Markov Chain Monte Carlo & Sampling Methods

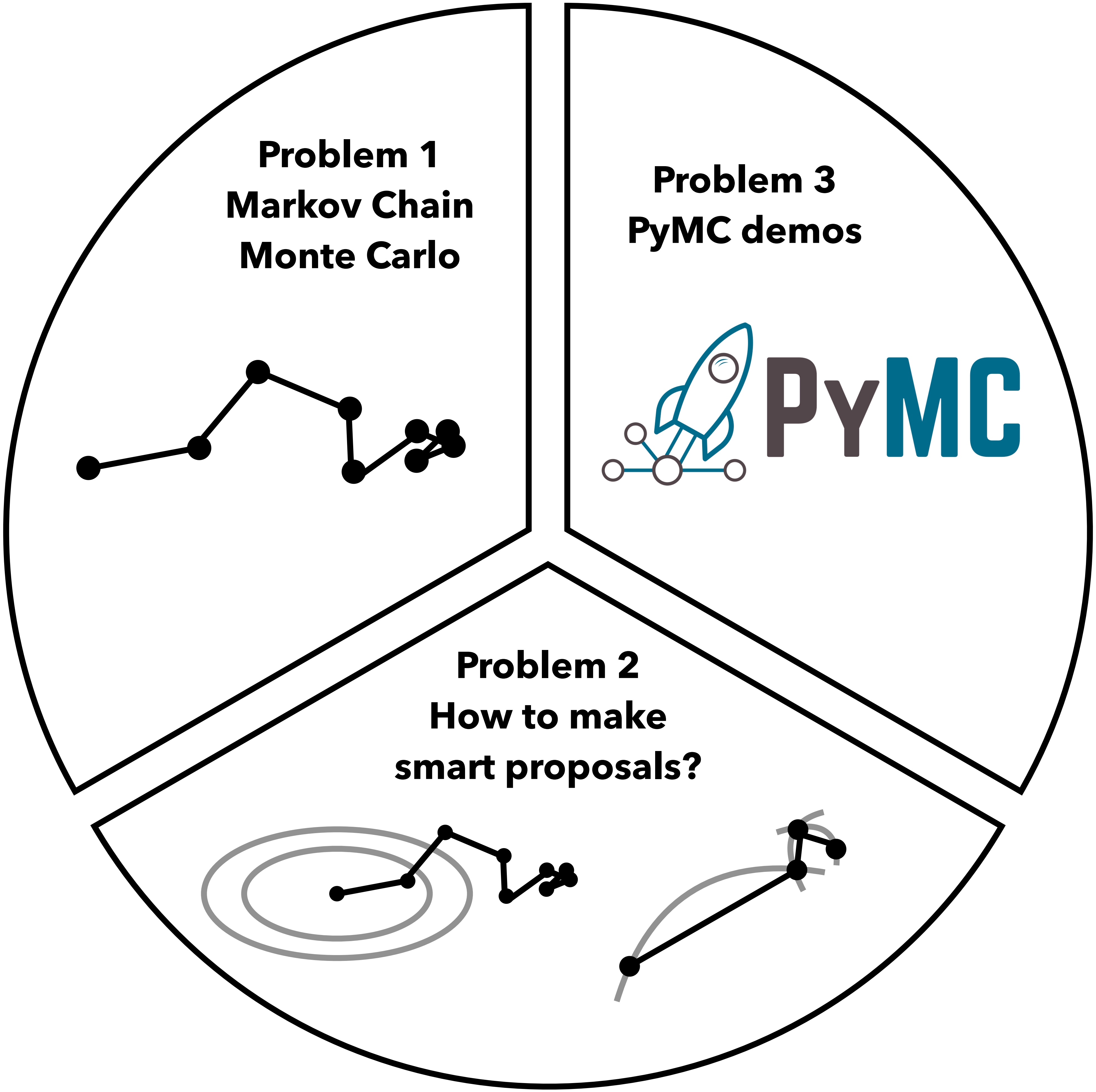
LSSTC Fellowship Program Session 16

Jiayin Dong, Flatiron Research Fellow
Center for Computational Astrophysics, Flatiron Institute
9/20/2022

In the last lecture, we learned to compute the posterior using "grid approximation".

The technique is limited to problems with a small number of random variables.

In this lecture, we will introduce another approach to approximate posteriors, the Markov Chain Monte Carlo.

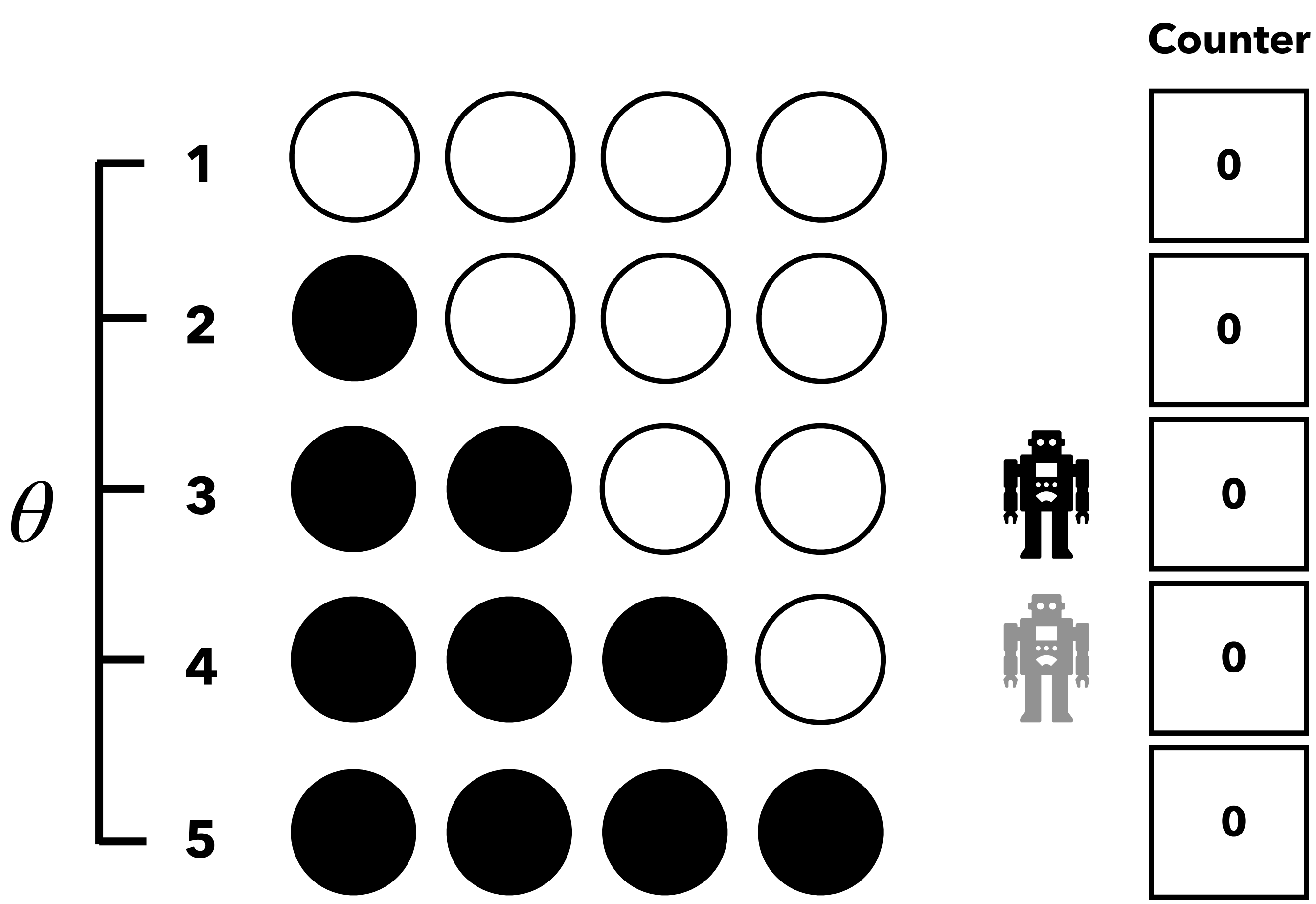

Problem 1
Markov Chain
Monte Carlo
Problem 3
PyMC demos
PyMC
Problem 2
How to make
smart proposals?

We have a bag containing 4 balls.

Each ball has two possible colors: black and white.

We draw 1 ball from the bag and it's black.

Q: What is the posterior of number of black balls?



Begin with a random state i.

Calculate $P(\theta_{\text{cur}} \,|\, D) = p(D \,|\, \theta_{\text{cur}}) p(\theta_{\text{cur}})$.

Flip a coin.

- **If head, propose to move to i+1**
- **If tail, propose to visit i-1**

Calculate $P(\theta_{\text{prop}} \,|\, D) = p(D \,|\, \theta_{\text{prop}}) p(\theta_{\text{prop}})$.

If $P(\theta_{\text{prop}} \,|\, D) > P(\theta_{\text{cur}} \,|\, D)$,

accept the proposal.

Else,

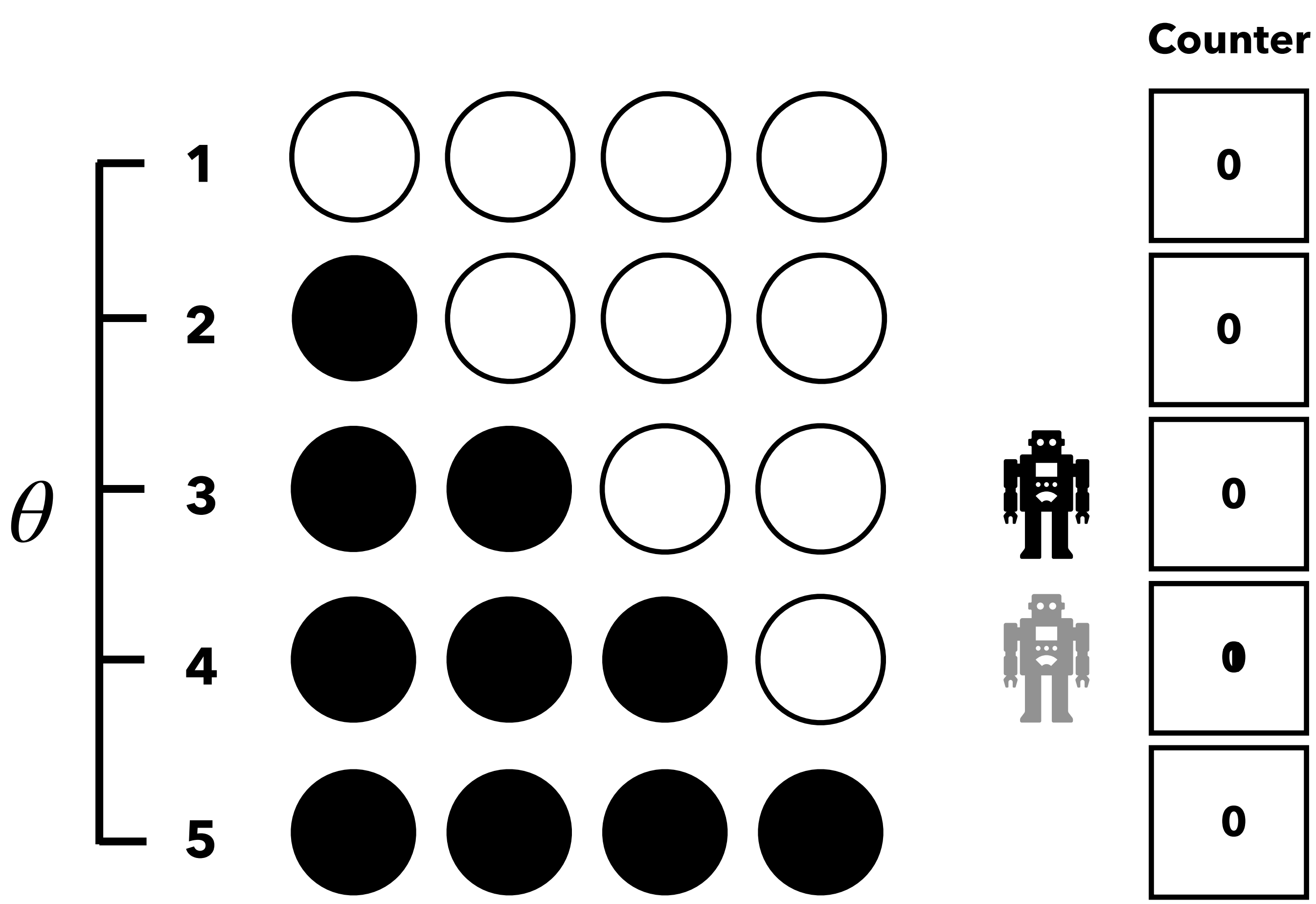
accept the proposal with probability of $P(\theta_{\text{prop}} \,|\, D) / P(\theta_{\text{cur}} \,|\, D)$.

We have a bag containing 4 balls.

Each ball has two possible colors: black and white.

We draw 1 ball from the bag and it's black.

Q: What is the posterior of number of black balls?



Begin with a random state i.
Calculate $P(\theta_{\mathrm{cur}} \,|\, D) = p(D \,|\, \theta_{\mathrm{cur}}) p(\theta_{\mathrm{cur}})$**.**

Flip a coin.

- **If head, propose to move to i+1**
- **If tail, propose to visit i-1**

Calculate $P(\theta_{\mathrm{prop}} \,|\, D) = p(D \,|\, \theta_{\mathrm{prop}}) p(\theta_{\mathrm{prop}})$**.**

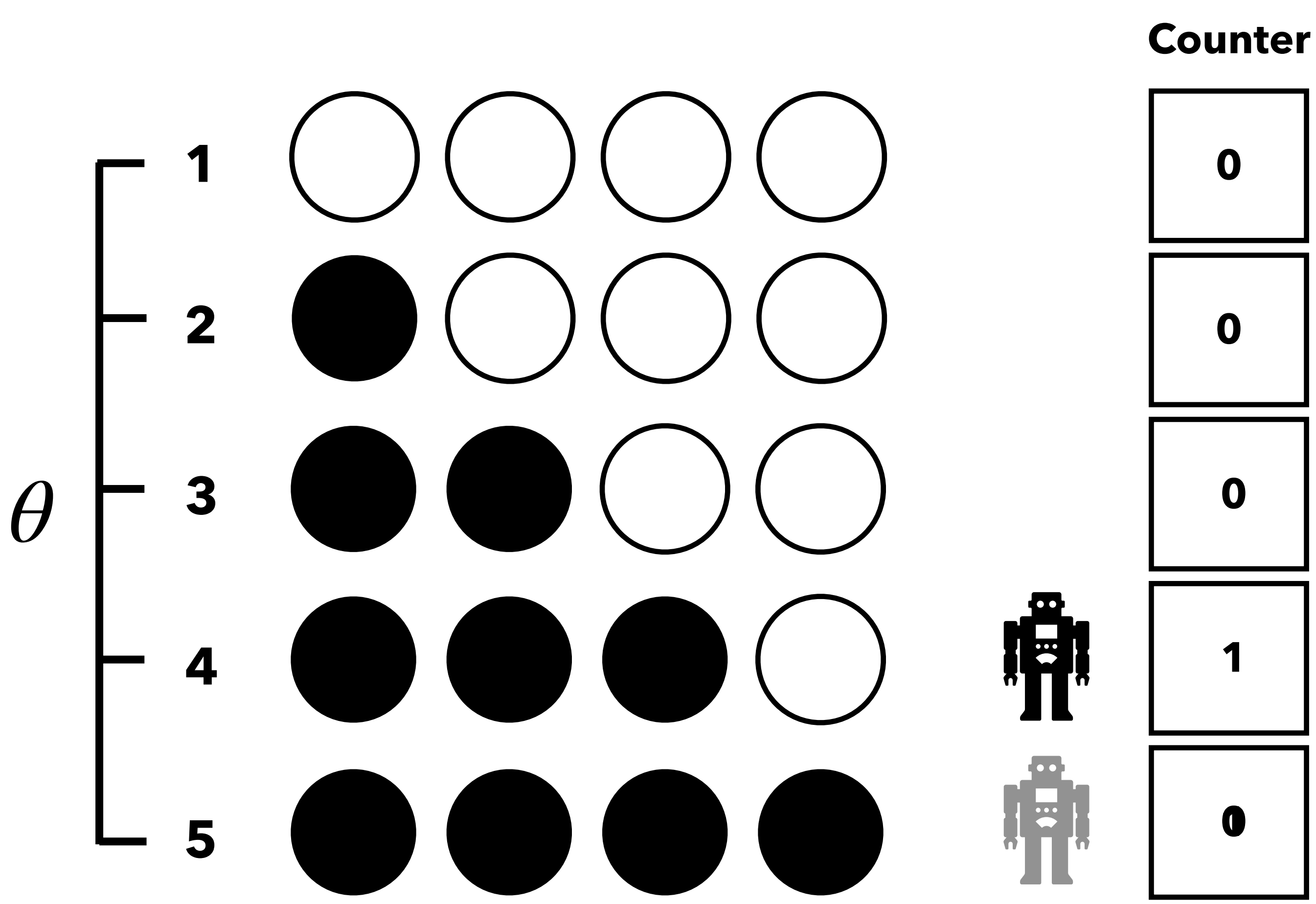
If $P(\theta_{\mathrm{prop}} \,|\, D) > P(\theta_{\mathrm{cur}} \,|\, D)$**,**
accept the proposal.
Else,
accept the proposal with probability of $P(\theta_{\mathrm{prop}} \,|\, D) / P(\theta_{\mathrm{cur}} \,|\, D)$**.**

We have a bag containing 4 balls.

Each ball has two possible colors: black and white.

We draw 1 ball from the bag and it's black.

Q: What is the posterior of number of black balls?



Begin with a random state i.

Calculate $P(\theta_{\text{cur}} \mid D) = p(D \mid \theta_{\text{cur}}) p(\theta_{\text{cur}})$.

Flip a coin.

- **If head, propose to move to i+1**
- **If tail, propose to visit i-1**

Calculate $P(\theta_{\text{prop}} \mid D) = p(D \mid \theta_{\text{prop}}) p(\theta_{\text{prop}})$.

If $P(\theta_{\text{prop}} \mid D) > P(\theta_{\text{cur}} \mid D)$,

accept the proposal.

Else,

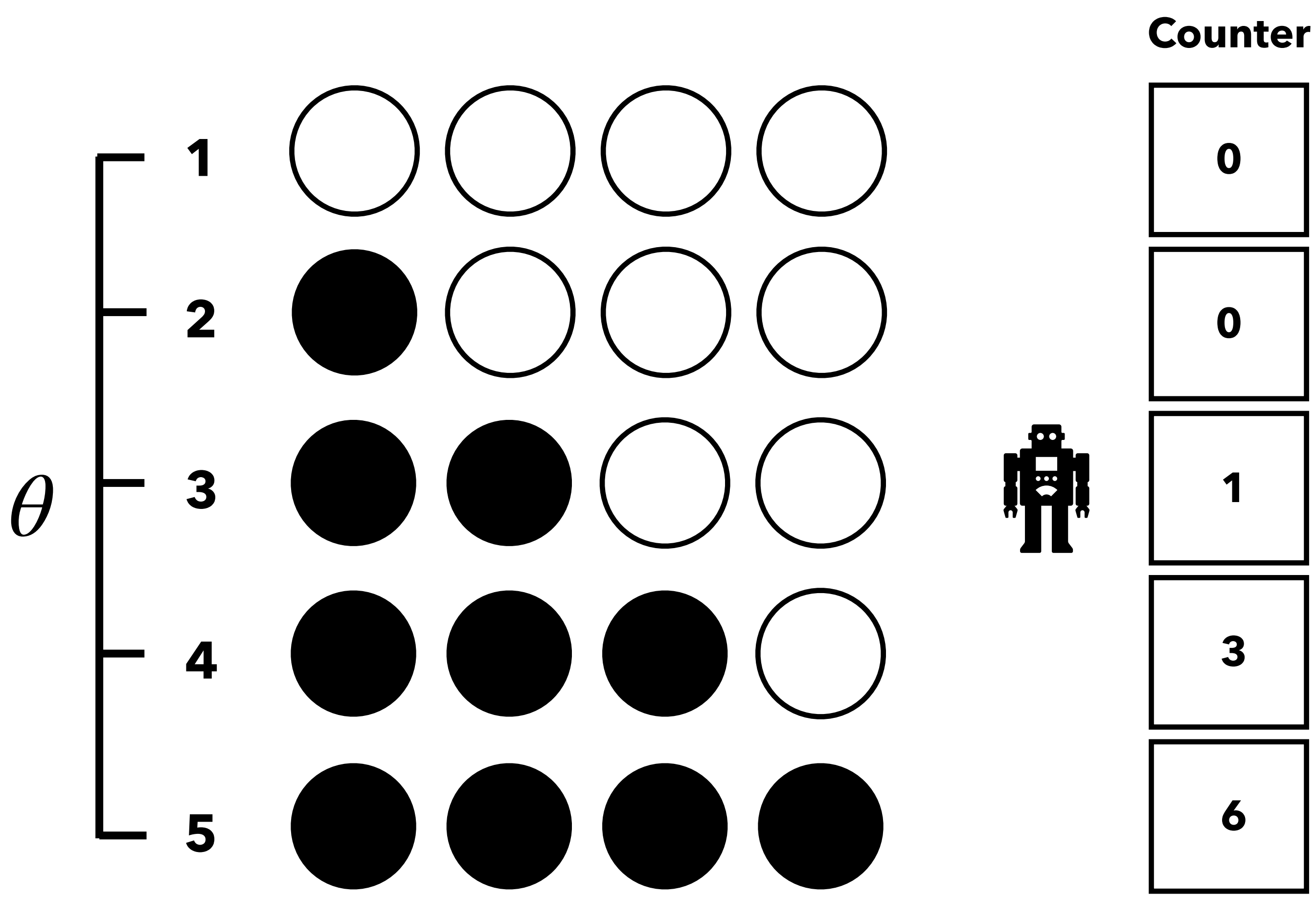
accept the proposal with probability of $P(\theta_{\text{prop}} \mid D) / P(\theta_{\text{cur}} \mid D)$.

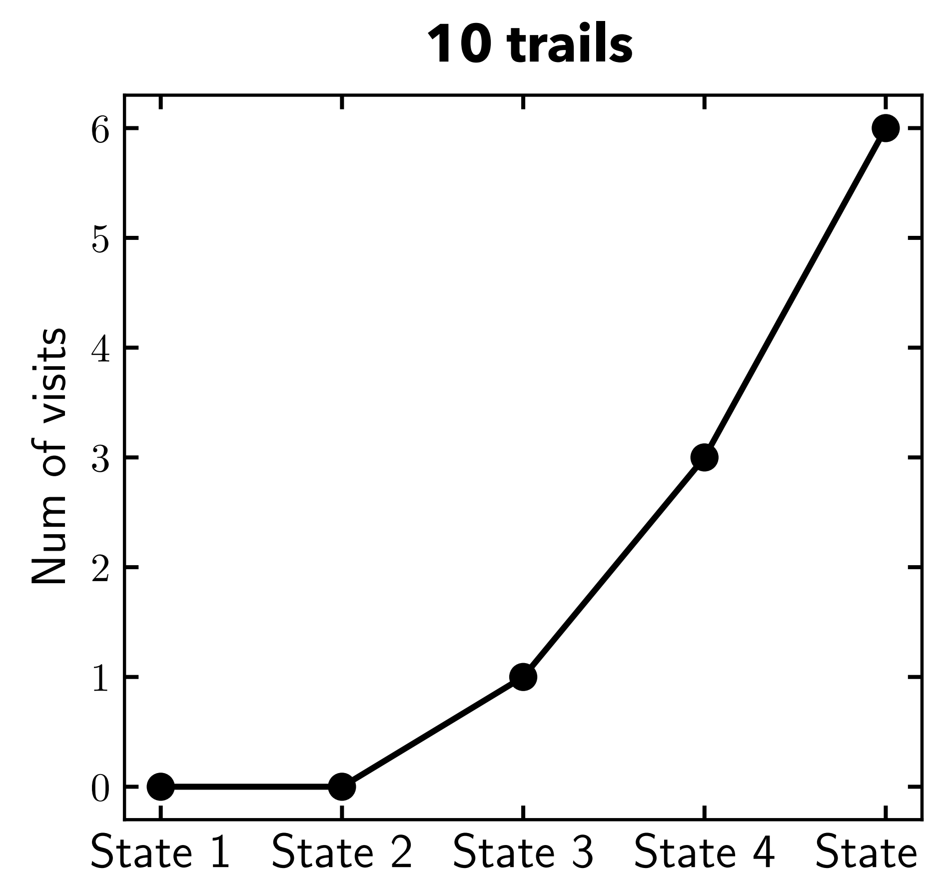
We have a bag containing 4 balls.

Each ball has two possible colors: black and white.

We draw 1 ball from the bag and it's black.

Q: What is the posterior of number of black balls?



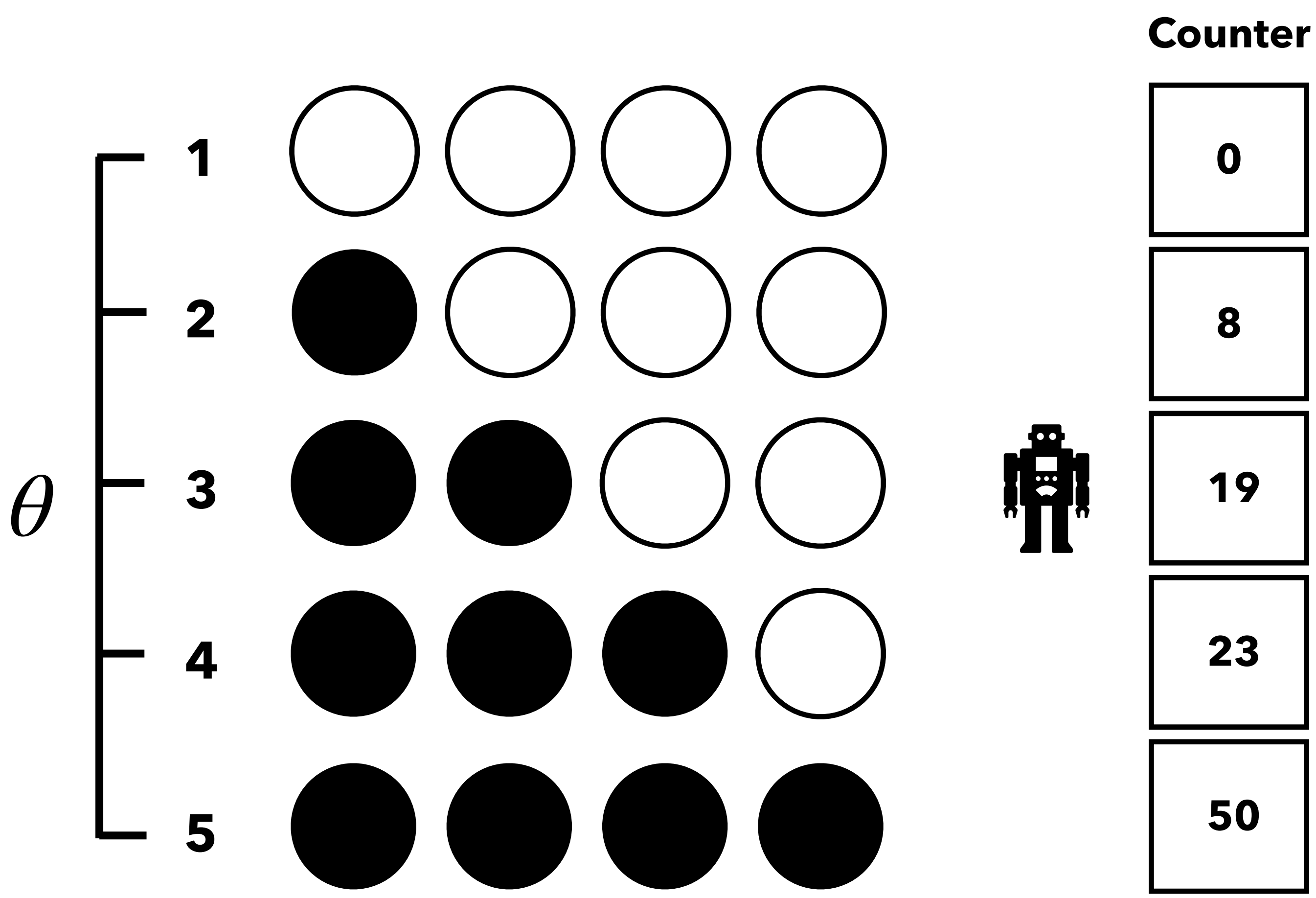


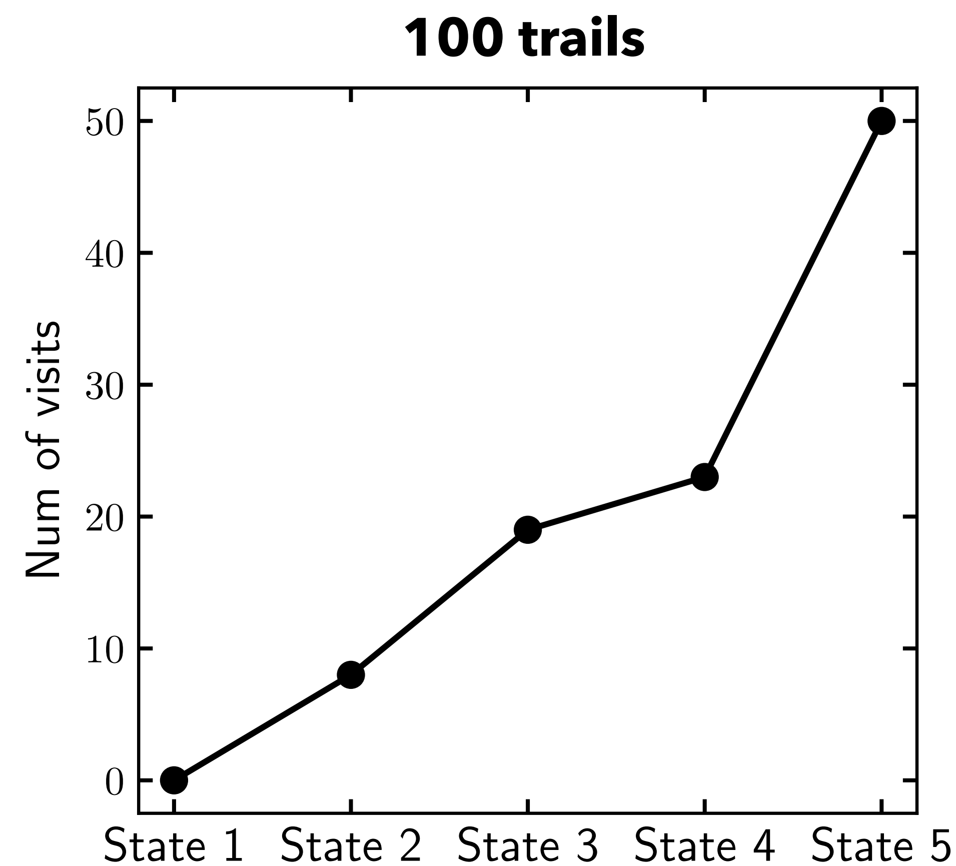
We have a bag containing 4 balls.

Each ball has two possible colors: black and white.

We draw 1 ball from the bag and it's black.

Q: What is the posterior of number of black balls?



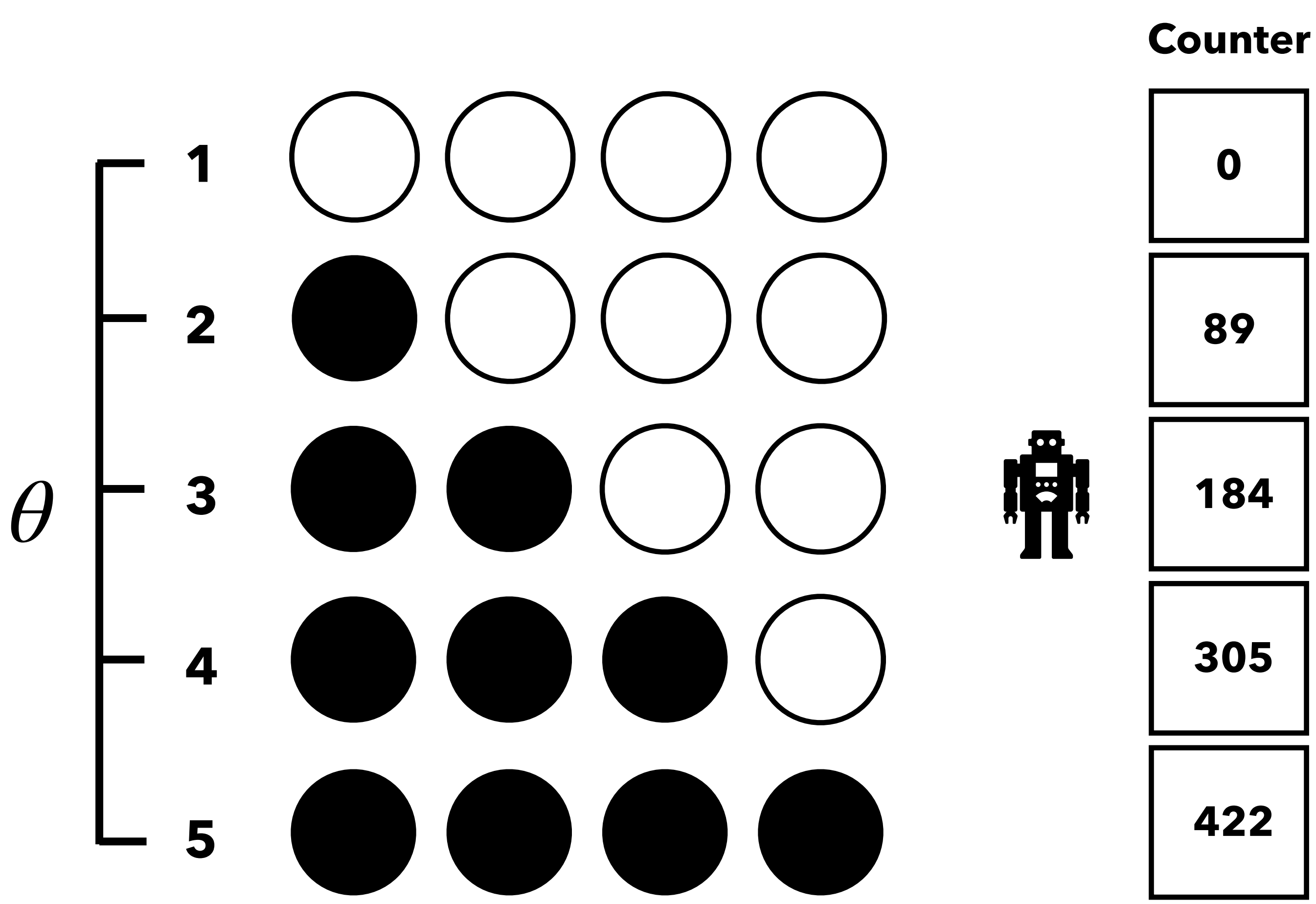


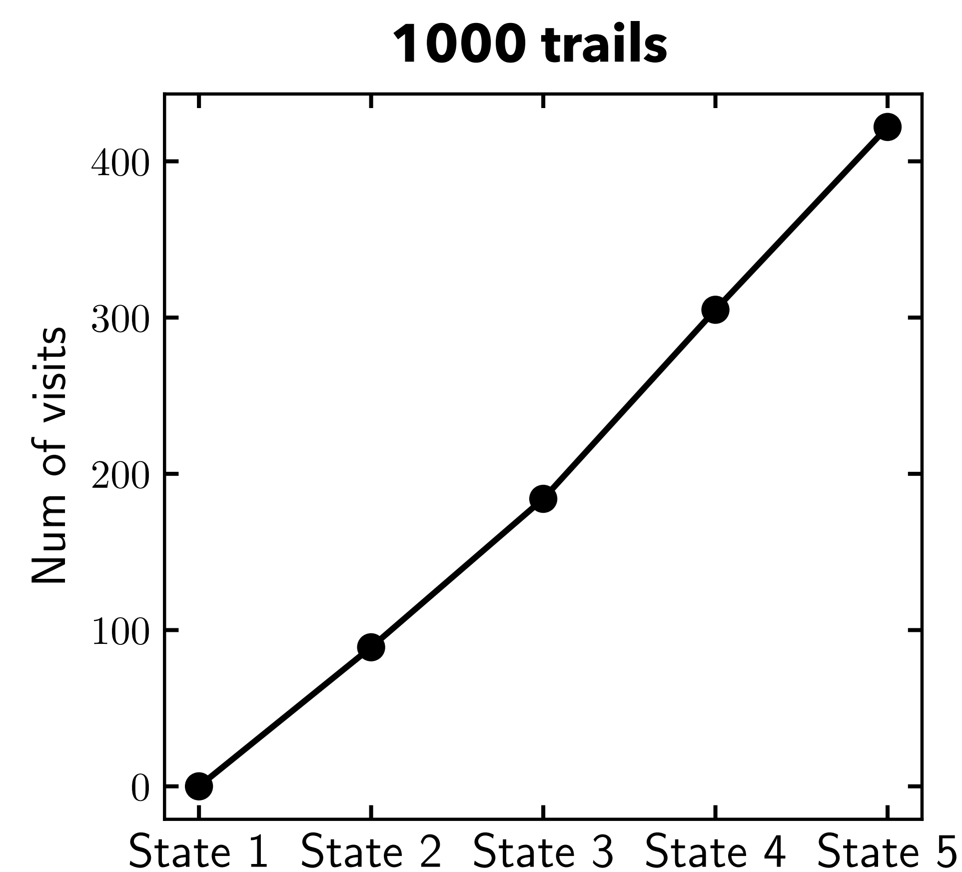
We have a bag containing 4 balls.

Each ball has two possible colors: black and white.

We draw 1 ball from the bag and it's black.

Q: What is the posterior of number of black balls?





Markov Chain Monte Carlo (MCMC)

- **The process of random sampling to approximate the posterior is a "Monte Carlo" process.**

- **Sampling a proposal only based on the current state is a "Markov Chain".**

- **<u>Philosophy of MCMC</u>: We want the number of visits to each state proportional to the posterior density.**

Metropolis-Hastings Algorithm

Metropolis Hastings

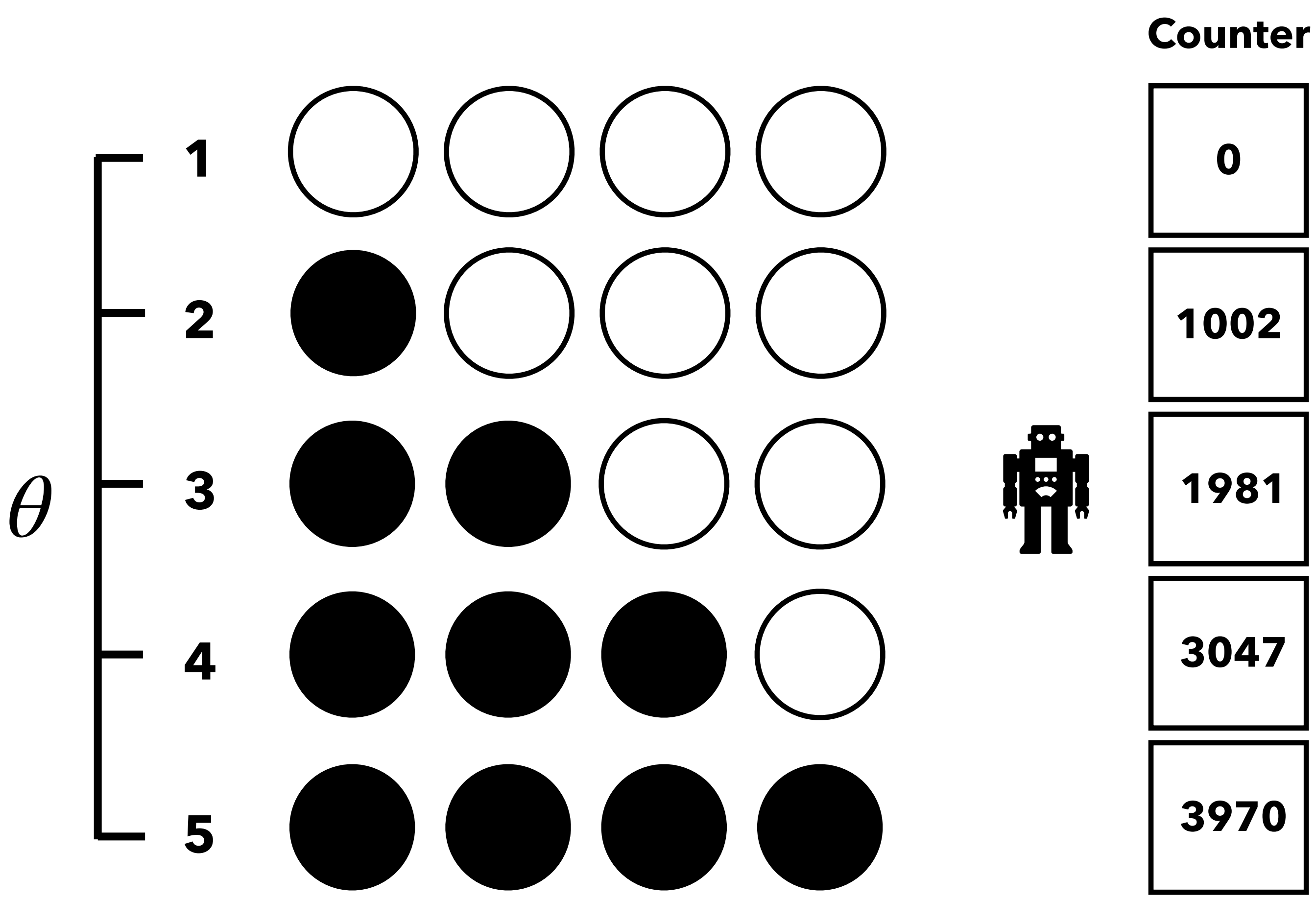
$$\alpha = \boxed{\frac{P(\theta_{\text{prop}} \mid D)}{P(\theta_{\text{cur}} \mid D)}} \times \boxed{\frac{q(\theta_{\text{cur}} \mid \theta_{\text{prop}})}{q(\theta_{\text{prop}} \mid \theta_{\text{cur}})}}$$

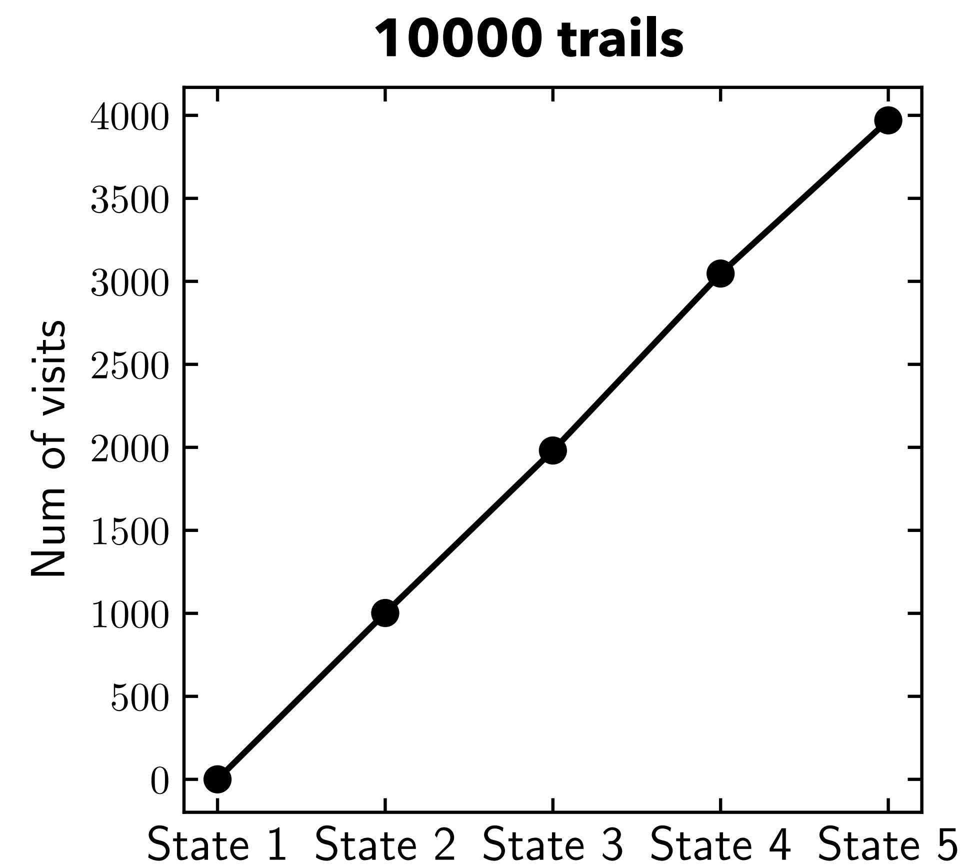
Proposal acceptance rate

$q(A \mid B)$ **is a proposal kernel from B to A.**
The ratio corrects for the bias introduced by the proposal kernel.

Metropolis-~~Hastings~~ Algorithm with an unbiased coin.

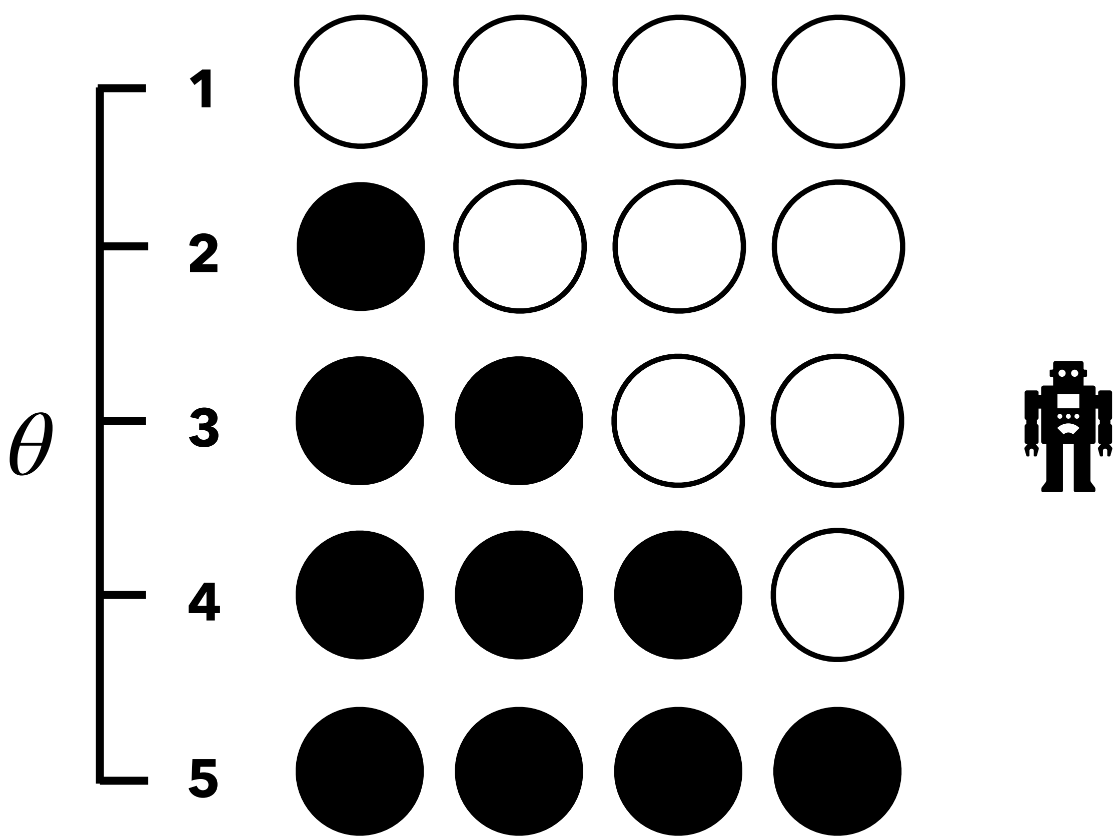
We draw from [-1, +1] to propose the next state.





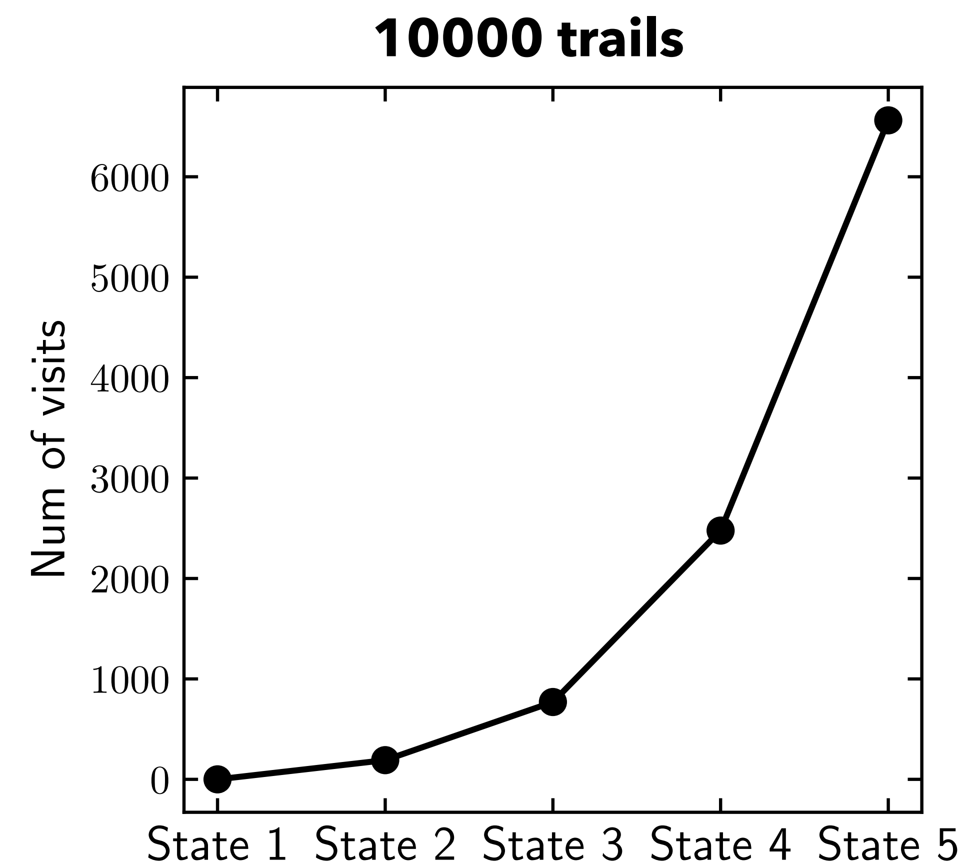
Metropolis-Hastings Algorithm with a biased coin.

We draw from [-1, -1, +1, +1, +1, +1] to propose the next state.



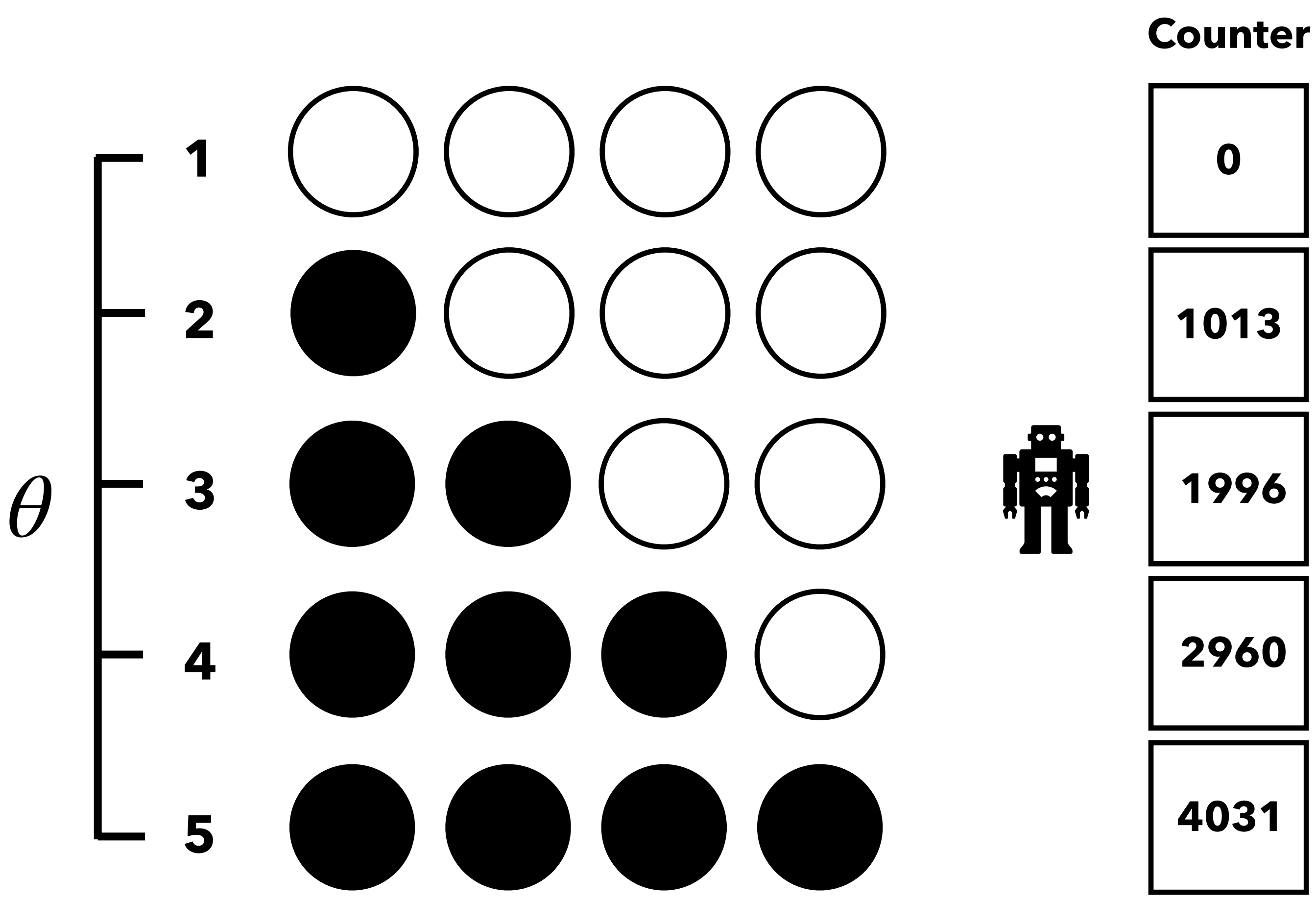
Counter
0
191
770
2477
6562

Before Hastings correction

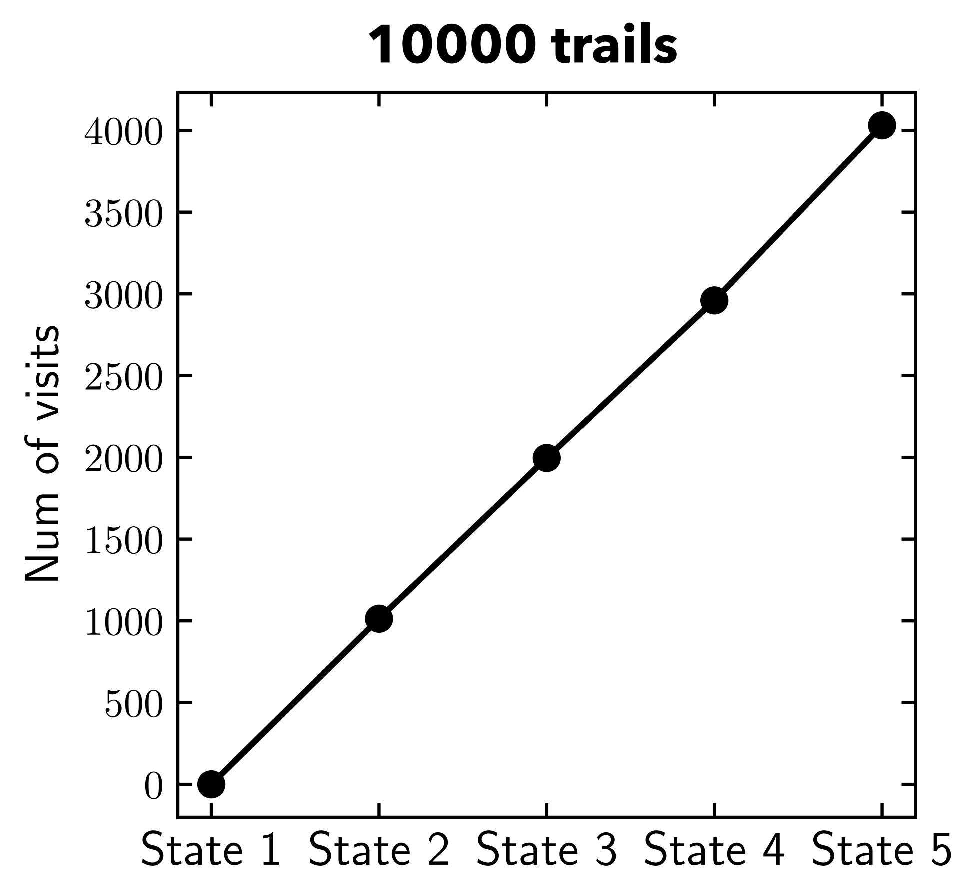


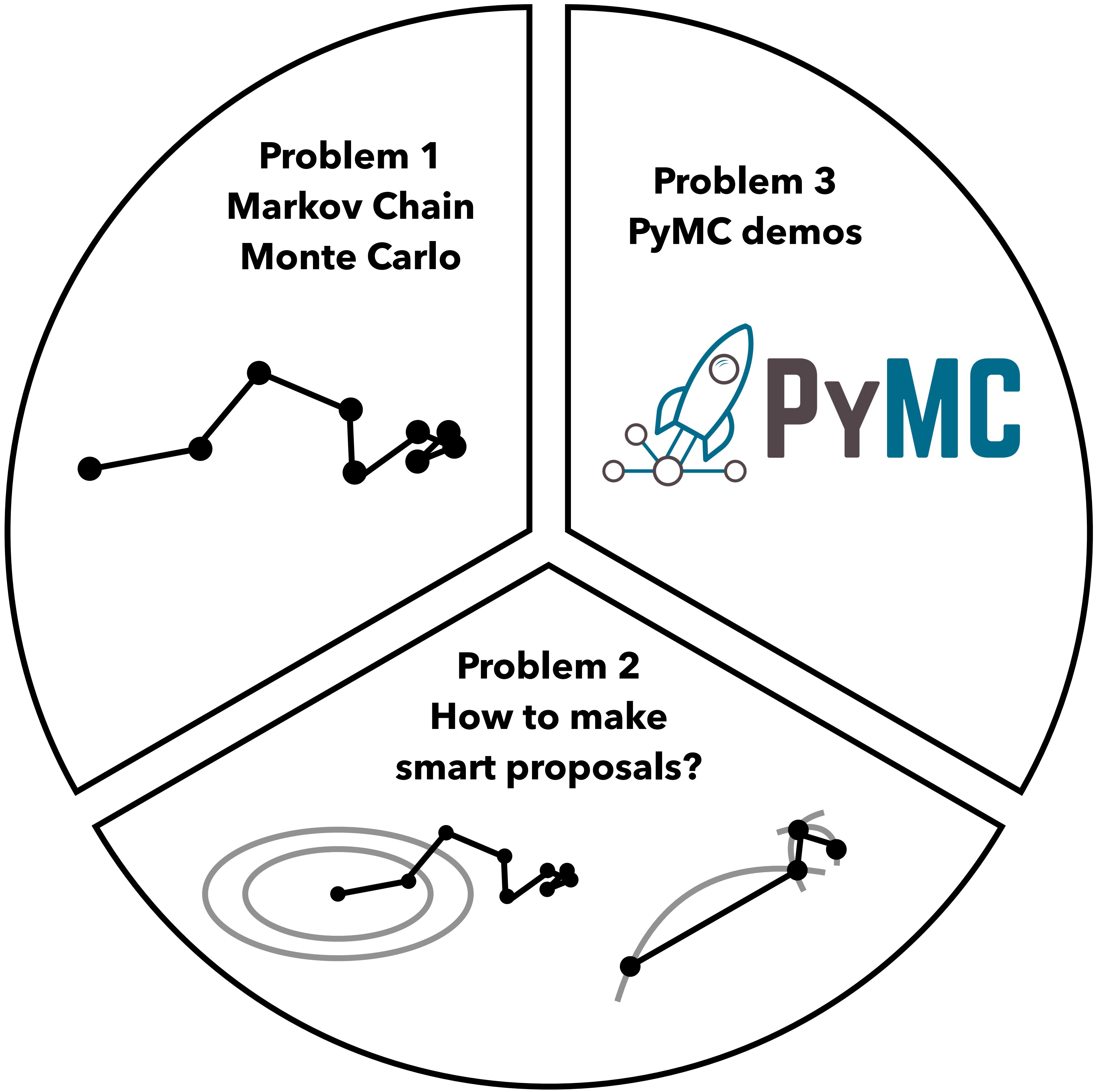
Metropolis-Hastings Algorithm with a biased coin.

We draw from [-1, -1, +1, +1, +1, +1] to propose the next state.



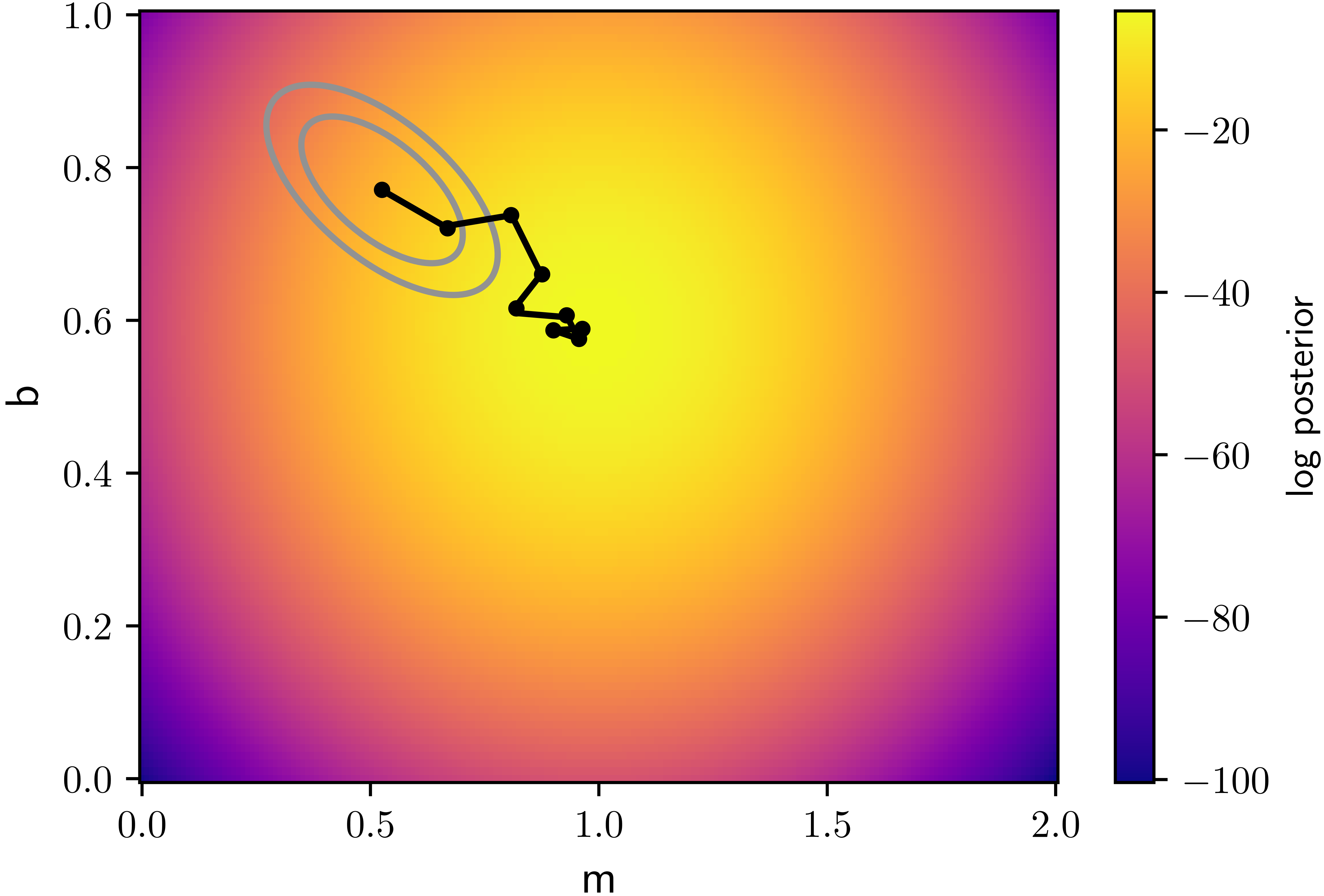
After Hastings correction



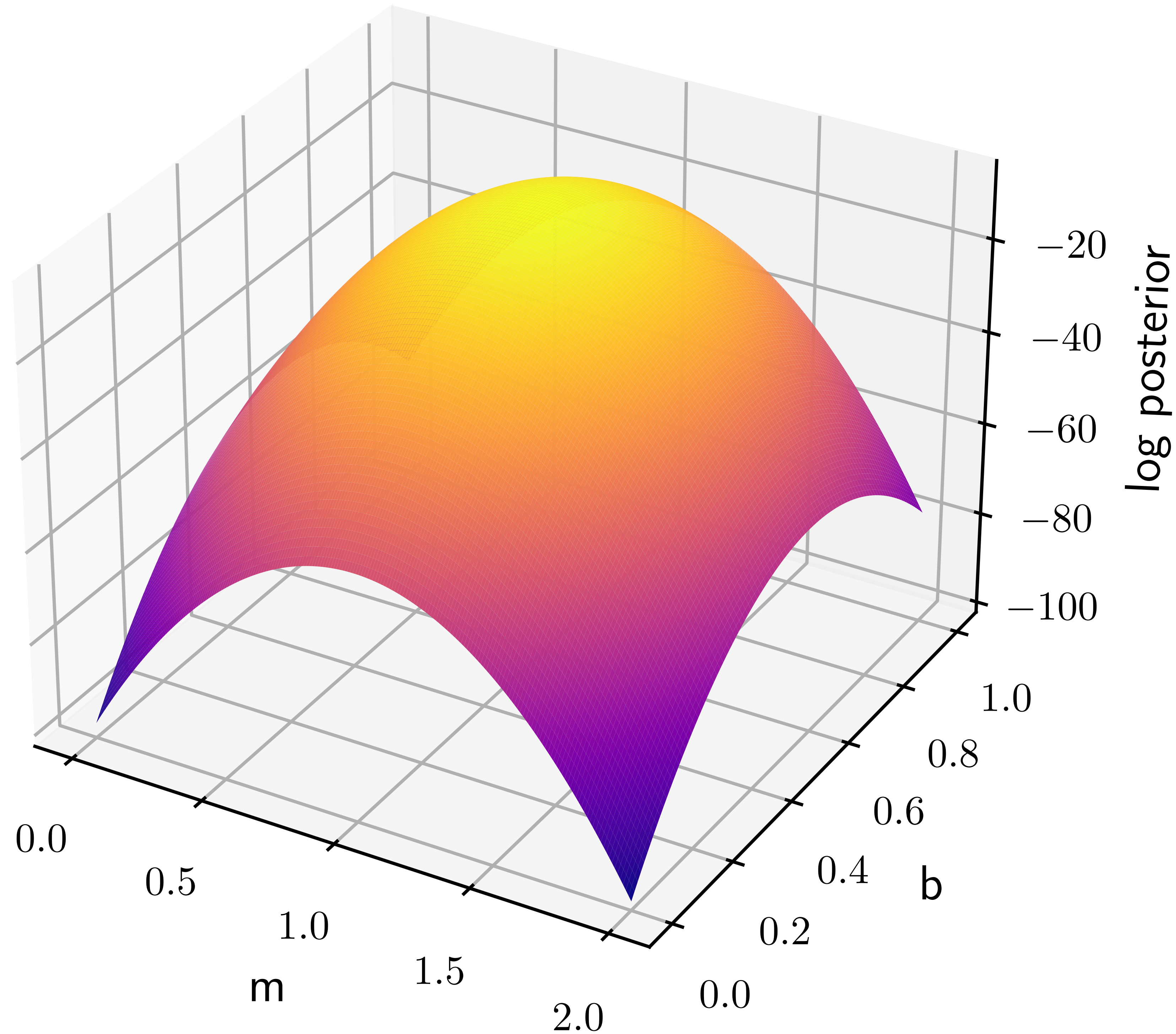


Problem 1
Markov Chain
Monte Carlo
Problem 3
PyMC demos
PyMC
Problem 2
How to make
smart proposals?

Random Walk Chain



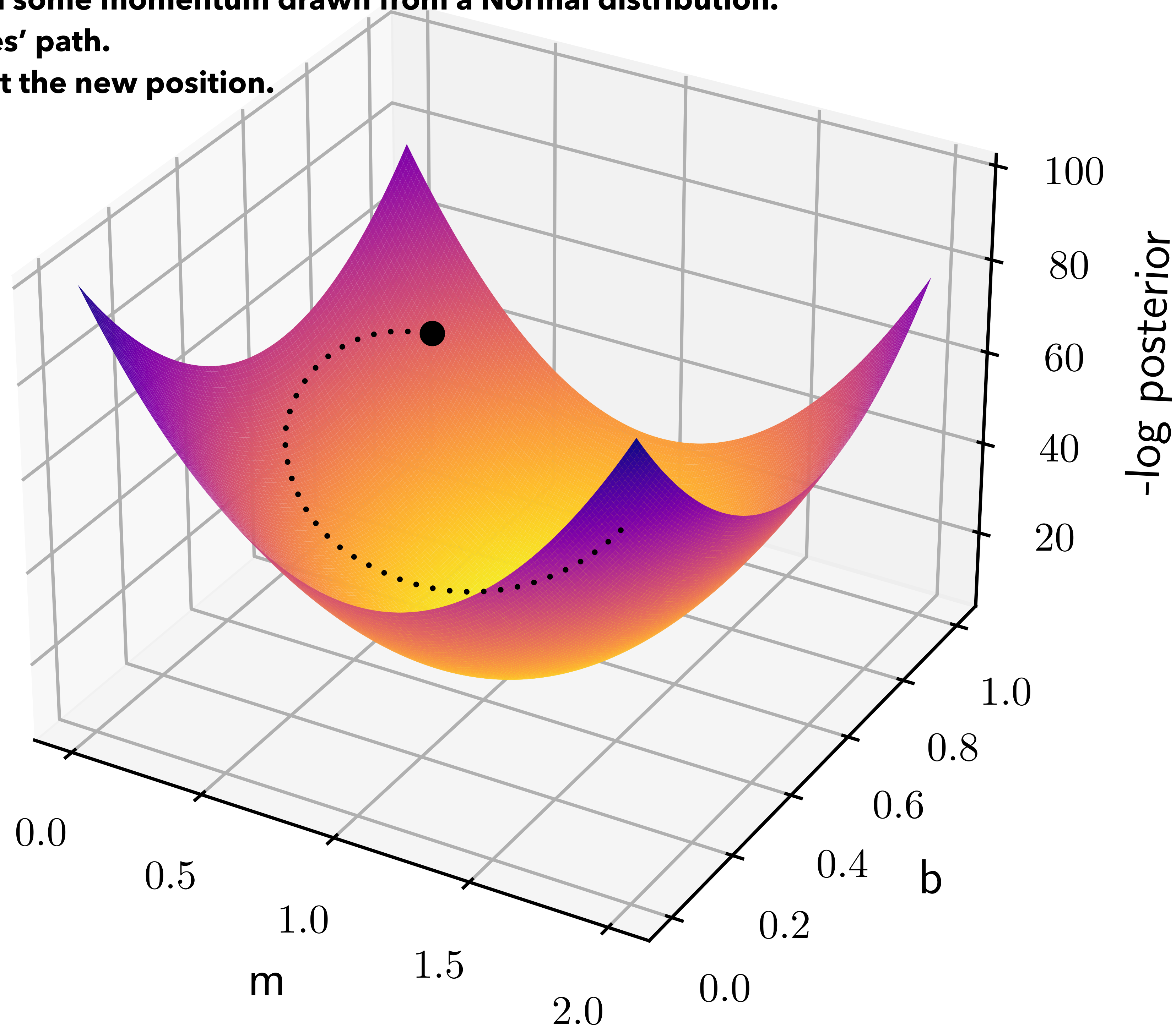
Hamiltonian Monte Carlo



Hamiltonian Monte Carlo

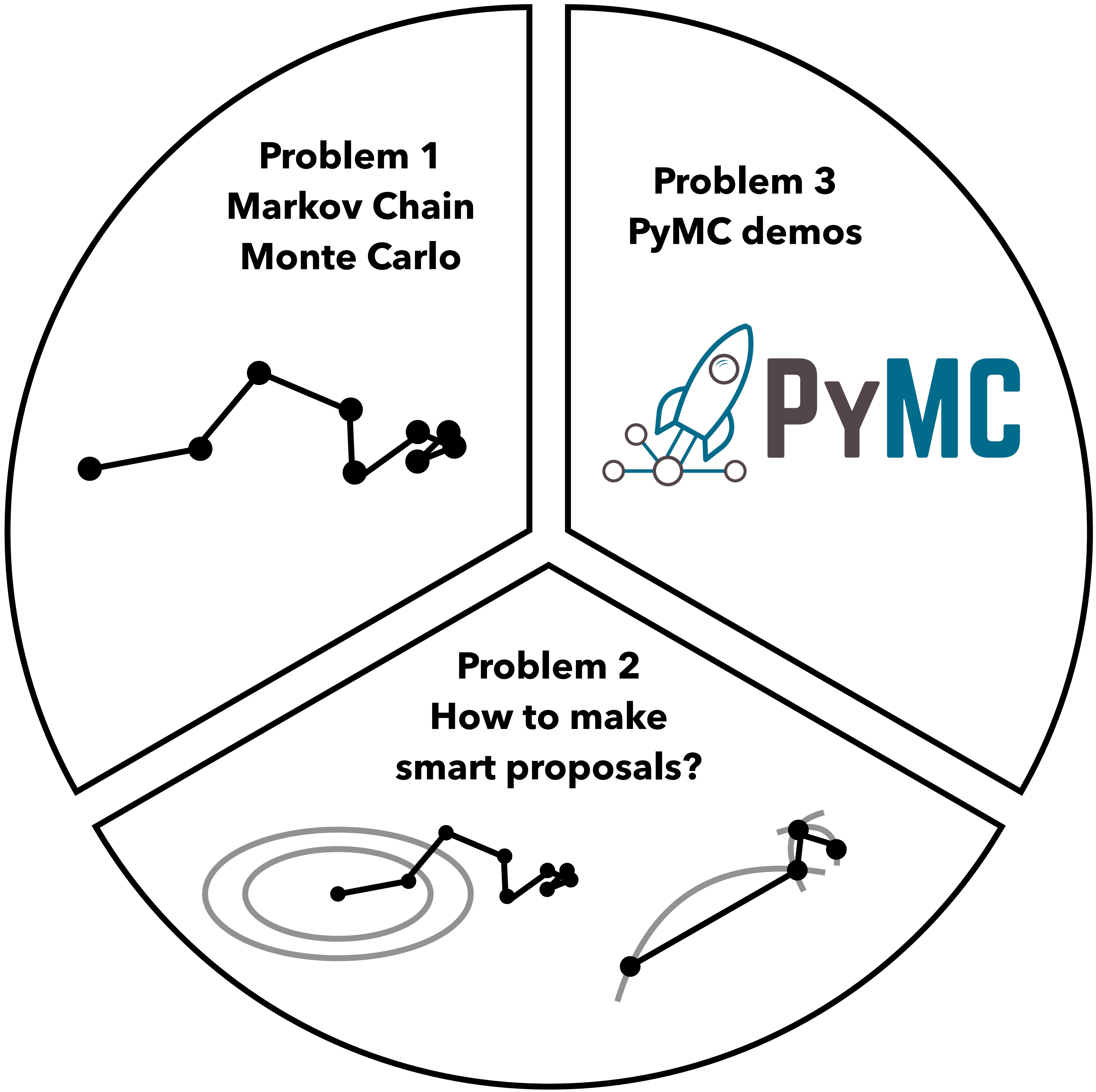
Begin with some random position.

- **Kick the particle with some momentum drawn from a Normal distribution.**
- **Integrate the particles' path.**
- **Almost always accept the new position.**



Hamiltonian Monte Carlo

- **Define a Hamiltonian system, where** $H = U(\theta) + K(v)$**.**
- **We can write the system's Gibbs canonical distribution as** $p(\theta, v) \propto e^{-H} \propto e^{-U(\theta)}e^{-K(v)}$**. The probability of observing** θ **at certain state is described as** $p(\theta) \propto e^{-U(\theta)}$**.**
- **Let's then define** $U(\theta) = -\log(P(\theta|D))$**.**
- **It's saying sampling particles in the space, which follows the canonical distribution, will also represent the posterior distribution.**


Problem 1
Markov Chain
Monte Carlo
Problem 3
PyMC demos
PyMC
Problem 2
How to make
smart proposals?

```python
# Step 1: built a pymc model
with pm.Model() as model:
  # priors
  mu = pm.Normal('mu', mu=1., sigma=0.1, initval=1.)
  sd = pm.HalfNormal('sd', sigma=1., initval=0.5)

  # likelihood
  logl = pm.Normal('logl', mu=mu, sigma=sd, observed=data)

# Step 2: sampling
with model:
  pm.sample(tune=1000, draws=1000, target_accept=0.9)
```

References

- McElreath, R. (2020). Statistical Rethinking: A Bayesian Course with Examples in R and Stan, 2nd Edition (2 ed.) CRC Press. (book)

- Hamiltonian Monte Carlo explained

  https://arogozhnikov.github.io/2016/12/19/markov_chain_monte_carlo.html