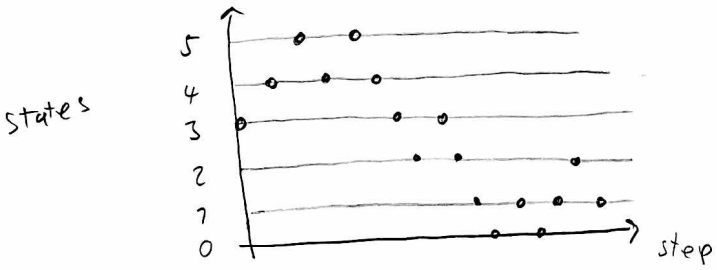
⊗ End lecture Oct 29.

Markov Chain (Andrey Markov)

- A random sequence in which the prob. for the next state only depends on the current state, not the history of states before that. (Markov property, memoryless)

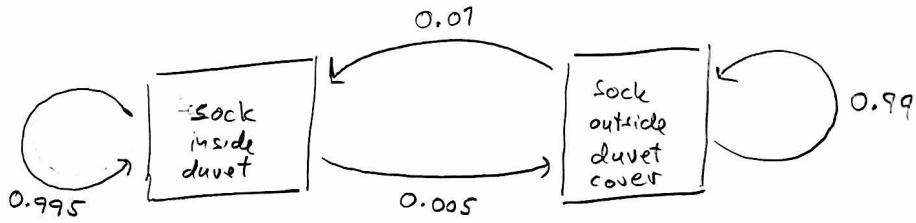
$[p(X_{i+1} | X_i, X_{i-1}, \ldots) = p(X_{i+1} | X_i)]$

Examples: "Drunkards walk": move +1 or -1 with equal prob.

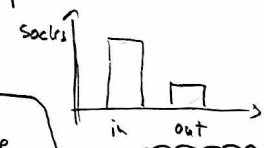


[Probs. for next number dep. only on current number, not how the chain got there.]

- "Socks in duvet cover"



- Distribute walkers across states and iterate → evolve towards stationary / equilib. populations across states

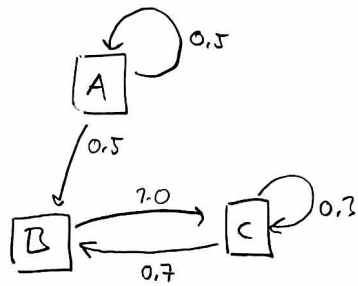


Start with 20 socks outside. Expect 13.3 socks inside!

- Important concept: Ergodicity

A chain is ergodic if any state can be reache from any other state with a finite number of steps

Non-ergodic chain:



Another example of Markov Chain: FHI's "agent-based models" for disease spread. Simulate a population of individuals.

Markov Chain Monte Carlo

A method that produces a Markov chain of samples / events / steps whose stationary distribution (long-run distribution) is a given prob. distribution

In other words: Generate a set of samples $x_i \sim p(x)$ by recording the steps of a Markov chain in x-space.

General procedure:

1) Current state x_i

2) Generate a new candidate x' using a proposal pdf that only depends on current state

3) Apply acceptance rule. If accepted: $x_{i+1} = x'$
   If rejected: $x_{i+1} = x_i$ ← Note: This is also a new sample!

4) Repeat

For the list of samples $\{x_1, x_2, x_3, \ldots\}$ to end up as a set of samples distributed according to $p(x)$, the acceptace rule should ensure

- Ergodicity (can visit all states)
- Detailed balance: $p(x_i)\,\text{Prob}(x_i \to x') = p(x')\,\text{Prob}(x' \to x_i)$

  can have: small large = large small

  This is where your chosen pdf appears!

- The Metropolis (or Metropolis-Hastings) algorithm is one example that ensures this [Basis: $\text{Prob}(x \to x') = T(x \to x') A(x \to x')$]

  - Generate candidate according to some <u>proposal prob.</u> $T(x_i \to x')$

  - Calculate <u>acceptance prob.</u>

$$A(x_i \to x') = \min\left(1, \frac{p(x')}{p(x_i)} \frac{T(x' \to x_i)}{T(x_i \to x')}\right)$$

If proposal is symmetric: $T(x' \to x_i) = T(x' \to x_i)$ (Metropolis algo.)

$$A(x_i \to x') = \min\left(1, \frac{p(x')}{p(x_i)}\right)$$

[See explanation on next page]

  - Generate random number r from $U(0,1)$ (Uniform dist on (0,1))
    - If $r \leq A(x_i \to x')$, then <u>accept</u>: $x_{i+1} = x'$
    - If $r > A(x_i \to x')$, then <u>reject</u>: $x_{i+1} = x_i$

- <u>Note:</u>
  - <u>Always</u> accepts jump to state with higher prob. ($p(x') > p(x)$)
  - <u>Sometimes</u> accepts jump to state with lower prob ($p(x') < p(x)$) (with prob. that ensures detailed balance.)

[Wait long enough $\Rightarrow$ all states should be explored (theoretically). Freedom in choosing proposal prob. dist.]

  - Only depends on <u>ratio</u> $\frac{p(x')}{p(x)}$, so normalization constants (like the partition function) <u>cancel</u>! [Very important in proj. 4!]

- Why $A(x_i \to x') = \min\left(1, \frac{p(x')}{p(x_i)}\right)$?

- Using our freedom to choose one of the A's. Detailed balance is just a requirement on the ratio

$$\frac{A(x_i \to x')}{A(x' \to x_i)} = \frac{p(x')}{p(x_i)} \qquad \left(\text{Let } T(x' \to x_i) = T(x_i \to x')\right)$$

- If $p(x') > p(x_i)$

$$\Rightarrow \frac{A(x_i \to x')}{A(x' \to x_i)} = \frac{p(x')}{p(x_i)} > 1$$

Choose $\boxed{A(x_i \to x') = 1}$

- If $p(x') < p(x_i)$

$$\Rightarrow \frac{A(x_i \to x')}{A(x' \to x_i)} = \frac{p(x')}{p(x_i)} < 1$$

Choose $A(x' \to x_i) = 1$

$$\Rightarrow \boxed{A(x_i \to x') = \frac{p(x')}{p(x_i)}} < 1$$

- In combination: $\boxed{A(x_i \to x') = \min\left(1, \frac{p(x')}{p(x_i)}\right)}$

- Setting $A(x_i \to x')$ this way
  - satisfies <u>detailed balance</u>
  - lets us move to a new state as <u>often as we can</u> (while being consistent with detailed balance)

- Suggested algorithm for Project 4 (Ising model):

Repeat N times = "one MC cycle"

due to strong correlations between subsequent samples in the state space.

- Generate candidate state $\bar{S}'$
  - Pick random spin from lattice
  - Flip it!

This is using a uniform distr. for $T(\bar{S}_i \to \bar{S}')$ for all "neighbour states" $\bar{S}'$ of $\bar{S}_i$

- Find value of $\frac{p(\bar{S}')}{p(\bar{S}_i)}$

In an efficient way! Don't evaluate $p(\bar{S}')$ and $p(\bar{S})$ separately, and don't evaluate exp(...) all the time!

- Generate random number $r \sim U(0,1)$

and carry out accept/reject step: accept if $r < \frac{p(\bar{S}')}{p(\bar{S}_i)}$

- Then use the new state $\bar{S}_{i+1}$ to compute energy, magnetization, expectation values after this number of cycles, etc.
- Store relevant numbers
- Repeat

Two views on the MCMC in Project 4

(1) As model of (discretized) time evolution of system

- Each time step there are $\sim N$ random spin–environment interactions that *can* flip the spin. So the Markov chain

$$\bar{S}_0 \to \bar{S}_1 \to \bar{S}_2 \to \bar{S}_3 \to \dots$$

effectively means

$$\bar{S}(t_0) \to \bar{S}(t_1) \to \bar{S}(t_2) \to \bar{S}(t_3) \to \dots$$

- If system starts in state $\bar{S}_0$ that has *low prob.* under equilibrium assumption (Boltzmann), then letting it evolve will "equilibrate the system", i.e. take $\bar{S}$ into regions of state space that are more likely in equilibrium.

  ↳ Connects to MCMC "*burn-in*"

  ↳ Some unfortunate terminology in the literature, which makes it sound like $p(\bar{S})$ is changing, when it's just the chain of $\bar{S}$ samples that is converging towards a good approx to $p(\bar{S})$.

(2) As a method for sampling from $p(\bar{S})$

- Just some way to draw a set of samples $\bar{S}_1, \bar{S}_2, \bar{S}_3, \dots$ such that the distribution of samples (in the long run) corresponds to $p(\bar{S})$.

- Not directly related to the physics — equally applicable for any pdf, regardless of topic.

- The physics of equilibrium was already contained in our choice to use the Boltzmann dist. for $p(\bar{S})$

Burn-in

- Samples in MCMC are <u>correlated</u>, i.e. not "independent and identically dist. draws" (iid)

$$p(x_{i+1} | x_i) \neq p(x_{i+1})$$

- In the limit $n \to \infty$ (n: number of samples), we will have $\frac{n_{x \in [x, x+dx]}}{n} = p(x)dx$

- But we cannot do $n \to \infty$ in practice!

  $\Rightarrow$ Our resulting distribution of samples (and hence our estimated expectation values) depends on starting point, which can be <u>unreasonably improbable</u> <u>compared to the number of samples we draw</u>!

- Usual solution:
  - <u>Throw away</u> first part of a chain of samples (burn-in)
  - Not use every new sample (even after burn in)
  - Run <u>many chains</u> from different starting points and combine results.