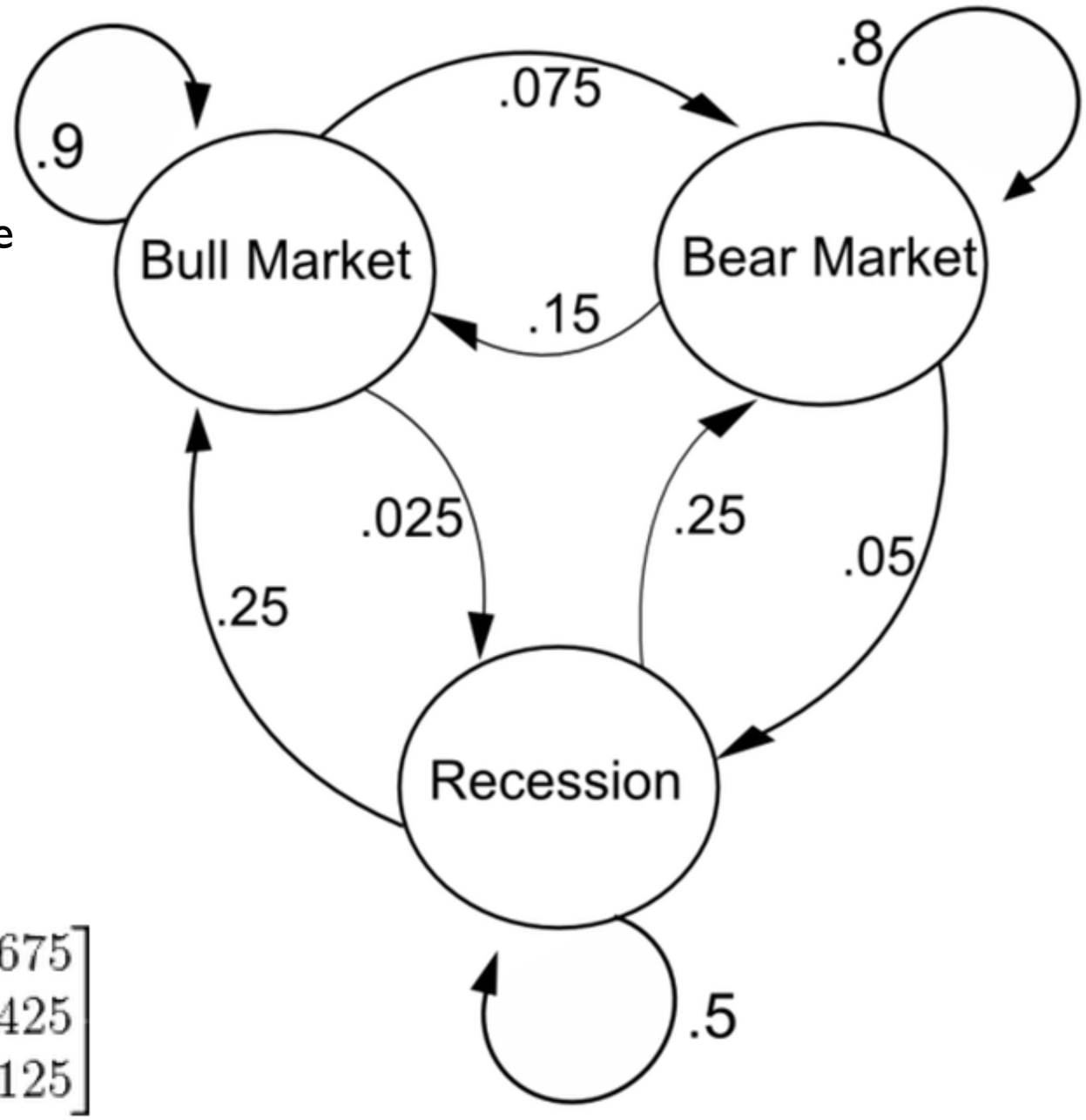
$$= x^{(n+1)}P^2 = \left(x^{(n)}P^2\right)P$$

$$= x^{(n)}P^3$$

$$= \begin{bmatrix} 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 0.9 & 0.075 & 0.025 \\ 0.15 & 0.8 & 0.05 \\ 0.25 & 0.25 & 0.5 \end{bmatrix}^3$$

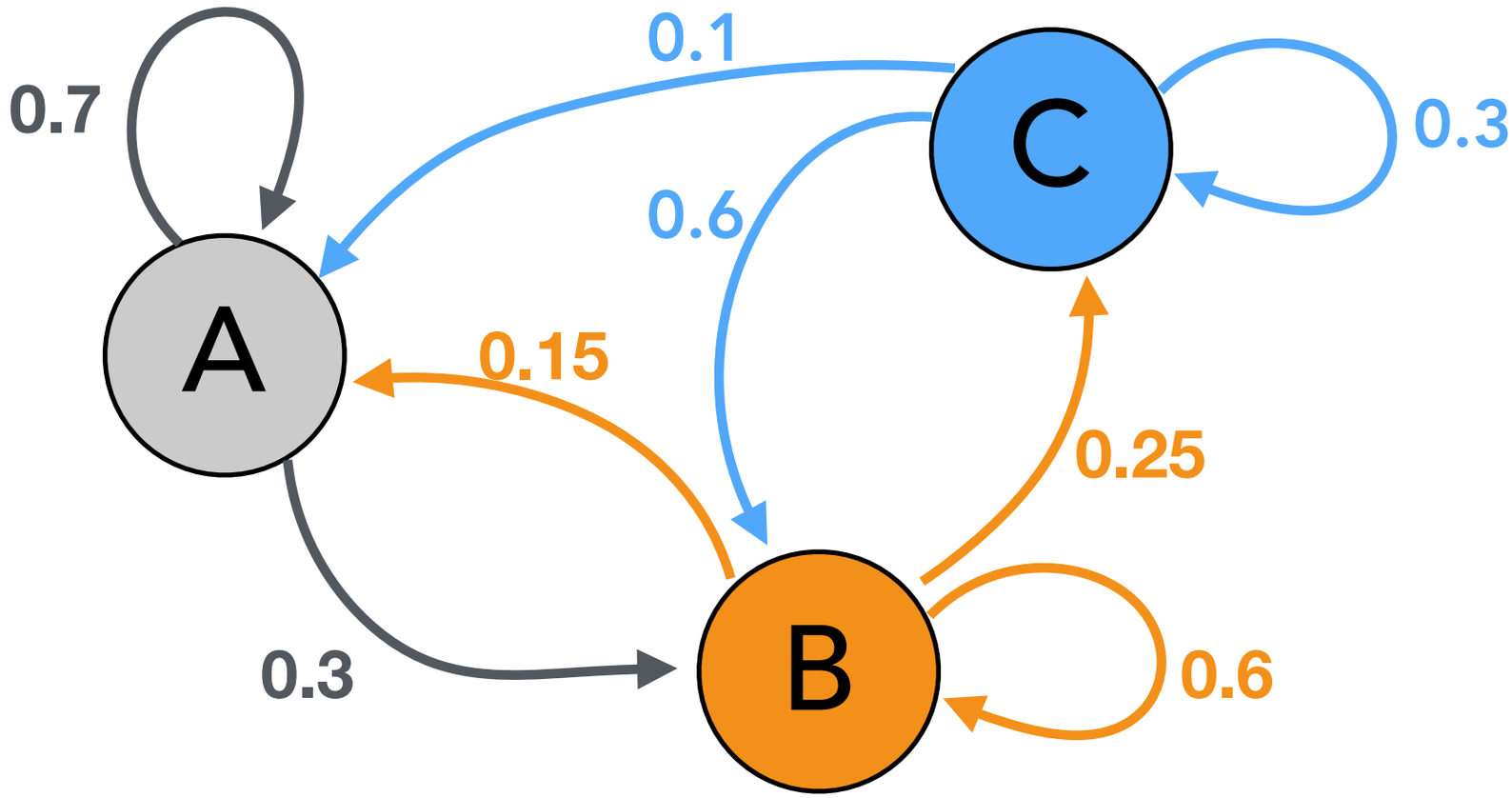
$$= \begin{bmatrix} 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 0.7745 & 0.17875 & 0.04675 \\ 0.3575 & 0.56825 & 0.07425 \\ 0.4675 & 0.37125 & 0.16125 \end{bmatrix}$$

$$= \begin{bmatrix} 0.3575 & 0.56825 & 0.07425 \end{bmatrix}.$$



Markov Chain

- Nicer visualization at http://setosa.io/ev/markov-chains/ for discrete state Markov chains
- It **is not** necessary to have discrete 'states' in order to use Markov chains. It **is** necessary to be able to have a transition probability between the current and potential next state



Exercise #2

- Let's make a simulation of a chain w/ two different starting points; one change starting at 100 and another at -27.
  - The step to X_{t+1} from X_t is governed by a random number drawn from a normalized gaussian PDF (scipy.stats.norm()) that is dependent on X_t
  - The PDF is of the form:

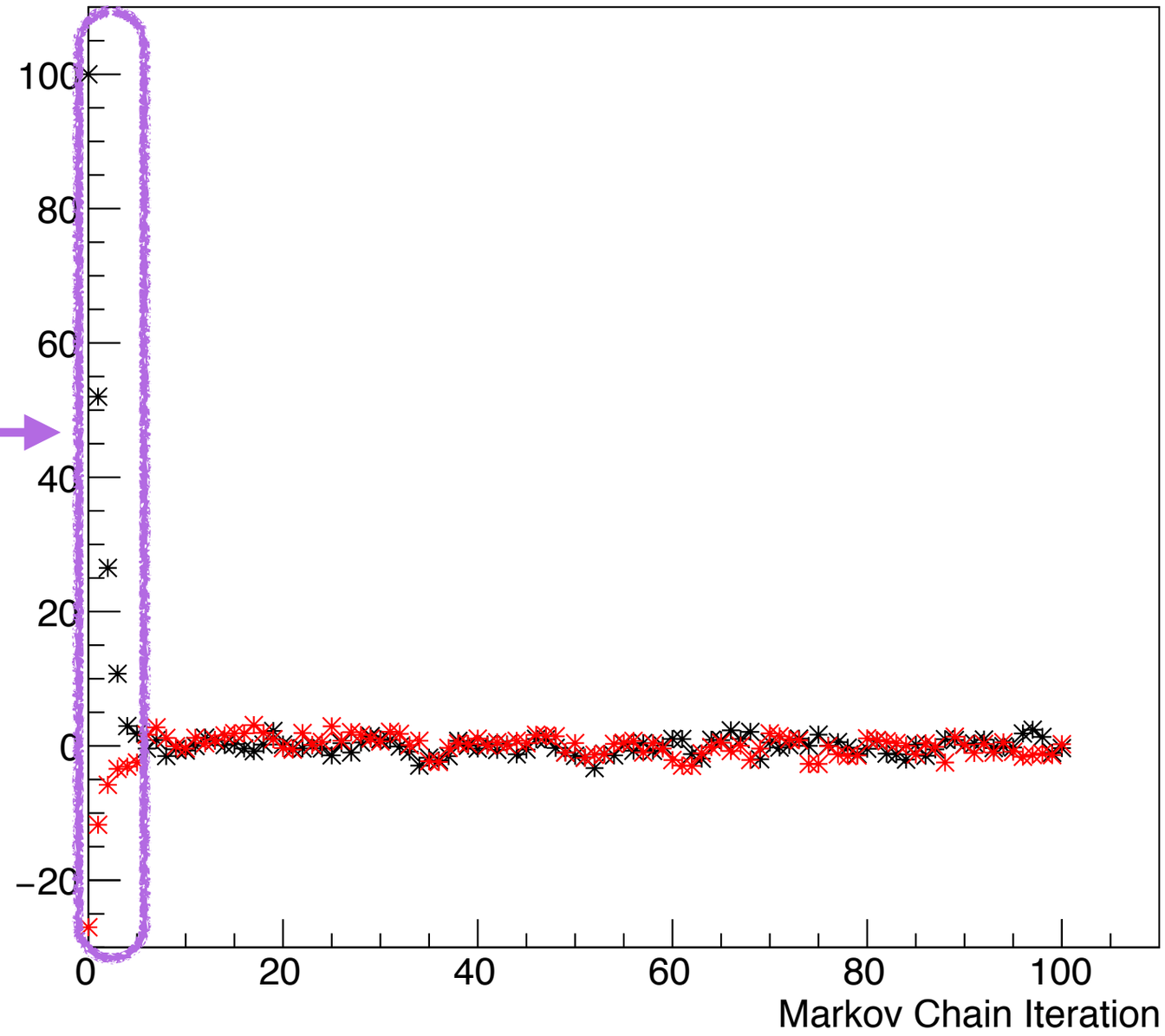
$$\frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-0.5X_t)^2}{2\sigma^2}} \qquad \sigma = 1$$

- Plot the values versus the iteration number. The point here is to see the Markov chain converge to a stationary distribution

Exercise #2 (plots)

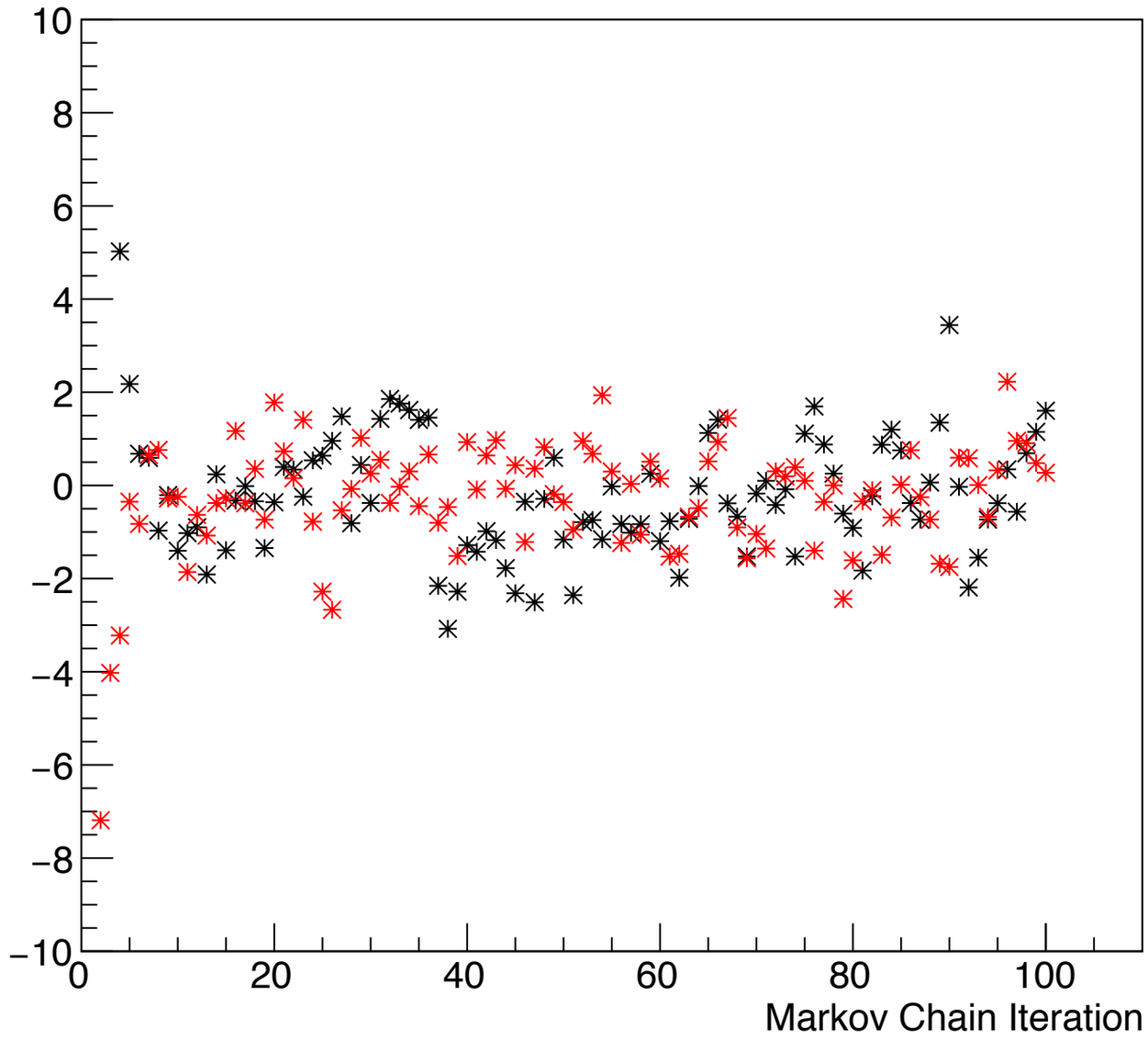
- After maybe 5-10 iterations from the starting point the chains look to converge to some stationary behavior

The samples before convergence are commonly known as ‘burn-in samples’ and are not often included when estimating the posterior distribution. They’re generally just discarded and understood as the cost of using Markov Chain Monte Carlos.



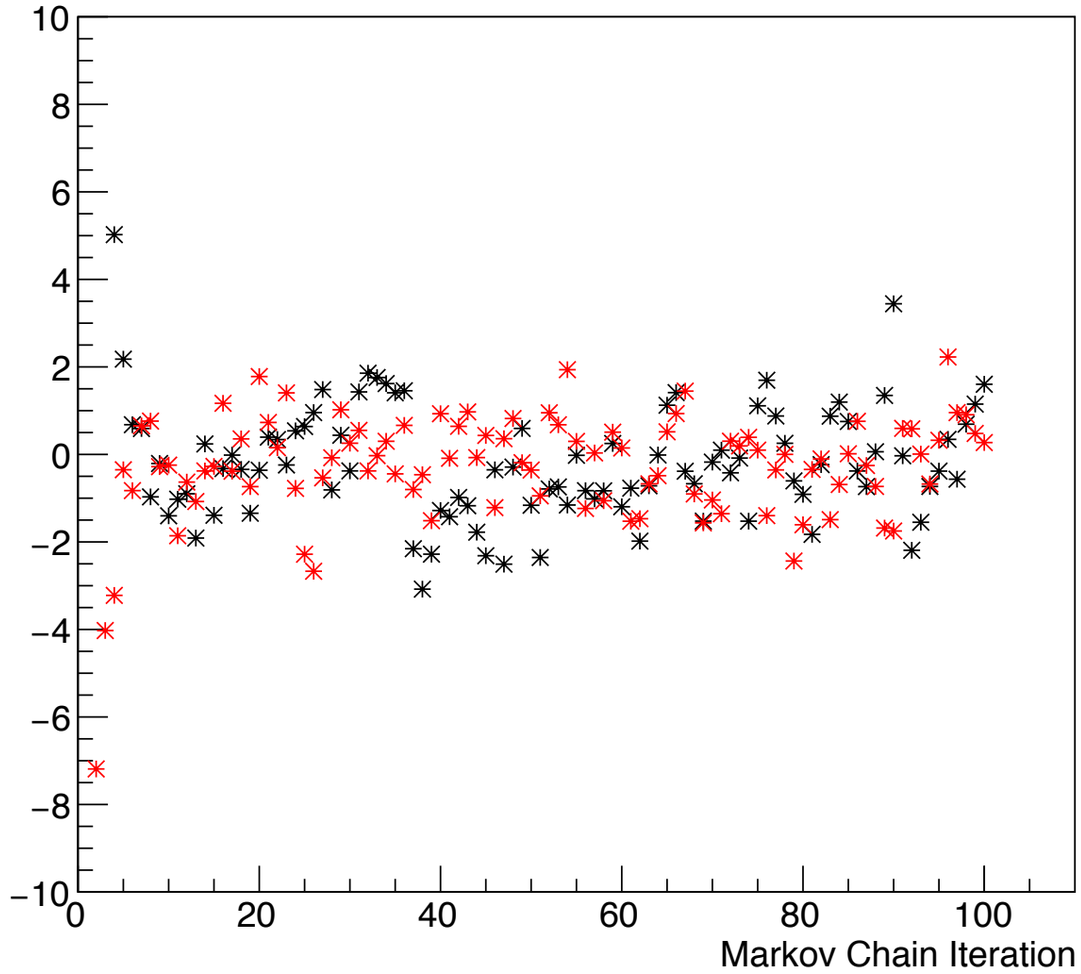
Exercise #2 (plots)

- When zoomed in we see that there is some stable distribution with some random scatter which the chain converges to irrespective of the starting point



Exercise #2 (extra)

- Assuming that the distribution that this chain converges to is a gaussian, use the samples after some number of iterations (~10) to estimate the mean and sigma

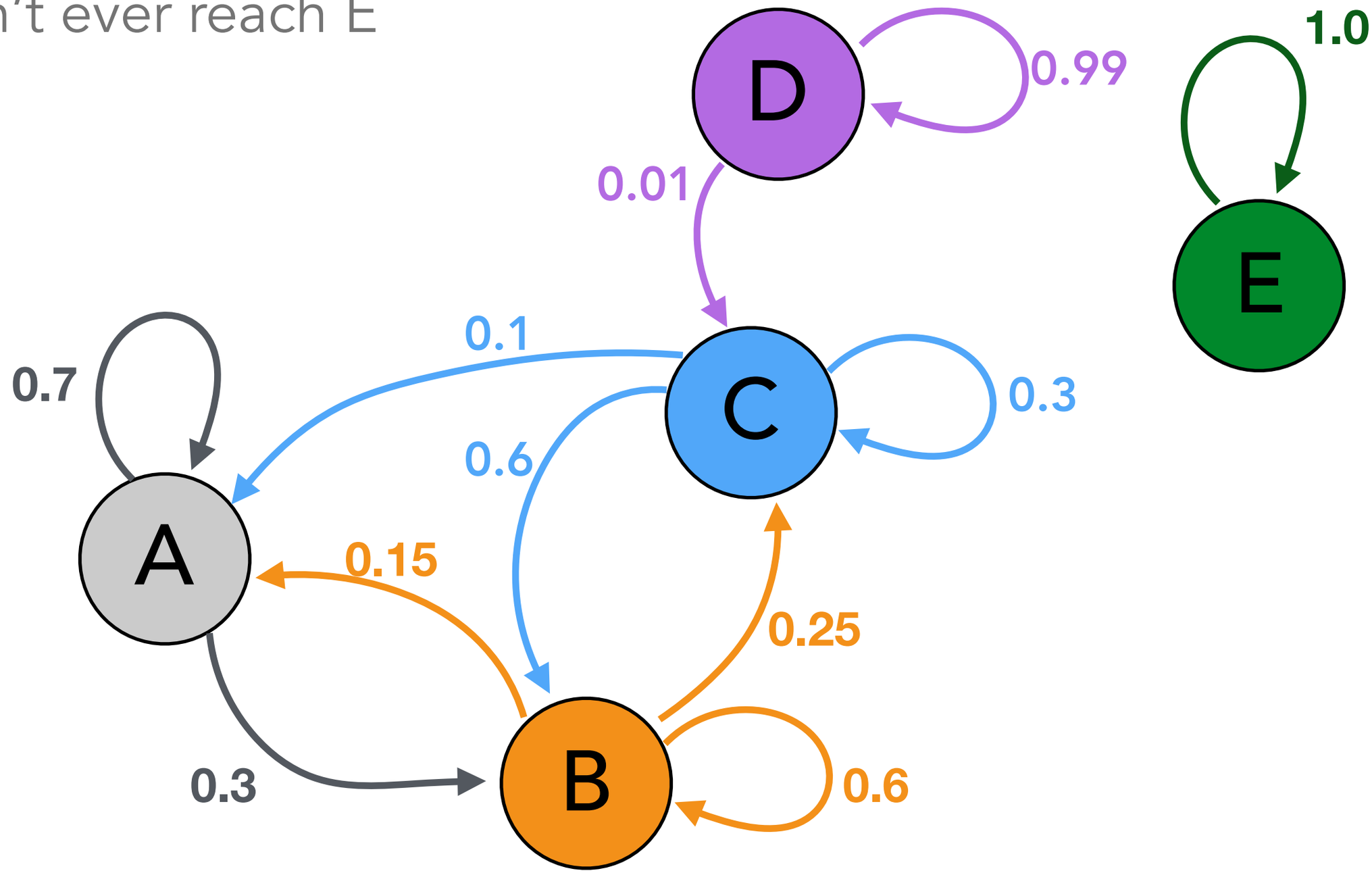


Markov Chain Feature

- The Markov chains have some very useful properties:
  - Irreducible: Assuming a stationary distribution exists, it is irreducible when any of the states can be reached from any other state in a finite number of transitions, e.g. all values are reachable no matter where you start. Non-zero probability of going from current state to any other state. The prob. can be be super-tiny, but just not zero.
  - Aperiodic: The chain does not return to the same state with predetermined transitions, i.e. no cycles.
  - Ergodic: Irreducible, aperiodic, and positive recurrent. Roughly speaking it should not be stuck or confined to any particular region of phase space, and can return to any previous state.
- Markov chains have a stationary distribution (often noted as $\pi(x)$)

Irreducibility

- Can’t get to state D w/o starting in state D
- Can’t ever reach E



Markov Chains for Bayes' Stuff

- So how does a Markov chain help with establishing Bayesian posterior distributions?
- Well, Markov chains will asymptotically approach a stable distribution, and we can give the Markov chain a distribution that is representative of the posterior. Remember that,

$$\text{posterior} \propto \text{prior} \times \text{likelihood}$$

- So using Markov Chain Monte Carlo, the chain can start at points that are not typical of the actual posterior (which we may not know well), but after enough Monte Carlo iterations it should converge to the posterior.
- Now we need to find transition schemes that cause the chain to converge to the stable or invariant posterior distribution.

Monte Carlo Markov Chains (MCMC)

- Thankfully, a host of Markov chain methods have already been produced, so it is unnecessary to derive our own for most problems
- While there are many types of Markov chains, there are at least two that you are likely to encounter
  - Metropolis-Hastings, which is actually a more general case than the original Metropolis algorithm
  - Gibbs sampling
- We will cover Metropolis-Hastings to get a feel for what is happening with the chains, but you are encouraged to use other Markov chain techniques too

Metropolis-Hastings

- We start with trying to establish what a ‘stationary’ distribution is, namely that going from state x to x′ is the same as going from x′ to x.

$$P(x)P(x \to x') = P(x')P(x' \to x)$$

- Which can be rewritten as

$$\frac{P(x \to x')}{P(x' \to x)} = \frac{P(x')}{P(x)}$$

- The probability of moving from x to x′ can be broken down into two sub-steps: picking a random value (x′) based on x according to some PDF (the proposal distribution) and then the accepting the move from x to x′ based on an acceptance distribution.

$$P(x \to x') = \underbrace{g(x \to x')}_{\text{proposal}} \underbrace{A(x \to x')}_{\text{acceptance}}$$

Metropolis-Hastings

- Thus,

$$\frac{P(x \to x')}{P(x' \to x)} = \frac{P(x')}{P(x)}$$

- becomes,

$$\frac{A(x \to x')}{A(x' \to x)} = \frac{P(x')g(x' \to x)}{P(x)g(x \to x')}$$

$$= \frac{P(x')g(x|x')}{P(x)g(x'|x)}$$

- I could have used x_t instead of just x, but x′ is **not** x_{t+1}. x′ is the **candidate** for transition
- P(x) is often written as π(x) to reflect that it is the stationary distribution

Metropolis-Hastings

- Let's take a moment to think about the acceptance ratio

$$r = \frac{A(x \to x')}{A(x' \to x)} = \frac{P(x')g(x|x')}{P(x)g(x'|x)}$$

- If r ≥ 1, it means that the move from x to x′ is a transition to a more/equally likely state and that it should be accepted, i.e. x_{t+1}=x′
- If r < 1, there is a non-zero probability that the transition is made, i.e. x_{t+1}=x', but only in proportion to the probability r. Otherwise, the transition is rejected and we re-sample at the current point, i.e. x_{t+1}=x.
  - This way the Markov chain does not get stuck in a local area because it always has some probability of a transition to a less probably state or can make a large (but unlikely) 'jump' that leads to a more probable state
  - When the Markov chain converges to the stationary distribution it will effectively start to sample the posterior distribution, which is exactly what we want

Quick Note

- Metropolis-Hastings corrects for a bias which could be introduced from having an asymmetric proposal PDF, e.g. g(xlx')
  - We might want an asymmetric proposal PDF because we will knowingly *start* the Markov chain always above/below the maximum a posteriori (MAP). Thus, the Markov chain is likely to converge quicker if the transitions in either the +/- direction are larger versus the opposite direction.
- But, when the Markov chain does converge, we want it sample the posterior distribution properly. So we need to include the probability that transitions from proposal function will not be symmetric, i.e. the probability of g(xlx') ≠ g(x'lx).

Proposal Function

- Having a proposal function (PDF) which is narrow can translate into many iterations before converging, which is inefficient
- A proposal PDF which is large can cause the Markov chain to ‘jump’ over the desired posterior distribution area, and converge in some other region
- Tuning MCMCs is often necessary to balance these two competing issues, and rerunning the MCMC multiple times helps to establish that you have actually found the posterior distribution
- Just like minimizing for negative likelihoods there are no concrete rules for establishing if the MCMC has found the invariant global posterior distribution

Metropolis-Hastings Walkthrough

- Depending on r, we have the transition

$$x_{t+1} = \begin{cases} x', r \geq 1 \\ x', \text{with probability } r \text{ if } r < 1 \\ x_t, \text{with probability } 1 - r \text{ if } r < 1 \end{cases}$$

- So with r calculated, we can get a random number from a uniform random number generator that samples from 0-1 (u) and see if the the transition to x’ is accepted or rejected, i.e. if r > u the transition to x’ is accepted.
- *Voila*, Metropolis-Hastings

Exercise #3

We're changing notation slightly:

$x' = \theta_p$

$x_t = \theta$

- In exercise #1 we scanned across the probability space (θ) to get the posterior distribution and now we're going use a Markov Chain Monte Carlo with the Metropolis-Hastings algorithm… by hand (no external packages, yet)
  - MCMC is complete overkill for a problem of this nature, but we're going to do it anyway to understand what is going on
  - Similar to exercise #1 the prior PDF is a beta-distribution with α=5 and β=17
  - The likelihood will be the same as before, a binomial PDF w/ n=100 and k=61 (heads)
  - The proposal function is θ_p=θ+PDF, where the proposal PDF is a normalized gaussian centered at μ=0 with a σ=0.3 (scipy.stats.norm.rvs(0, 0.3))
- Because you should already have the likelihood and prior coded, now we just have to add the MH algorithm

Exercise #3 (cont.)

- It might seem difficult, but break it down into the component steps

$$r = \frac{P(x')g(x|x')}{P(x)g(x'|x)}$$

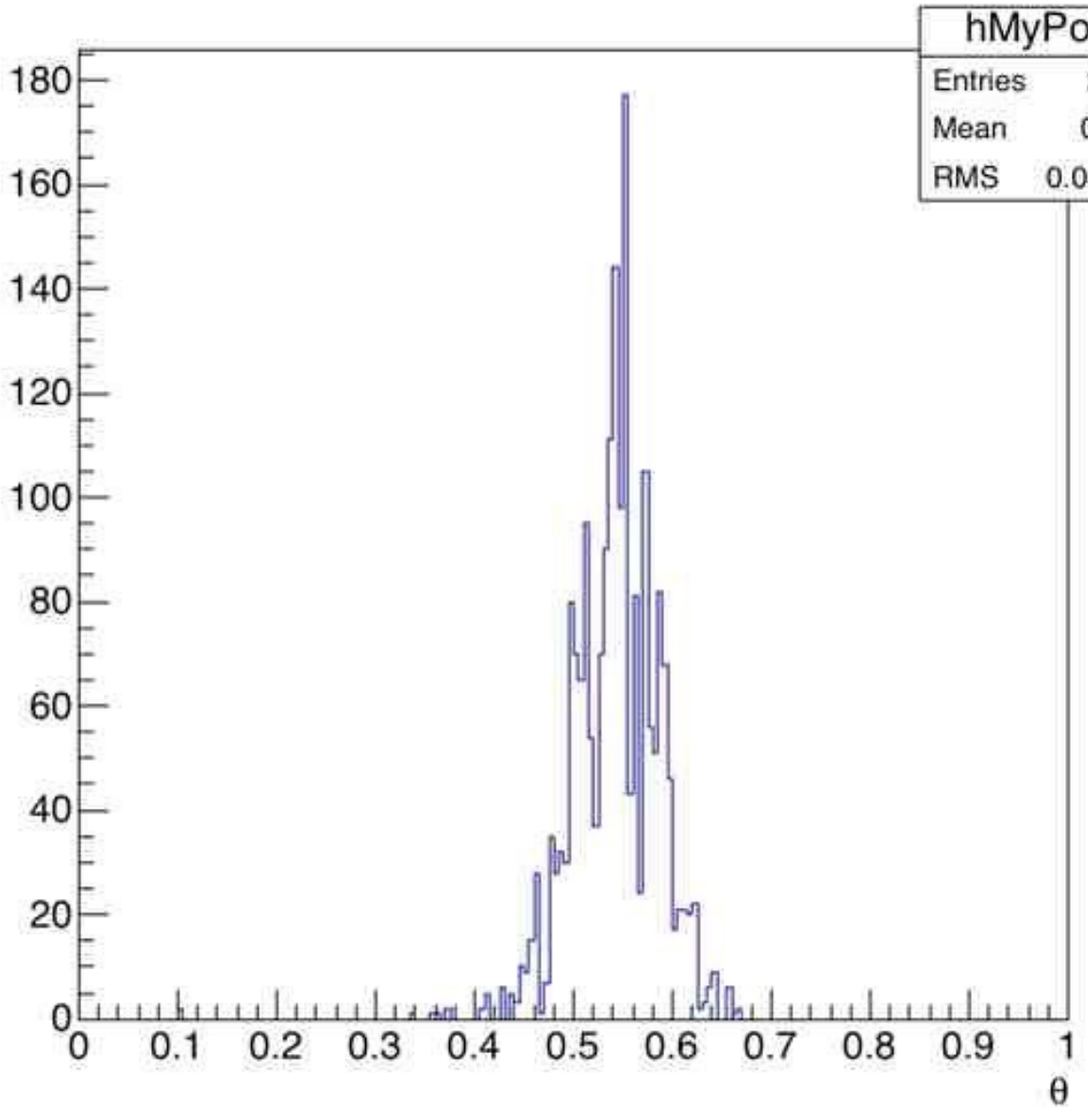
$$P(x') = posterior(x') \propto prior(x') * likelihood(x')$$

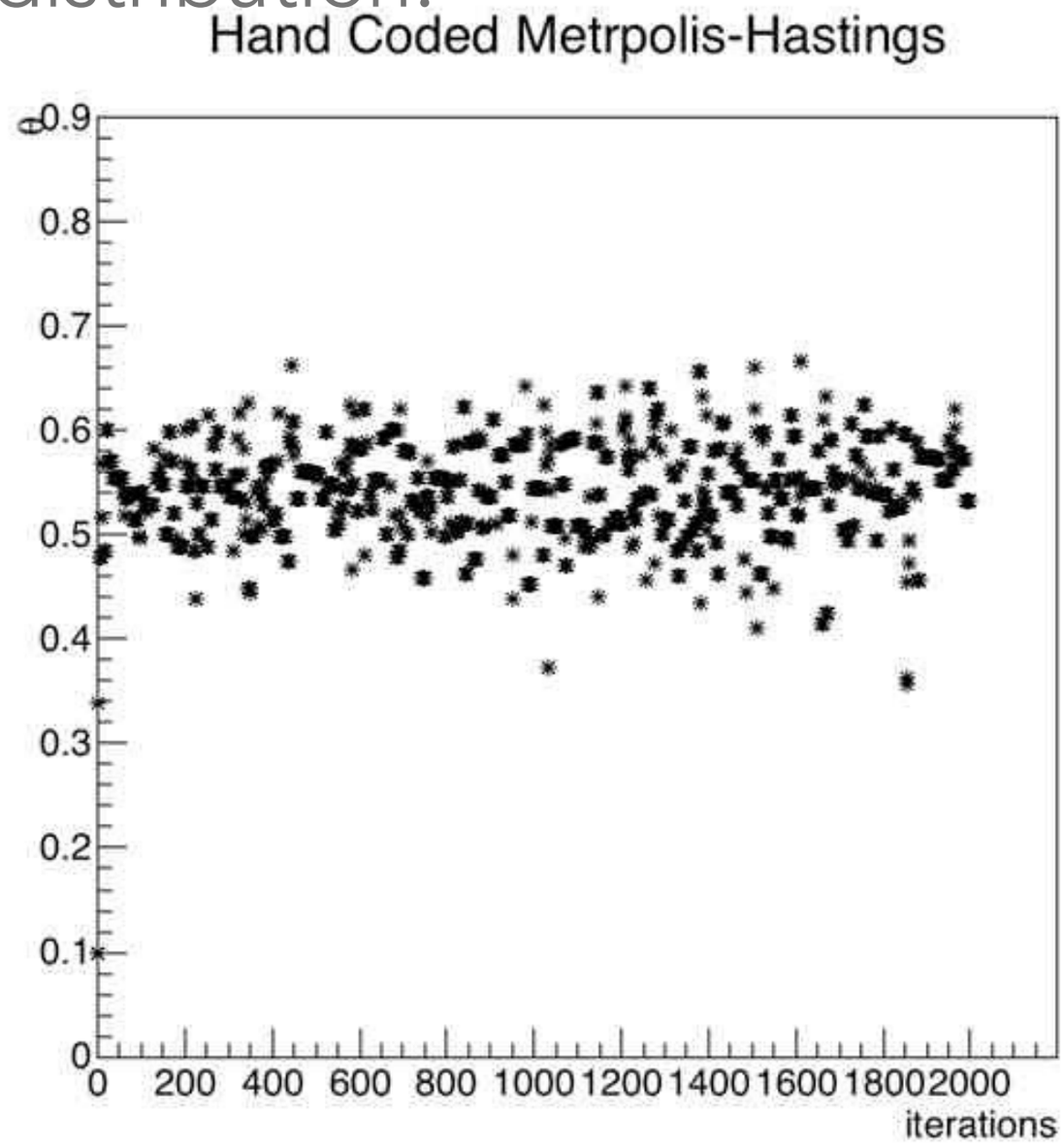
- A useful trait here is that the proposal function PDF is symmetric so

$$\frac{g(x|x')}{g(x'|x)} = 1$$

Exercise #3 (cont.)

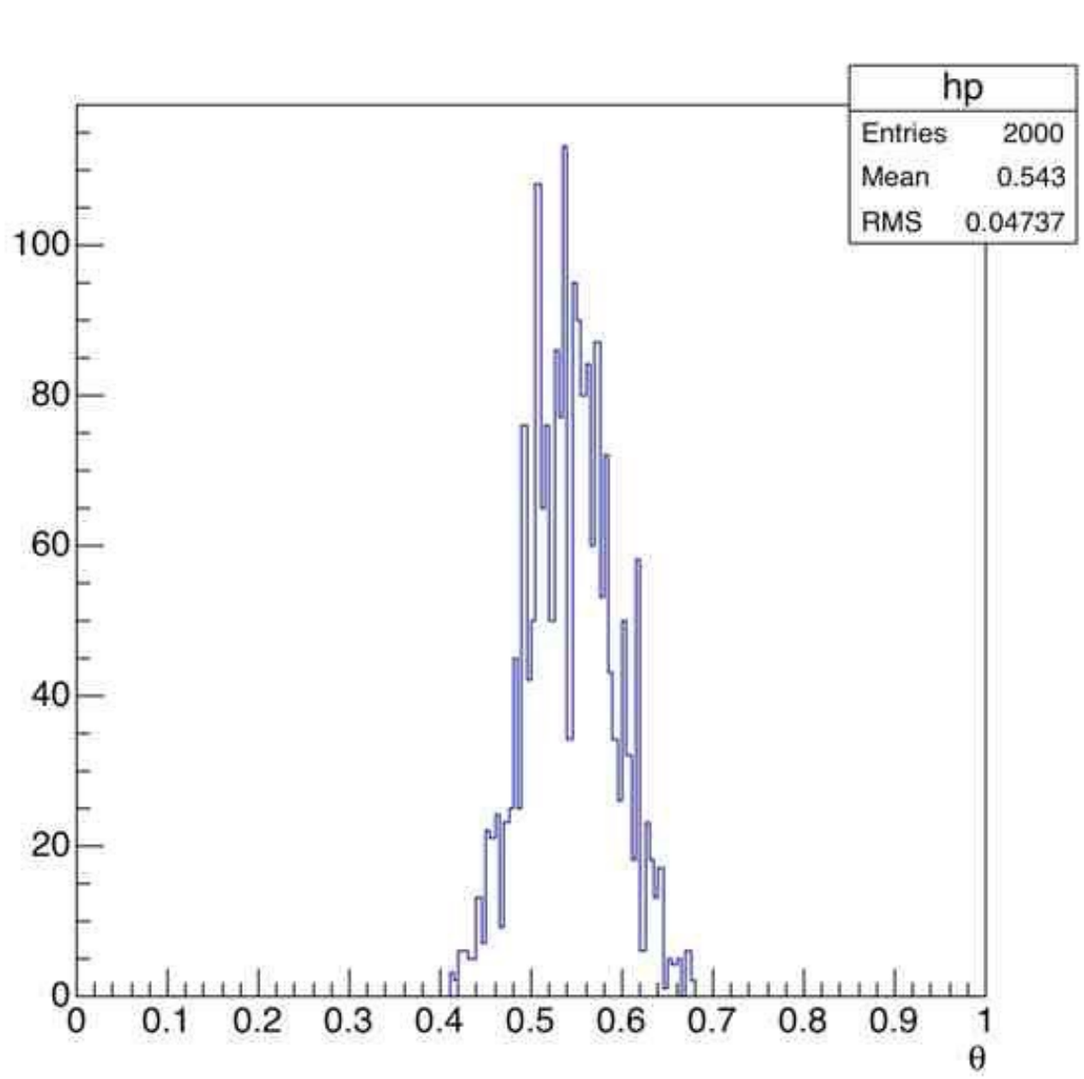
- For 2000 iterations plot Markov Chain Monte Carlo samples as a function of iteration, as well as a histogram of the samples, i.e. the posterior distribution.

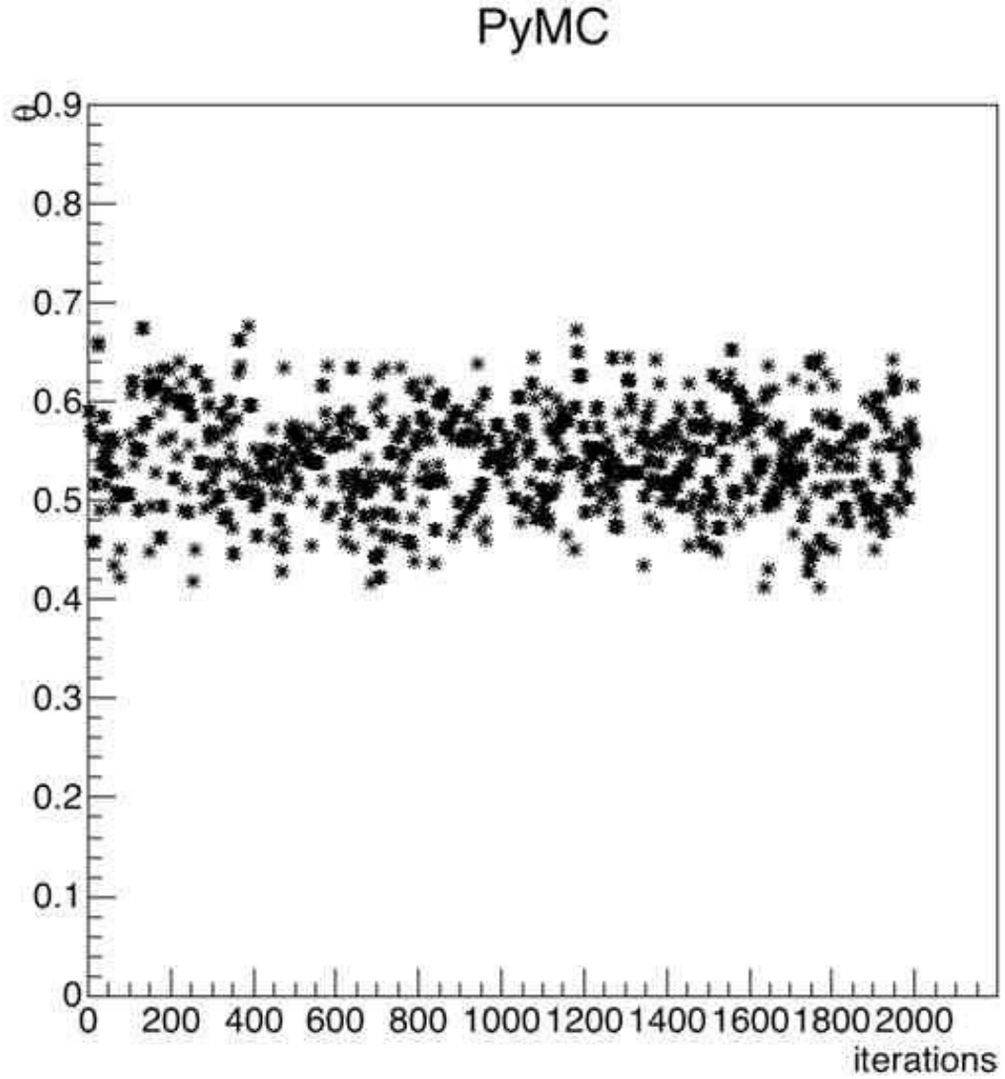




Exercise #4

- Using an external MCMC package, redo exercise 3.
  - You don't have to use Metropolis-Hastings
- Do you get similar results?





Additional

- Switch to a non-symmetric proposal PDF, and see if your personally coded Metropolis-Hastings matches the results from the external package MCMC.