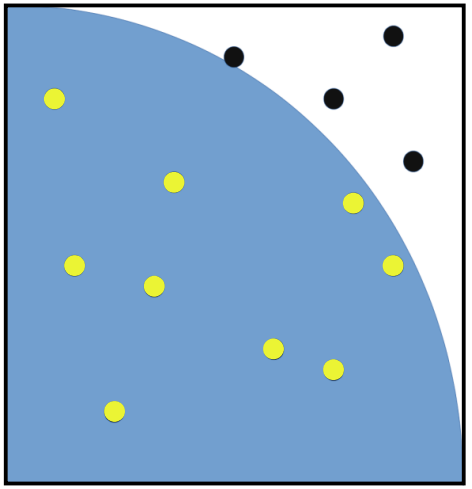
Markov Chain Monte Carlo

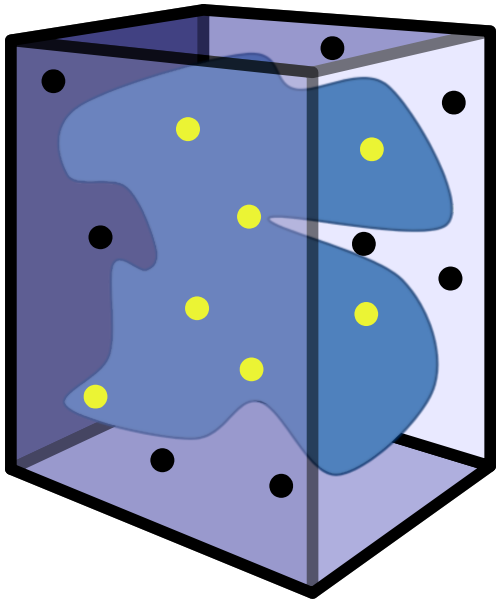
Estimating volumes with Monte Carlo

- Basic example: estimate the area of a quarter circle (2-dimensional problem)
- Throw n points uniformly in a square $[0, 1]^2$
- For each point, check if $x^2 + y^2 \leq 1$
- Estimate the area as: $\left(\frac{\text{points inside}}{\text{total points}}\right) \times (\text{area of the square})$



Estimating volumes with Monte Carlo

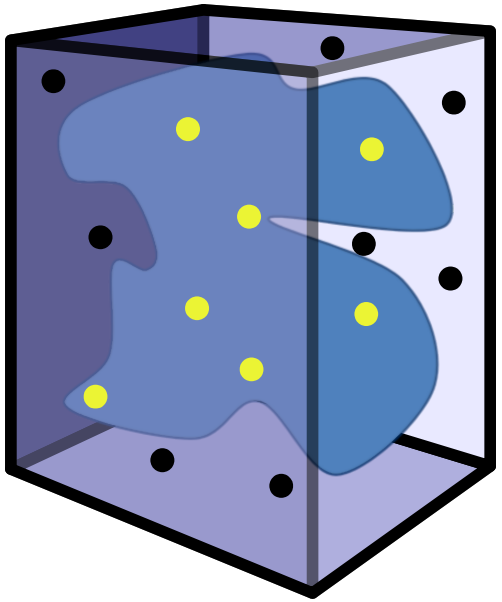
- Basic example: estimate the area of a quarter circle (2-dimensional problem)
- Throw n points uniformly in a square $[0, 1]^2$
- For each point, check if $x^2 + y^2 \leq 1$
- Estimate the area as: $\left(\frac{\text{points inside}}{\text{total points}}\right) \times (\text{area of the square})$



Let's generalize it

Estimating volumes with Monte Carlo

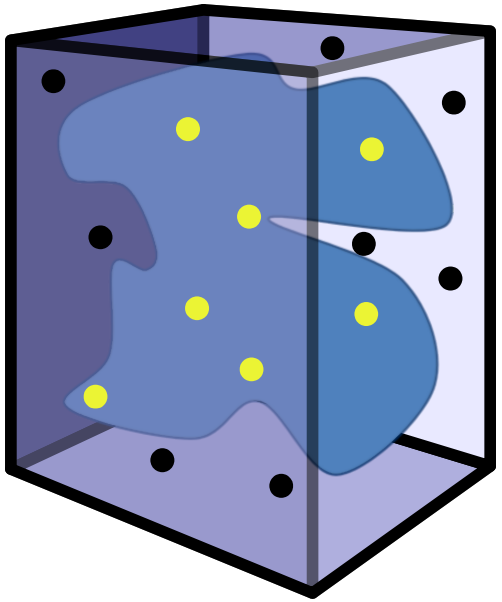
- General case: estimate the volume of a region (m-dimensional problem)
- Throw n points uniformly in a square $[0,1]^2$
- For each point, check if $x^2 + y^2 \leq 1$
- Estimate the area as: $\left(\frac{\text{points inside}}{\text{total points}}\right) \times (\text{area of the square})$



Let's generalize it

Estimating volumes with Monte Carlo

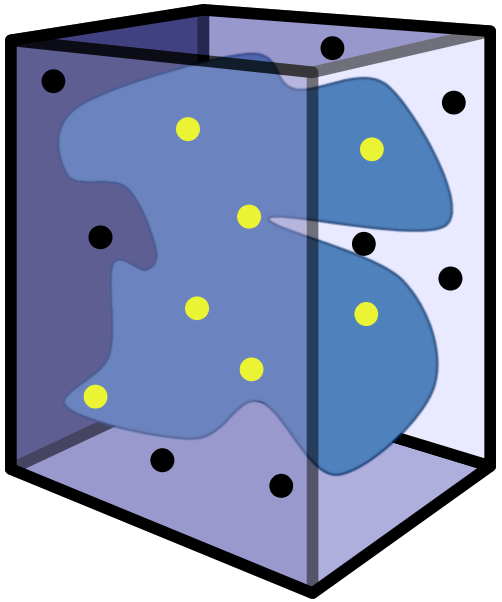
- General case: estimate the volume of a region (m-dimensional problem)
- Throw n points uniformly in a bounding box of known volume
- For each point, check if $x^2 + y^2 \leq 1$
- Estimate the area as: $\left(\frac{\text{points inside}}{\text{total points}}\right) \times (\text{area of the square})$



Let's generalize it

Estimating volumes with Monte Carlo

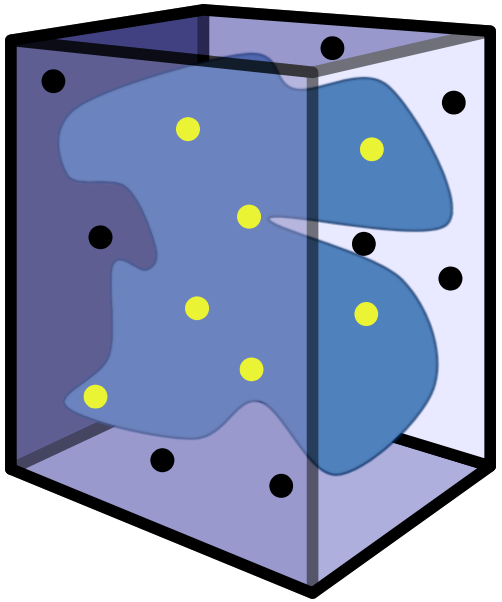
- General case: estimate the volume of a region (m-dimensional problem)
- Throw n points uniformly in a bounding box of known volume
- For each point, check if it's inside the region
- Estimate the area as: $\left(\frac{\text{points inside}}{\text{total points}}\right) \times (\text{area of the square})$



Let's generalize it

Estimating volumes with Monte Carlo

- General case: estimate the volume of a region (m-dimensional problem)
- Throw n points uniformly in a bounding box of known volume
- For each point, check if it's inside the region
- Estimate the volume as: $\left(\frac{\text{points inside}}{\text{total points}}\right) \times (\text{volume of the box})$

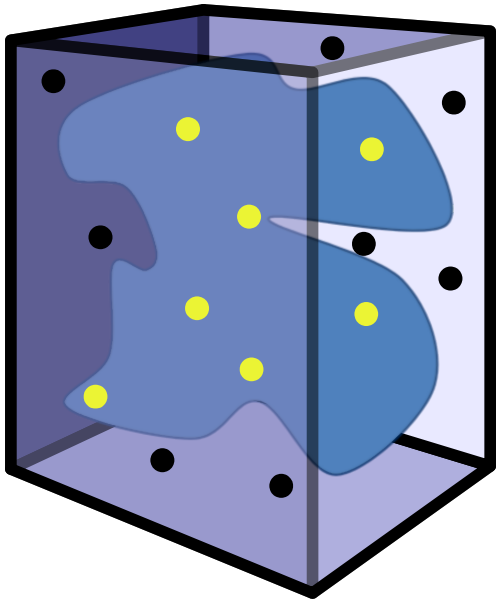


Let's generalize it

Estimating volumes with Monte Carlo

- General case: estimate the volume of a region (m-dimensional problem)
- Throw n points uniformly in a bounding box of known volume
- For each point, check if it's inside the region
- Estimate the volume as: $\left(\frac{\text{points inside}}{\text{total points}}\right) \times (\text{volume of the box})$

- **Error decreases like $1/\sqrt{n}$ (independent of m)**
- **Using a grid (Riemann-style) would decrease much more slowly, like $1/\sqrt[m]{n}$**
- **(Details in the notes)**

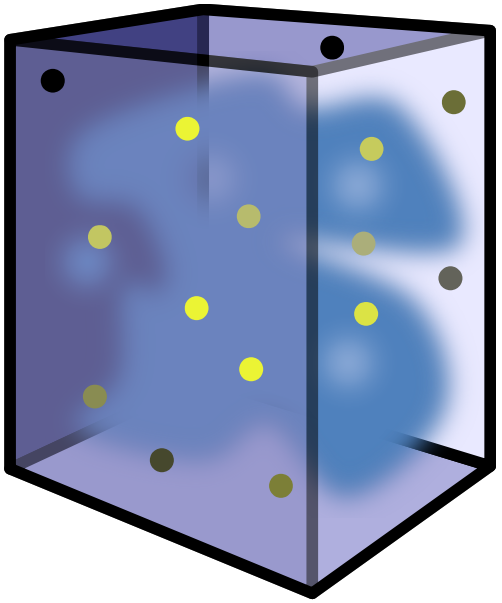


Requires:

- **Tight-enough bounding box (good chance of hitting the region)**
- **Fast check of inclusion in the region**

Generalization to integrals

- Estimate the value of a multidimensional integral (m-dimensional problem)
- Throw n points uniformly in a bounding box of known volume
- Weigh each point with the value of the function
- Estimate the integral as: $\left(\frac{1}{n}\sum_{i=1}^{n} f\left(\vec{x}_i\right)\right) \times (\text{volume of the box})$

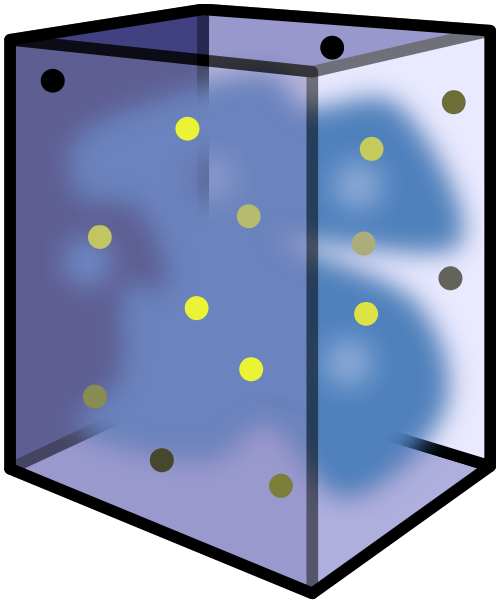


Generalization to integrals

- Estimate the value of a multidimensional integral (m-dimensional problem)
- Throw n points uniformly in a bounding box of known volume
- Weigh each point with the value of the function
- Estimate the integral as: $\left(\frac{1}{n}\sum_{i=1}^{n} f\left(\vec{x}_i\right)\right) \times (\text{volume of the box})$

Works well, but :

- **What if the function is almost 0 almost everywhere except in tiny regions where it's large?**
- **It may be hard to hit those regions by chance**
- **If the function is smooth enough, we might find them...**



Detour: stochastic processes

(rather informal, see notes)

- A *stochastic process* is a sequence of random variables $X^0, X^1, X^2, \ldots$, assuming that each of them produces values from the same space $\mathcal{X}$. The sequence may be finite or infinite. The space $\mathcal{X}$ may be continuous or discrete, and it's often multidimensional. We'll focus on the discrete case.
- A realization of the stochastic process is a sequence of values $x^0, x^1, x^2, \ldots$ each of which belongs to $\mathcal{X}$. This is called a *trajectory*, because we think of the index of each value as representing a time. You may think of a point moving inside $\mathcal{X}$, in (random) steps.
- We are normally interested in studying how the probability distribution of one of the variables at time t, X^t, depends on what happened at the previous times. In formulas:

$$P\left(X^t = x^t \mid X^0 = x^0, \ldots, X^{t-1} = x^{t-1}\right)$$

Probability of being at x^t at time t ...

...given the trajectory $x^0, x^1, \ldots, x^{t-1}$ up to time $t-1$

Stochastic processes: general comments

- We use stochastic processes to model sequences of events that happen at random, but that could be influenced but what happened in the past.
- This idea is very general and very useful. Examples abound (in physics, biology, finance...).
- It's so general in fact that most examples are quite hard, even impossible, to analyze.

Forgetful processes: Markov chains

- The simplest possible kind of stochastic process is one in which every variable X^t is independent of the others. This represents disconnected events (e.g. repeatedly tossing a coin) and completely loses the idea of a "process" – time is irrelevant.
- Thus the simplest "interesting" case is when each new event only depends on the previous one, and all of the more distant past is "forgotten" (ignored). In formulas:

$$P\left(x^{t+1} \mid x^t, x^{t-1}, \ldots, x^0\right) = P\left(x^{t+1} \mid x^t\right)$$

Markov property

Simplified notation!
$X^t = x^t$ became x^t

Dependence expressed by conditional probabilities

- The probability for the whole trajectory (up to some t) can thus be decomposed as a chain:

$$P\left(x^0, x^1, \ldots, x^t\right) = P\left(x^t \mid x^{t-1}\right) \ldots P\left(x^2 \mid x^1\right) P\left(x^1 \mid x^0\right) P\left(x^0\right)$$

$$= \left(\prod_{t'=0}^{t-1} P\left(x^{t'+1} \mid x^{t'}\right)\right) P\left(x^0\right)$$

Markov chain

A useful formula for Markov chains

- The probability of ending up in x^t at time t can be written as:

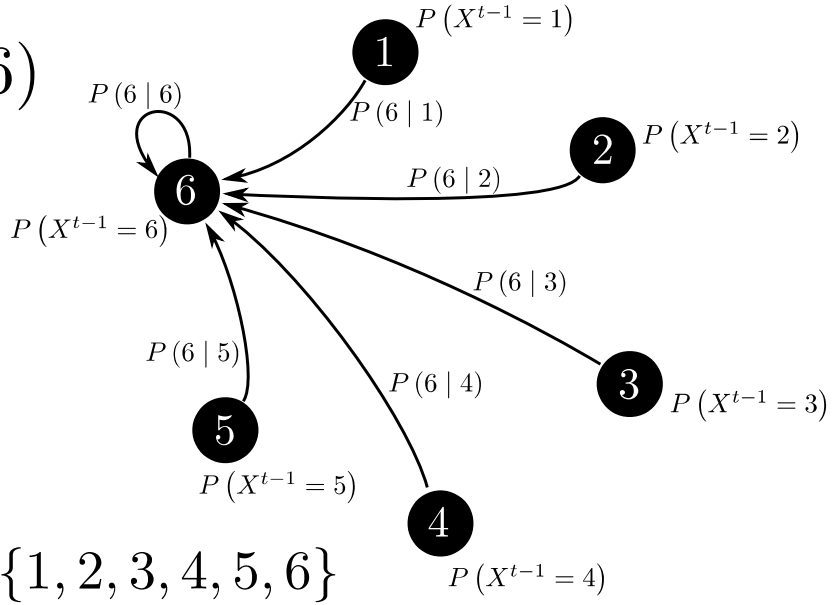
$$P\left(x^t\right) = \sum_{x^{t-1}} P\left(x^t \mid x^{t-1}\right) P\left(x^{t-1}\right)$$

Transition probability

Marginal probabilities

- A graphical interpretation: it's a sort of "incoming flow" into a state. Each state has a "mass" at time $t-1$ which gets multiplied by a "transition rate", and they all get summed up. Example:

$$P\left(X^t = 6\right)$$



$$\mathcal{X} = \{1, 2, 3, 4, 5, 6\}$$

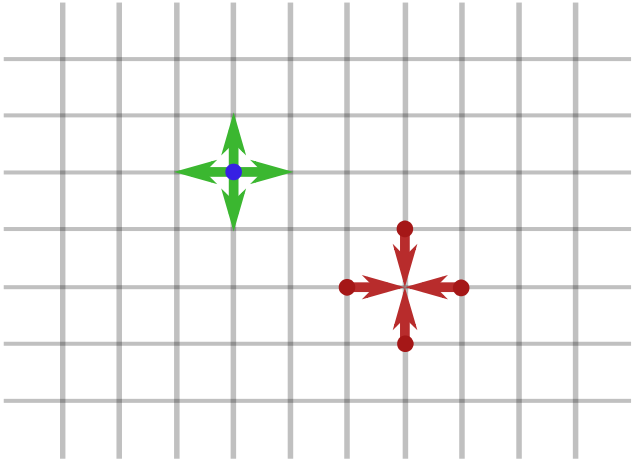
Simple Markov chains example: random walk on a grid

- Suppose you have an infinite regular square grid on a plane. The collection of all possible states $\mathcal{X}$ is represented by all possible pairs of integer numbers (i, j).
- Say we start at $(0, 0)$. This means that $P\left(X^0 = (i,j)\right) = \begin{cases} 1 & \text{if } (i,j) = (0,0) \\ 0 & \text{otherwise} \end{cases}$
- At each step, we pick one of the 4 possible directions at random and move there [green arrows]

$$P\left(X^{t+1} = (i,j) \mid X^t = (k,l)\right) = \begin{cases} 1/4 & \text{if } (i = k \pm 1 \text{ and } j = l) \text{ or } (i = k \text{ and } j = l \pm 1) \\ 0 & \text{otherwise} \end{cases}$$

use in probability update formula (previous slide)

$$\begin{aligned} P\left(X^{t+1} = (i,j)\right) = {} & \frac{1}{4} P\left(X^t = (i+1,j)\right) + \\ & \frac{1}{4} P\left(X^t = (i-1,j)\right) \\ & \frac{1}{4} P\left(X^t = (i,j+1)\right) \\ & \frac{1}{4} P\left(X^t = (i,j-1)\right) \end{aligned}$$

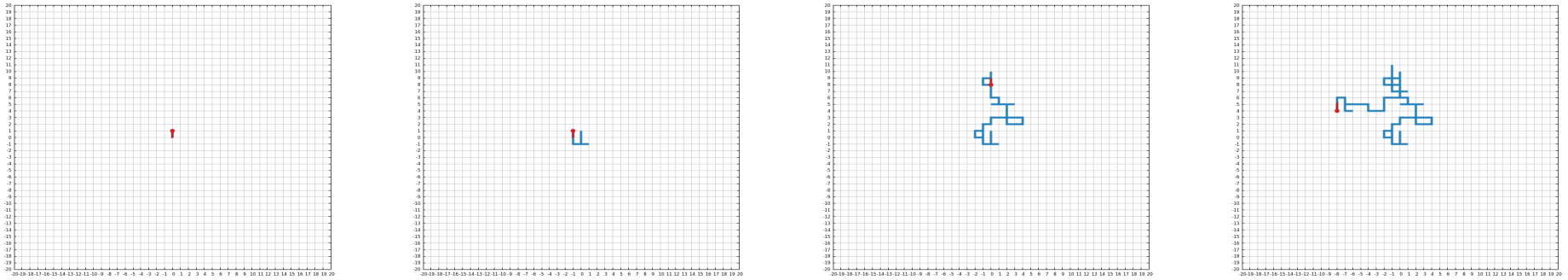


green: individual trajectory step
red: probability update

Evolution: One trajectory vs probability

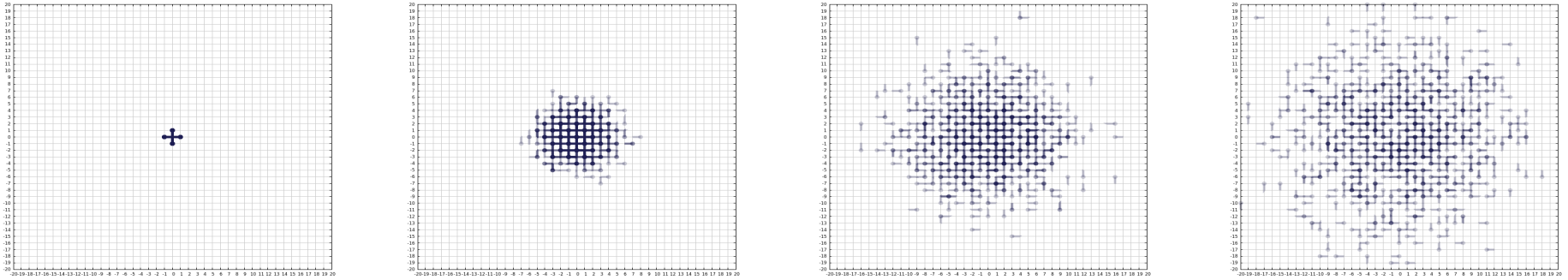
t=1 t=10 t=50 t=100

trajectory



see animation

probability



see animation

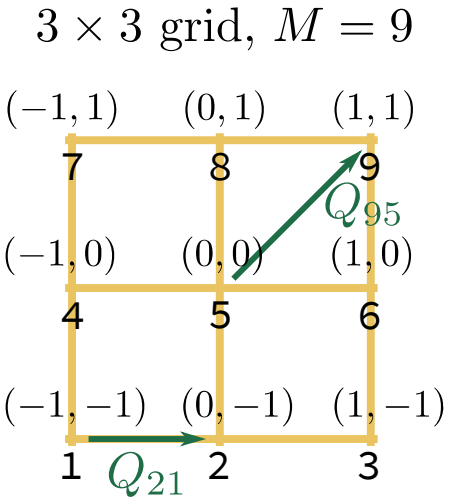
Here we sample 1000 trajectories in parallel and take snapshots at different times, and it starts looking like the (marginal) probability distribution $P\left(x^t\right)$

Markov chains probability evolution → Linear algebra!

- Say that $\mathcal{X}$ is discrete and finite. Say that it has M elements. Say that you order them somehow, thus you have a label for each element, $1, 2, \ldots, M$.
- Then you can encode the transition rates $P\left(x^{t+1} \mid x^t\right)$ in a $M \times M$ matrix. For each "source" state i and "destination" state j and time t, you have a probability of getting to j at $t+1$ assuming that you are in i at time t. Call this matrix $Q_{ji}^{(t)}$.
- You can also encode the marginal probabilities $P\left(x^t\right)$ in a vector of size M, call it $\vec{\rho}^{(t)}$

Notational gotchas here!

- $x^t, x^{t+1}, \cdots \in \mathcal{X}$ denote configurations. E.g. in the random walk case they could be $(0,0)$ or $(-3,6)$
- $i, j, \cdots \in \{1, 2, \ldots, M\}$ are configuration labels. Think of a random walk on a finite $L \times L$ grid, then start labeling row by row
- in $Q_{ji}^{(t)}$, i is the source, j is the destination



Markov chains probability evolution → Linear algebra!

- Say that $\mathcal{X}$ is discrete and finite. Say that it has M elements. Say that you order them somehow, thus you have a label for each element, $1, 2, \ldots, M$.
- Then you can encode the transition rates $P\left(x^{t+1} \mid x^t\right)$ in a $M \times M$ matrix. For each "source" state i and "destination" state j and time t, you have a probability of getting to j at $t+1$ assuming that you are in i at time t. Call this matrix $Q_{ji}^{(t)}$.
- You can also encode the marginal probabilities $P\left(x^t\right)$ in a vector of size M, call it $\vec{\rho}^{(t)}$
- Then the formula for the evolution of the marginals in a Markov chain can be expressed in linear algebra notation

$$P\left(x^{t+1}\right) = \sum_{x^t} P\left(x^{t+1} \mid x^t\right) P\left(x^t\right) \longrightarrow \rho_j^{(t+1)} = \sum_{i=1}^{M} Q_{ji}^{(t)} \rho_i^{(t)} \longrightarrow \boxed{\vec{\rho}^{(t+1)} = Q^{(t)} \vec{\rho}^{(t)}}$$

- Say that the transition rates do not depend on time (see e.g. random walk). We can write:

$$\begin{aligned}\vec{\rho}^{(t)} &= Q\vec{\rho}^{(t-1)} = Q\left(Q\vec{\rho}^{(t-2)}\right) = \cdots = Q\left(Q\left(Q\ldots\left(Q\vec{\rho}^{(0)}\right)\right)\right) \\ &= (QQQ\ldots Q)\,\vec{\rho}^{(0)} = Q^t\vec{\rho}^{(0)}\end{aligned}$$

A very important property: stochastic matrices

- From any given initial state, we must end up somewhere. Therefore:

$$\sum_{x'} P\left(x' \mid x\right) = 1 \qquad \forall x \in \mathcal{X}$$

- In terms of the matrix notation:

$$\sum_{j=1}^{M} Q_{ji} = 1 \qquad \forall i \in \{1, \ldots, M\}$$

- Read: the sum of the elements along each column must be 1. A matrix of non-negative numbers with this property is a *stochastic matrix*.
- Alternative way to write this: isolate the case in which we stay in the same configuration:

$$Q_{ii} = 1 - \sum_{j \neq i} Q_{ji} \qquad \forall i \in \{1, \ldots, M\}$$

(in a stochastic matrix, the diagonal elements are determined by the other elements)

Stationary distributions

- So we have $\vec{\rho}^{(t)} = Q^t \vec{\rho}^{(0)}$
- What happens as $t \to \infty$? In general it depends on Q and $\vec{\rho}^{(0)}$, but notice:
  - If we think about a single trajectory, it will generally just keep going, jumping between states randomly (e.g. random walk)
  - But if we think about the probability, that instead may converge to a limit distribution: $\vec{\rho}^{(t)} \xrightarrow[t\to\infty]{} \vec{\rho}^{\star}$
- A *stationary distribution* is characterized by the property $\vec{\rho}^{\star} = Q\vec{\rho}^{\star}$

  Doesn't change with time.
  In linear algebra terms: it's an eigenvector with eigenvalue 1.
- Given a stochastic process, the stationary distribution (if it exists) can be simpler to describe and easier to compute than the intermediate steps.
- In rough practical terms: run a simulation "long enough", and then when you look at which state you find it in, you can think of it as having been sampled from $\vec{\rho}^{\star}$.

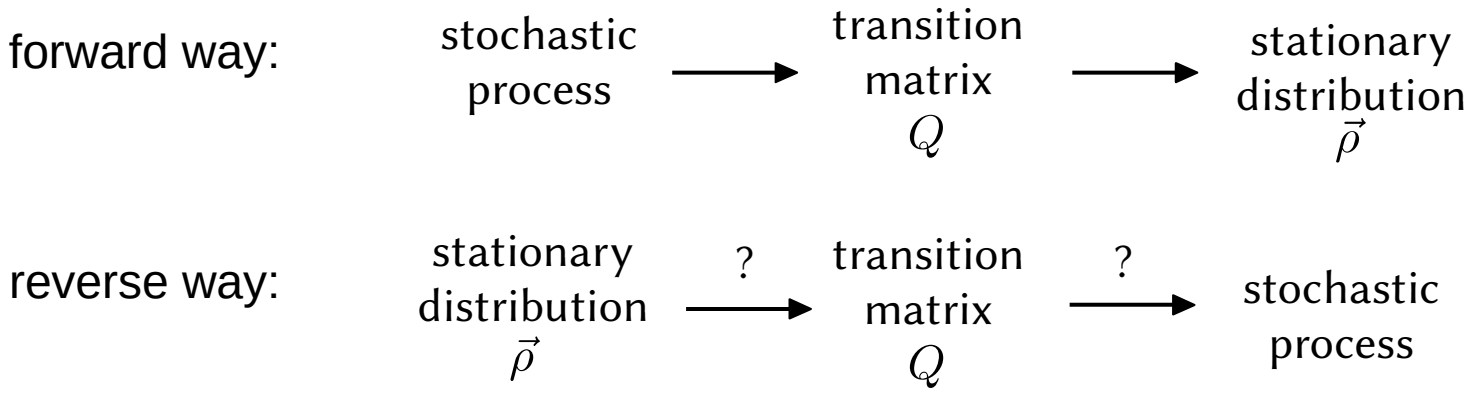
Going in reverse: given a stationary distribution, find a Markov chain

- We can use the last observation as follows: suppose that you want to obtain samples from some distribution $\vec{\rho}$, but that for some reason (we'll see examples later) it's hard to do it directly.

dropping the $\star$ here...

- Can we instead build a Markov chain (i.e. can we find a transition matrix Q) such that its stationary distribution is $\vec{\rho}$? If we can, then we can obtain samples by just running the process for a long time [provided the process we found is *ergodic*; see notes].

- As it turns out, we can. Actually, there are in general infinite processes that realize any given stationary distribution. Moreover, there is always one family of processes that can be built in a systematic way from a very simple rule. This is called the Metropolis-Hastings algorithm.



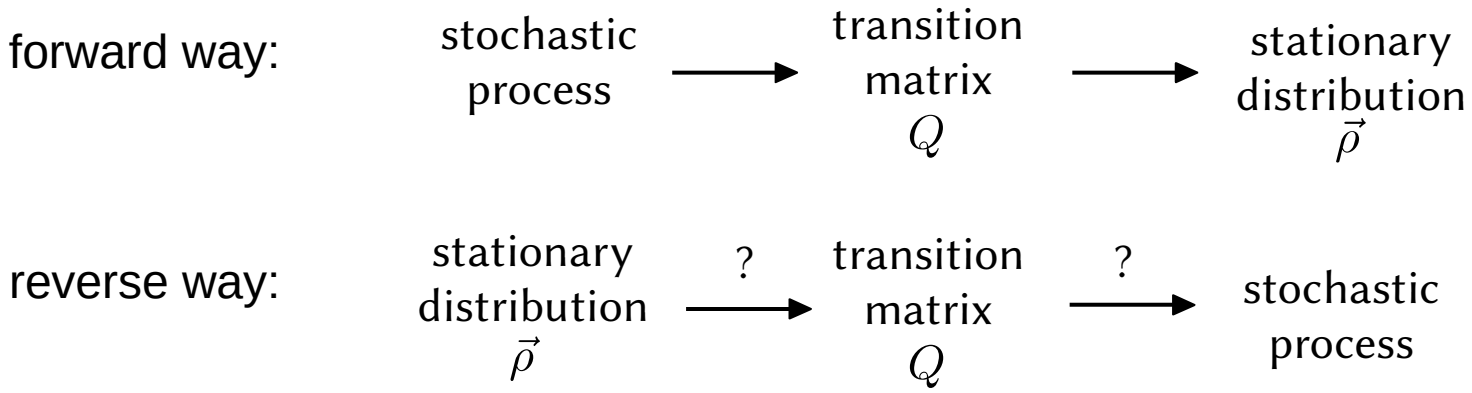
Going in reverse: given a stationary distribution, find a Markov chain

- We can use the last observation as follows: suppose that you want to obtain samples from some distribution $\vec{\rho}$, but that for some reason (we'll see examples later) it's hard to do it directly.

dropping the $\star$ here...

- Can we instead build a Markov chain (i.e. can we find a transition matrix Q) such that its stationary distribution is $\vec{\rho}$? If we can, then we can obtain samples by just running the process for a long time [provided the process we found is *ergodic*; see notes].

- As it turns out, we can. Actually, there are in general infinite processes that realize any given stationary distribution. Moreover, there is always one family of processes that can be built in a systematic way from a very simple rule. This is called the Metropolis-Hastings algorithm.



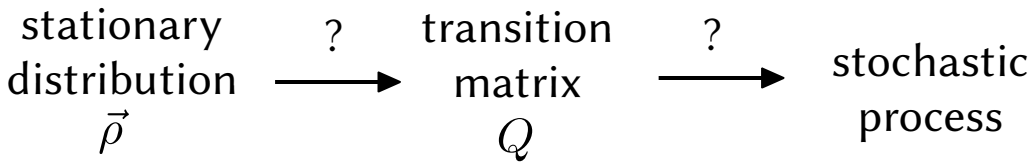
Markov Chain Monte Carlo (MCMC)

The Metropolis-Hastings algorithm: overview

The most popular MCMC method

- We are given $\vec{\rho}$, and looking for a suitable Q
- We split the transitions in two parts: "proposal" and "acceptance"
- At each step, we [stochastically] propose a move towards a different configuration
  - The proposal matrix can be chosen almost arbitrarily. Generally we propose moves from a configuration to a "nearby" one, and also keep it simple.
- Then we [stochastically] decide whether to accept the proposal. Refusing the move means not moving, which is the same as transitioning to the current state.
  - The acceptance rule cannot be completely arbitrary.
  - Metropolis-Hastings provides a simple and effective formula for the acceptance rule.

reverse way:



The Metropolis-Hastings algorithm (I)

- Start from the stationarity condition: $\vec{\rho} = Q\vec{\rho}$
- Rewrite using indices explicitly, isolate the diagonal term, use the stochasticity property, rearrange (see notes) ⟶ global balance condition [necessary and sufficient; also not restrictive enough]

$$\forall i: \quad \sum_{j \neq i} Q_{ji}\rho_i = \sum_{j \neq i} Q_{ij}\rho_j$$

For each configuration: total "outflow" of probabilities equals the total "influx" of probabilities

- Stronger condition: **detailed balance** [sufficient, not necessary]

$$\forall i, j: \quad Q_{ij}\rho_j = Q_{ji}\rho_i$$

The probability flows are now balanced for every pair of configurations

- Split the non-diagonal part in two: "proposal" C_{ij} and "acceptance" A_{ij}

$$Q_{ij} = C_{ij}A_{ij} \quad \text{if } i \neq j$$

The diagonal terms are set by the stochasticity property!

- The proposal is normalized (it's a stochastic matrix), and it's zero along the diagonal (we always propose a different configuration from the current one)

reverse way:

The Metropolis-Hastings algorithm (II)

- Given the detailed balance condition and the structure of the proposals (see next slide), we get a condition on the acceptance rule (see notes)
- The most effective (in some sense, see notes) acceptance rule is the Metropolis-Hastings rule:

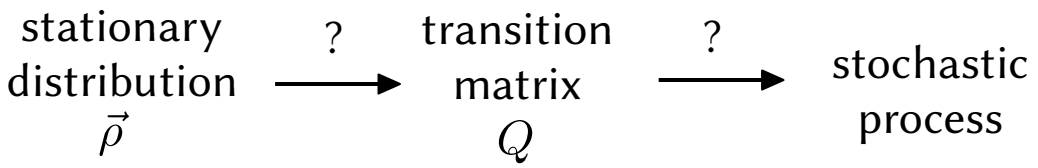
$$A_{ij} = \min\left(1, \frac{C_{ji}\rho_i}{C_{ij}\rho_j}\right)$$

Guarantees detailed balance

- If the proposals are symmetric, i.e. $C_{ij} = C_{ji}$, then the condition on the acceptance rule is simplified (sometimes called Metropolis rule):

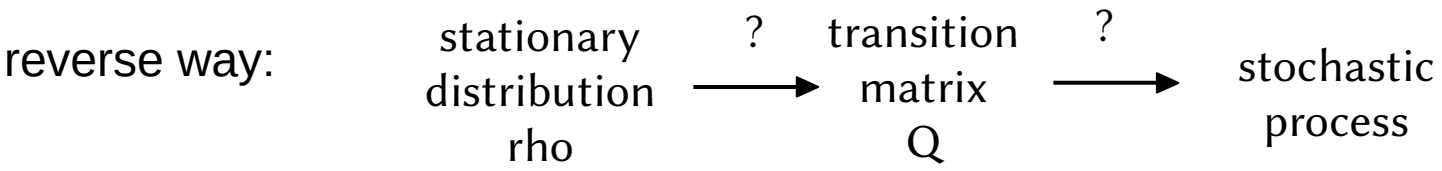
$$A_{ij} = \min\left(1, \frac{\rho_i}{\rho_j}\right)$$

reverse way:



The Metropolis-Hastings algorithm (III)

- The one thing we're missing is: how do we set the proposals?
- There are a few conditions that need to be enforced [required for *ergodicity*; see notes]:
  - No diagonal terms — **Always propose a change**
  - Connectedness — **It must be possible to get to any configuration from any other, given enough steps, with non-zero probability**
  - Aperiodicity — **There must be "enough" randomness not to get trapped in cycles**
- Apart from those, it's mostly arbitrary. But in practice some choices are better than others, for various reasons. For example:
  - Propose moves "close" to the current configuration
  - Try to keep it simple (e.g. symmetric, uniform, ...)
  - Try to get things to "mix up" well (extremely heuristic and vague, I know.. we'll see examples.)



A simple worked-out example

- Suppose we have $\mathcal{X} = \{1, 2, 3\}$ and we want $\vec{\rho} = \left(\frac{1}{2}, \frac{1}{4}, \frac{1}{4}\right)$
- Choose C
- Compute A from M.-H. rule
- Compute final Q

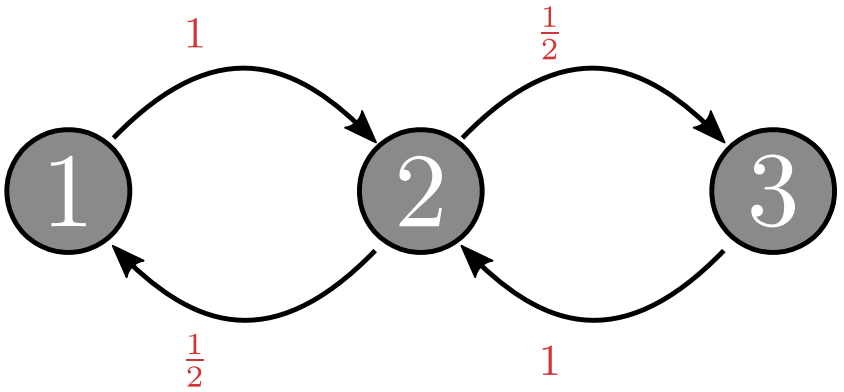
Simple case in which configurations ≡ labels

np.random.choice

A simple worked-out example

- Suppose we have $\mathcal{X} = \{1, 2, 3\}$ and we want $\vec{\rho} = \left(\frac{1}{2}, \frac{1}{4}, \frac{1}{4}\right)$
- Choose C
- Compute A from M.-H. rule
- Compute final Q

$$C = \begin{bmatrix} 0 & \frac{1}{2} & 0 \\ 1 & 0 & 1 \\ 0 & \frac{1}{2} & 0 \end{bmatrix}$$

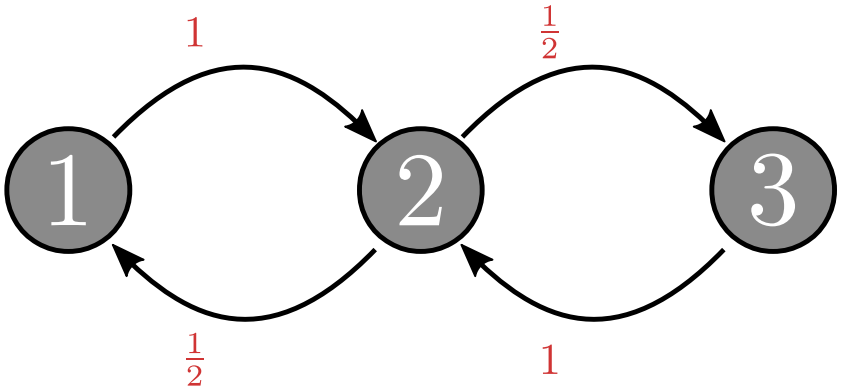


Diagonal is always 0.
Let's decide to only link neighbors.
This forces us to have a non-symmetric matrix.

A simple worked-out example

- Suppose we have $\mathcal{X} = \{1, 2, 3\}$ and we want $\vec{\rho} = \left(\frac{1}{2}, \frac{1}{4}, \frac{1}{4}\right)$
- Choose C
- Compute A from M.-H. rule
- Compute final Q

$$C = \begin{bmatrix} 0 & \frac{1}{2} & 0 \\ 1 & 0 & 1 \\ 0 & \frac{1}{2} & 0 \end{bmatrix}$$



$$A = \begin{bmatrix} 0 & \min\left(1, \frac{C_{21}\rho_1}{C_{12}\rho_2}\right) & 0 \\ \min\left(1, \frac{C_{12}\rho_2}{C_{21}\rho_1}\right) & 0 & \min\left(1, \frac{C_{32}\rho_2}{C_{23}\rho_3}\right) \\ 0 & \min\left(1, \frac{C_{23}\rho_3}{C_{32}\rho_2}\right) & 0 \end{bmatrix}$$

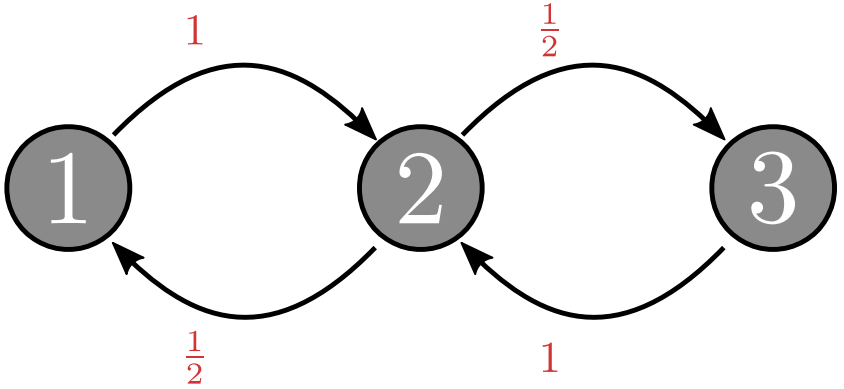
No symmetry in the proposals, we need to use the more general formula

$$A_{ij} = \min\left(1, \frac{C_{ji}\rho_i}{C_{ij}\rho_j}\right)$$

A simple worked-out example

- Suppose we have $\mathcal{X} = \{1, 2, 3\}$ and we want $\vec{\rho} = \left(\frac{1}{2}, \frac{1}{4}, \frac{1}{4}\right)$
- Choose C
- Compute A from M.-H. rule
- Compute final Q

$$C = \begin{bmatrix} 0 & \frac{1}{2} & 0 \\ 1 & 0 & 1 \\ 0 & \frac{1}{2} & 0 \end{bmatrix}$$



$$A = \begin{bmatrix} 0 & \min\left(1, \frac{1\frac{1}{2}}{\frac{1}{2}\frac{1}{4}}\right) & 0 \\ \min\left(1, \frac{\frac{1}{2}\frac{1}{4}}{1\frac{1}{2}}\right) & 0 & \min\left(1, \frac{\frac{1}{2}\frac{1}{4}}{1\frac{1}{4}}\right) \\ 0 & \min\left(1, \frac{1\frac{1}{4}}{\frac{1}{2}\frac{1}{4}}\right) & 0 \end{bmatrix}$$

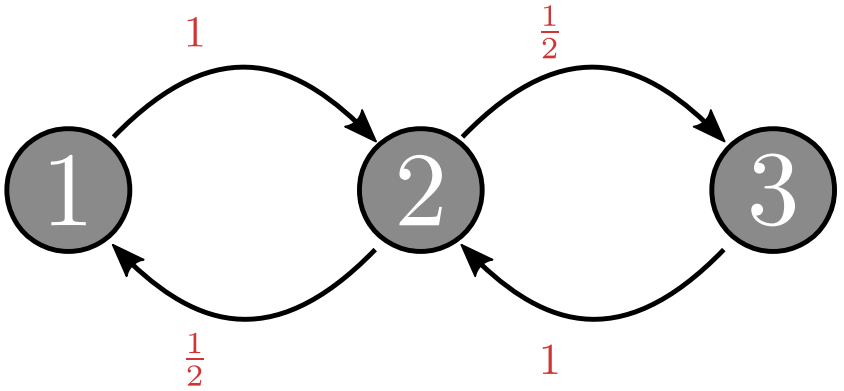
No symmetry in the proposals, we need to use the more general formula

$$A_{ij} = \min\left(1, \frac{C_{ji}\rho_i}{C_{ij}\rho_j}\right)$$

A simple worked-out example

- Suppose we have $\mathcal{X} = \{1, 2, 3\}$ and we want $\vec{\rho} = \left(\frac{1}{2}, \frac{1}{4}, \frac{1}{4}\right)$
- Choose C
- Compute A from M.-H. rule
- Compute final Q

$$C = \begin{bmatrix} 0 & \frac{1}{2} & 0 \\ 1 & 0 & 1 \\ 0 & \frac{1}{2} & 0 \end{bmatrix}$$



$$A = \begin{bmatrix} 0 & \min(1, 4) & 0 \\ \min\left(1, \frac{1}{4}\right) & 0 & \min\left(1, \frac{1}{2}\right) \\ 0 & \min(1, 2) & 0 \end{bmatrix}$$

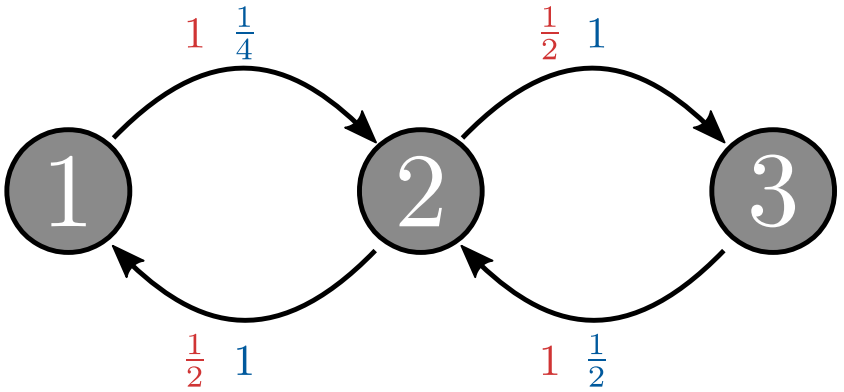
No symmetry in the proposals, we need to use the more general formula

$$A_{ij} = \min\left(1, \frac{C_{ji}\rho_i}{C_{ij}\rho_j}\right)$$

A simple worked-out example

- Suppose we have $\mathcal{X} = \{1, 2, 3\}$ and we want $\vec{\rho} = \left(\frac{1}{2}, \frac{1}{4}, \frac{1}{4}\right)$
- Choose C
- Compute A from M.-H. rule
- Compute final Q

$$C = \begin{bmatrix} 0 & \frac{1}{2} & 0 \\ 1 & 0 & 1 \\ 0 & \frac{1}{2} & 0 \end{bmatrix}$$



$$A = \begin{bmatrix} 0 & 1 & 0 \\ \frac{1}{4} & 0 & \frac{1}{2} \\ 0 & 1 & 0 \end{bmatrix}$$

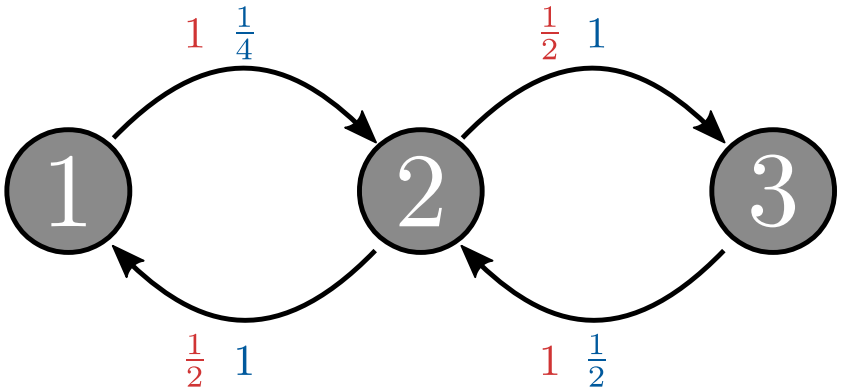
No symmetry in the proposals, we need to use the more general formula

$$A_{ij} = \min\left(1, \frac{C_{ji}\rho_i}{C_{ij}\rho_j}\right)$$

A simple worked-out example

- Suppose we have $\mathcal{X} = \{1, 2, 3\}$ and we want $\vec{\rho} = \left(\frac{1}{2}, \frac{1}{4}, \frac{1}{4}\right)$
- Choose C
- Compute A from M.-H. rule
- Compute final Q

$$C = \begin{bmatrix} 0 & \frac{1}{2} & 0 \\ 1 & 0 & 1 \\ 0 & \frac{1}{2} & 0 \end{bmatrix}$$



$$A = \begin{bmatrix} 0 & 1 & 0 \\ \frac{1}{4} & 0 & \frac{1}{2} \\ 0 & 1 & 0 \end{bmatrix}$$

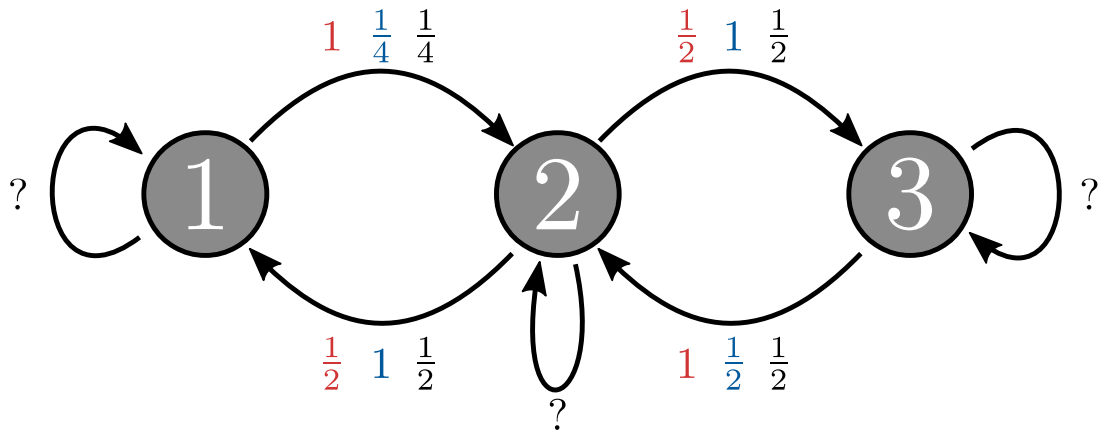
$$Q = \begin{bmatrix} ? & \frac{1}{2}1 & 0 \\ 1\frac{1}{4} & ? & 1\frac{1}{2} \\ 0 & \frac{1}{2}1 & ? \end{bmatrix}$$

Off-diagonals are simply products of proposal and acceptance.
Diagonal terms must be computed from normalization.

A simple worked-out example

- Suppose we have $\mathcal{X} = \{1, 2, 3\}$ and we want $\vec{\rho} = \left(\frac{1}{2}, \frac{1}{4}, \frac{1}{4}\right)$
- Choose C
- Compute A from M.-H. rule
- Compute final Q

$$C = \begin{bmatrix} 0 & \frac{1}{2} & 0 \\ 1 & 0 & 1 \\ 0 & \frac{1}{2} & 0 \end{bmatrix}$$



$$A = \begin{bmatrix} 0 & 1 & 0 \\ \frac{1}{4} & 0 & \frac{1}{2} \\ 0 & 1 & 0 \end{bmatrix}$$

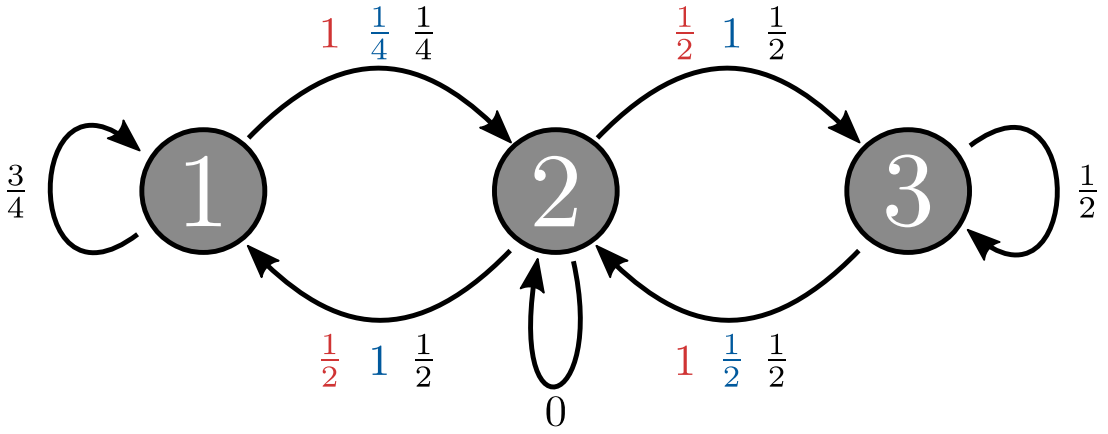
$$Q = \begin{bmatrix} ? & \frac{1}{2} & 0 \\ \frac{1}{4} & ? & \frac{1}{2} \\ 0 & \frac{1}{2} & ? \end{bmatrix}$$

Off-diagonals are simply products of proposal and acceptance.
Diagonal terms must be computed from normalization.

A simple worked-out example

- Suppose we have $\mathcal{X} = \{1, 2, 3\}$ and we want $\vec{\rho} = \left(\frac{1}{2}, \frac{1}{4}, \frac{1}{4}\right)$
- Choose C
- Compute A from M.-H. rule
- Compute final Q

$$C = \begin{bmatrix} 0 & \frac{1}{2} & 0 \\ 1 & 0 & 1 \\ 0 & \frac{1}{2} & 0 \end{bmatrix}$$



$$A = \begin{bmatrix} 0 & 1 & 0 \\ \frac{1}{4} & 0 & \frac{1}{2} \\ 0 & 1 & 0 \end{bmatrix}$$

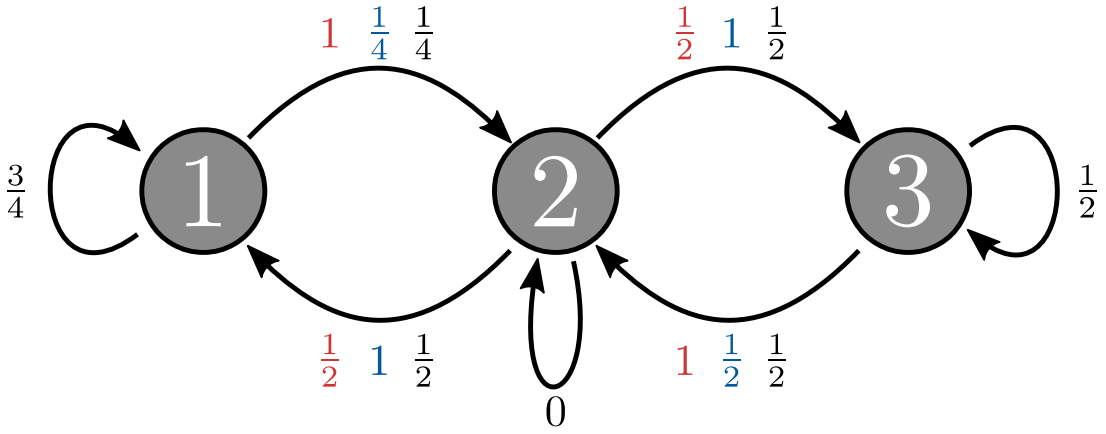
$$Q = \begin{bmatrix} \frac{3}{4} & \frac{1}{2} & 0 \\ \frac{1}{4} & 0 & \frac{1}{2} \\ 0 & \frac{1}{2} & \frac{1}{2} \end{bmatrix}$$

Off-diagonals are simply products of proposal and acceptance.
Diagonal terms must be computed from normalization.

A simple worked-out example

- Suppose we have $\mathcal{X} = \{1, 2, 3\}$ and we want $\vec{\rho} = \left(\frac{1}{2}, \frac{1}{4}, \frac{1}{4}\right)$
- Choose C
- Compute A from M.-H. rule
- Compute final Q

$$C = \begin{bmatrix} 0 & \frac{1}{2} & 0 \\ 1 & 0 & 1 \\ 0 & \frac{1}{2} & 0 \end{bmatrix}$$



$$A = \begin{bmatrix} 0 & 1 & 0 \\ \frac{1}{4} & 0 & \frac{1}{2} \\ 0 & 1 & 0 \end{bmatrix}$$

$$Q = \begin{bmatrix} \frac{3}{4} & \frac{1}{2} & 0 \\ \frac{1}{4} & 0 & \frac{1}{2} \\ 0 & \frac{1}{2} & \frac{1}{2} \end{bmatrix}$$

Done! You should check the stationarity condition $\vec{\rho} = Q\vec{\rho}$.
You should also write a code that simulates this process.