STA 414/2104: Statistical Methods for Machine Learning II

Week 5 - 1/2: MCMC

Michal Malyska

University of Toronto

Overview

- Markov chains

Overview

- Markov chains
- Metropolis-Hastings

Overview

- Markov chains
- Metropolis-Hastings
- Markov chain Monte Carlo

Overview

- Markov chains
- Metropolis-Hastings
- Markov chain Monte Carlo
- Assignment 2 to be released today.

Sequential data

So far, we only considered methods in which the samples we generate are i.i.d:

- We generated T samples $x_{1:T} = \{x_1, ..., x_T\}$.

Sequential data

So far, we only considered methods in which the samples we generate are i.i.d:

- We generated T samples $x_{1:T} = \{x_1, ..., x_T\}$.
- But each sample was independent from each other

$$x_t \sim p(x) \;\; \text{i.i.d.}$$

Sequential data

So far, we only considered methods in which the samples we generate are i.i.d:

- We generated T samples $x_{1:T} = \{x_1, ..., x_T\}$.
- But each sample was independent from each other

$$x_t \sim p(x) \quad \text{i.i.d.}$$

- This lecture, we will generate samples that are dependent.

Sequential data

This also comes up when modelling the data: We generally assume data was i.i.d, however this may be a poor assumption:

- Sequential data is common in time-series modelling (e.g. stock prices, speech, video analysis) or ordered (e.g. textual data, gene sequences).

Sequential data

This also comes up when modelling the data: We generally assume data was i.i.d, however this may be a poor assumption:

- Sequential data is common in time-series modelling (e.g. stock prices, speech, video analysis) or ordered (e.g. textual data, gene sequences).
- Recall the general joint factorization via the chain rule

$$p(x_{1:T}) = \prod_{t=1}^{T} p(x_t | x_{t-1}, ..., x_1) \text{ where } p(x_1|x_0) = p(x_1).$$

Sequential data

This also comes up when modelling the data: We generally assume data was i.i.d, however this may be a poor assumption:

- Sequential data is common in time-series modelling (e.g. stock prices, speech, video analysis) or ordered (e.g. textual data, gene sequences).
- Recall the general joint factorization via the chain rule

$$p(x_{1:T}) = \prod_{t=1}^{T} p(x_t | x_{t-1}, ..., x_1) \text{ where } p(x_1|x_0) = p(x_1).$$

- But this quickly becomes intractable for high-dimensional data -each factor requires exponentially many parameters to specify as a function of T.

Sequential data

This also comes up when modelling the data: We generally assume data was i.i.d, however this may be a poor assumption:

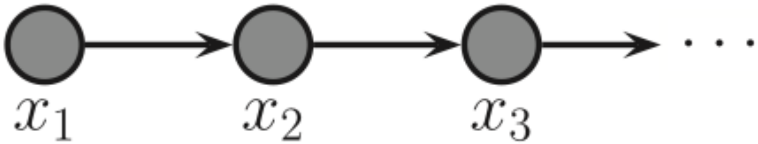
- Sequential data is common in time-series modelling (e.g. stock prices, speech, video analysis) or ordered (e.g. textual data, gene sequences).
- Recall the general joint factorization via the chain rule

$$p(x_{1:T}) = \prod_{t=1}^{T} p(x_t | x_{t-1}, ..., x_1) \text{ where } p(x_1|x_0) = p(x_1).$$

- But this quickly becomes intractable for high-dimensional data -each factor requires exponentially many parameters to specify as a function of T.
- So we make the simplifying assumption that our data can be modeled as a **first-order Markov chain**

$$p(x_t | x_{1:t-1}) = p(x_t | x_{t-1})$$

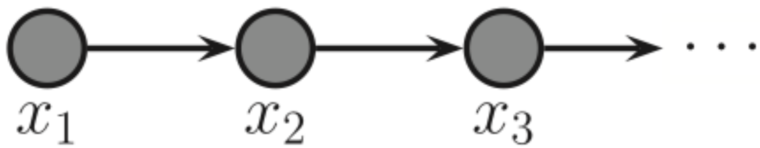
Markov chains



- We make the simplifying **first-order Markov chain** assumption:

$$p(x_t|x_{1:t-1}) = p(x_t|x_{t-1})$$

Markov chains



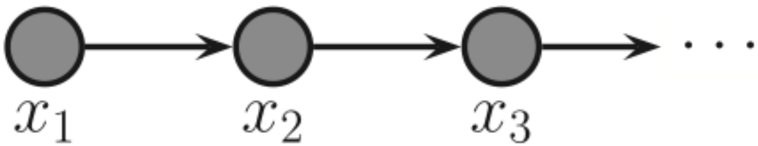
- We make the simplifying **first-order Markov chain** assumption:

$$p(x_t|x_{1:t-1}) = p(x_t|x_{t-1})$$

- This assumption greatly simplifies the factors in the joint distribution

$$p(x_{1:T}) = \prod_{t=1}^{T} p(x_t|x_{t-1})$$

$P(x_1 | x_0) = P(x_1)$



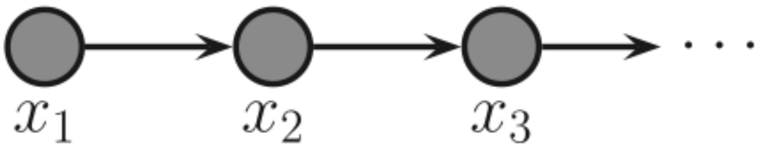
- A useful distinction to make at this point is between stationary and non-stationary distributions that generate our data
  - **Stationary Markov chain**: the distribution generating the data does not change through time:
    $p(x_{t+1} = y | x_t = x) = p(x_{t+2} = y | x_{t+1} = x)$

$x_t = x$

$p(x_{t+1} = y) | x_t = x$

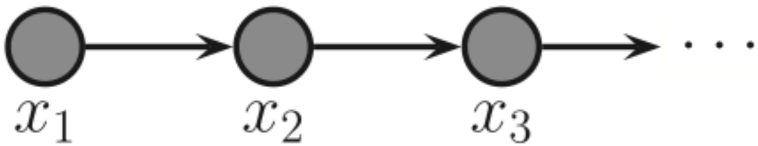
$= p(x_{t+2} = y) | x_{t+1} = x$

Markov chains



- A useful distinction to make at this point is between stationary and non-stationary distributions that generate our data
  - **Stationary Markov chain**: the distribution generating the data does not change through time: $p(x_{t+1} = y|x_t = x) = p(x_{t+2} = y|x_{t+1} = x)$
  - **Non-stationary Markov chain**: the distribution generating the data is a function of time: The transition probabilities $p(x_{t+1} = y|x_t = x)$ depend on the time t.

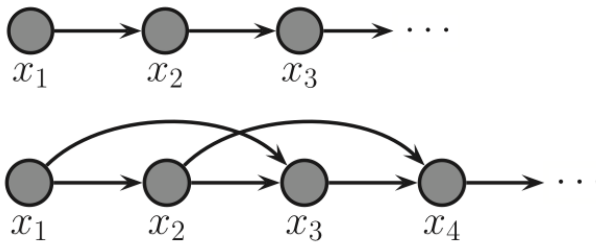
Markov chains



- A useful distinction to make at this point is between stationary and non-stationary distributions that generate our data
  - **Stationary Markov chain**: the distribution generating the data does not change through time: $p(x_{t+1} = y | x_t = x) = p(x_{t+2} = y | x_{t+1} = x)$
  - **Non-stationary Markov chain**: the distribution generating the data is a function of time: The transition probabilities $p(x_{t+1} = y | x_t = x)$ depend on the time t.

We only consider stationary Markov chains, (aka homogenous MCs).

Higher-order Markov chains



In some cases, the first-order assumption may be restrictive (such as when modeling natural language, where long-term dependencies occur often). We can generalize to high-order dependence trivially

- Second order:

$$p(x_t|x_{1:t-1}) = p(x_t|x_{t-1}, x_{t-2})$$

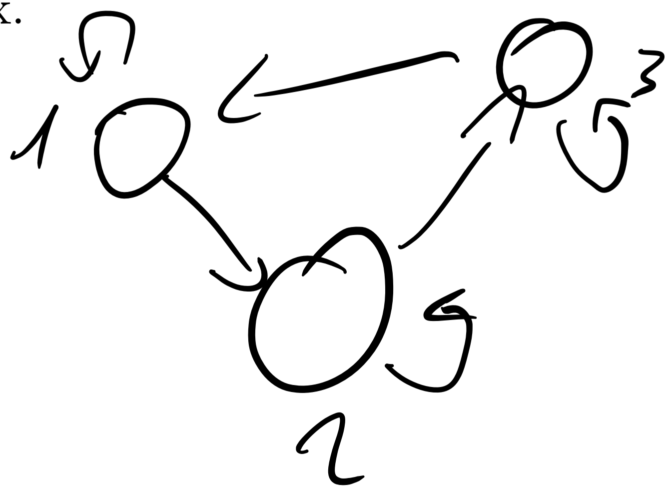
- m-th-order

$$p(x_t|x_{1:t-1}) = p(x_t|x_{t-1:t-m})$$

Transition matrix

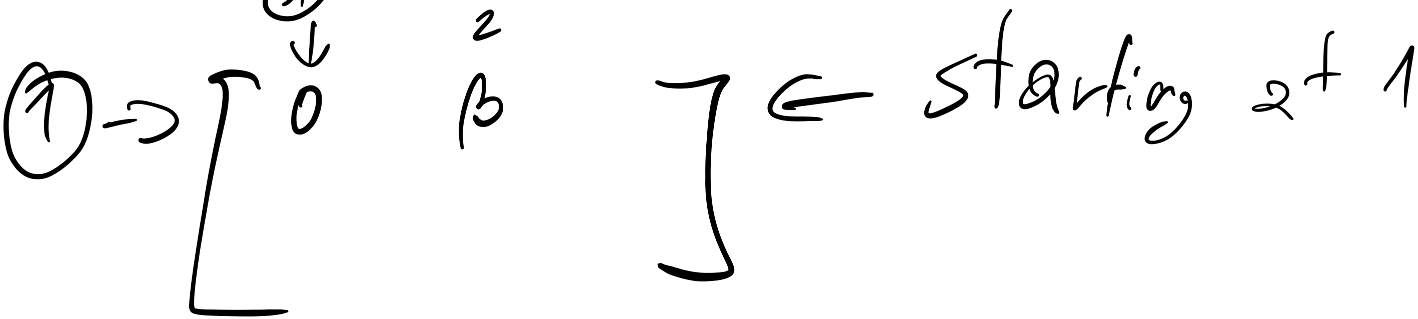
$K = 3$

- When x_t is discrete (e.g. $x_t \in \{1, ..., K\}$ which is called state space), the conditional distribution $p(x_t | x_{t-1})$ can be written as a $K \times K$ matrix.



Transition matrix

- When x_t is discrete (e.g. $x_t \in \{1, ..., K\}$ which is called state space), the conditional distribution $p(x_t | x_{t-1})$ can be written as a $K \times K$ matrix.
- We call this the transition matrix A: $A_{ij} = p(x_t = j | x_{t-1} = i)$, the probability of going from state i to state j.



Transition matrix

- When x_t is discrete (e.g. $x_t \in \{1, ..., K\}$ which is called state space), the conditional distribution $p(x_t|x_{t-1})$ can be written as a $K \times K$ matrix.
- We call this the transition matrix A: $A_{ij} = p(x_t = j|x_{t-1} = i)$, the probability of going from state i to state j.
- Notice

$$
\begin{aligned}
p(x_t = j) &= \sum_i p(x_t = j|x_{t-1} = i) p(x_{t-1} = i), \\
&= \sum_i A_{ij} p(x_{t-1} = i).
\end{aligned}
$$

Transition matrix

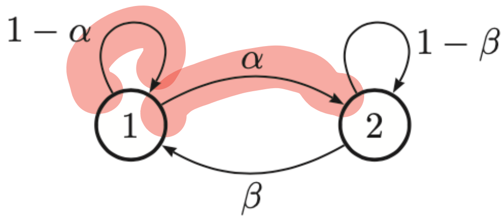
- When x_t is discrete (e.g. $x_t \in \{1, ..., K\}$ which is called state space), the conditional distribution $p(x_t|x_{t-1})$ can be written as a $K \times K$ matrix.
- We call this the transition matrix A: $A_{ij} = p(x_t = j|x_{t-1} = i)$, the probability of going from state i to state j.
- Notice

$$
\begin{aligned}
p(x_t = j) &= \sum_i p(x_t = j|x_{t-1} = i)p(x_{t-1} = i), \\
&= \sum_i A_{ij} p(x_{t-1} = i).
\end{aligned}
$$

- Each row of the matrix sums to one, $\sum_j A_{ij} = 1$, so this is called a stochastic matrix.

Transition matrix

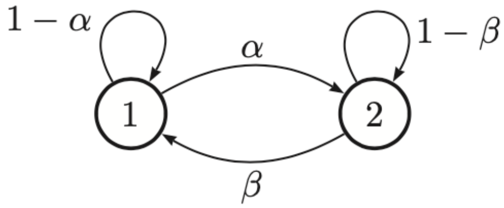
- The transition matrix A: $A_{ij} = p(x_t = j | x_{t-1} = i)$ is the probability of going from state i to state j.



  - We can visualize Markov chains via a directed graph, where nodes represent states and arrows represent legal transitions, i.e., non-zero elements of A.

Transition matrix

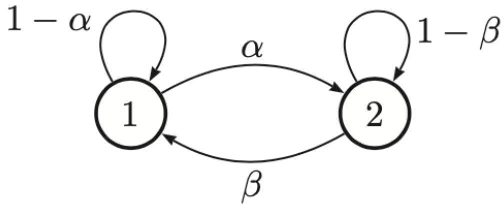
- The transition matrix A: $A_{ij} = p(x_t = j | x_{t-1} = i)$ is the probability of going from state i to state j.



  - We can visualize Markov chains via a directed graph, where nodes represent states and arrows represent legal transitions, i.e., non-zero elements of A.
  - This is known as a state transition diagram.

Transition matrix

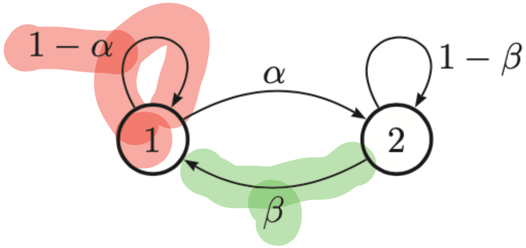
- The transition matrix A: $A_{ij} = p(x_t = j | x_{t-1} = i)$ is the probability of going from state i to state j.



  - We can visualize Markov chains via a directed graph, where nodes represent states and arrows represent legal transitions, i.e., non-zero elements of A.
  - This is known as a state transition diagram.

Transition matrix

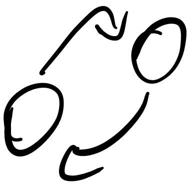
- The transition matrix A: $A_{ij} = p(x_t = j | x_{t-1} = i)$ is the probability of going from state i to state j.

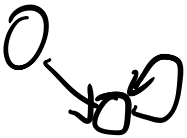


  - We can visualize Markov chains via a directed graph, where nodes represent states and arrows represent legal transitions, i.e., non-zero elements of A.
  - This is known as a state transition diagram.

- The weights associated with the arcs are the probabilities.
- For example, the trainsition matrix for the 2-state chain shown above is given by

$$A = \begin{bmatrix} 1-\alpha & \alpha \\ \beta & 1-\beta \end{bmatrix}$$





Chapman-Kolmogorov equations

- The n-step transition matrix $A(n)$ is defined as

$$A_{ij}(n) = p(x_{t+n} = j | x_t = i)$$

which is the probability of getting from i to j in exactly n steps.

Chapman-Kolmogorov equations

- The n-step transition matrix $A(n)$ is defined as

$$A_{ij}(n) = p(x_{t+n} = j | x_t = i)$$

  which is the probability of getting from i to j in exactly n steps.
- Notice that $A(1) = A$.

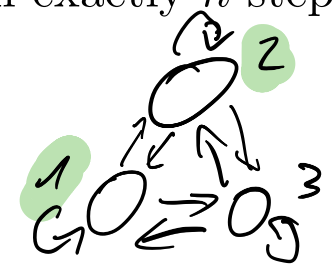
Chapman-Kolmogorov equations

- The n-step transition matrix $A(n)$ is defined as

$$A_{ij}(n) = p(x_{t+n} = j | x_t = i)$$

which is the probability of getting from i to j in exactly n steps.

- Notice that $A(1) = A$.
- **Chapman-Kolmogorov** equations state that



$$A_{ij}(m+n) = \sum_{k=1}^{K} A_{ik}(m) A_{kj}(n) \quad \text{equivalently} \quad A(m+n) = A(m)A(n)$$

the probability of getting from i to j in $m+n$ steps is just the probability of getting from i to k in m steps, and then from k to j in n steps, summed up over all k.

Chapman-Kolmogorov equations

- The n-step transition matrix $A(n)$ is defined as

$$A_{ij}(n) = p(x_{t+n} = j | x_t = i)$$

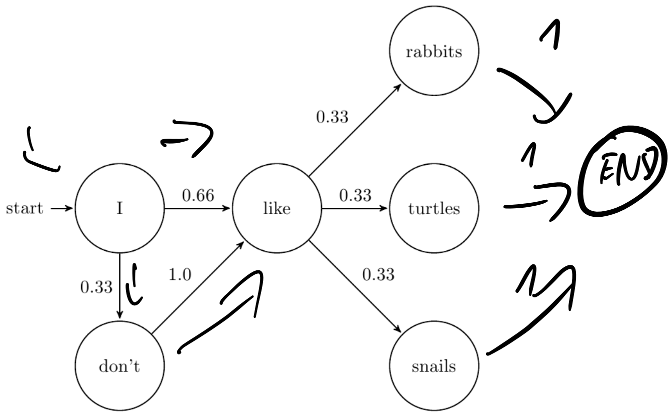
  which is the probability of getting from i to j in exactly n steps.
- Notice that $A(1) = A$.
- **Chapman-Kolmogorov** equations state that

$$A_{ij}(m+n) = \sum_{k=1}^{K} A_{ik}(m) A_{kj}(n) \text{ equivalently } A(m+n) = A(m)A(n)$$

  the probability of getting from i to j in $m + n$ steps is just the probability of getting from i to k in m steps, and then from k to j in n steps, summed up over all k.
- So $A(n) = A \times A(n-1) = A \times A \times A(n-2) = \cdots = A^n$.

Application: Markov Language Models

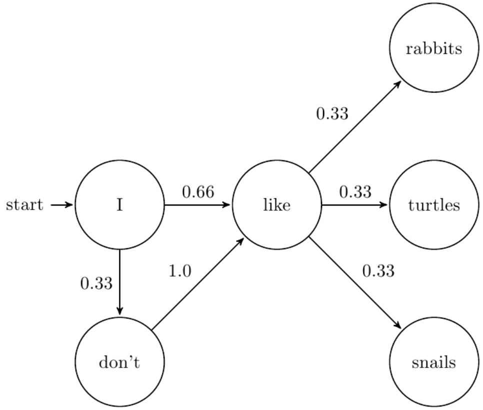
- We could use Markov chains as language models, which are distributions over sequences of words.
- State space is all words and x_t denotes the t-th word in a sentence.
- We use a first-order Markov model, then $p(x_t = k | x_{t-1} = j)$.



I don't like turtles

Application: Markov Language Models

- We could use Markov chains as language models, which are distributions over sequences of words.
- State space is all words and x_t denotes the t-th word in a sentence.
- We use a first-order Markov model, then $p(x_t = k | x_{t-1} = j)$.
  - We estimate the transition matrix A. The probability of any particular sentence of length T



$$\begin{aligned} p(x_{1:T}|\theta) =& \pi(x_1)A(x_1, x_2) \cdots A(x_{T-1}, x_T) \\ =& \prod_{j=1}^{K} \pi_j^{1[x_1=j]} \prod_{t=2}^{T} \prod_{j=1}^{K} \prod_{k=1}^{K} A_{jk}^{1[x_t=k, x_{t-1}=j]} \end{aligned}$$

where $\pi(x_1)$ is the probability of the sentence starting with word x_1.

Application: Markov Language Models

- We use MLE to estimate A from data $\mathcal{D} = \{x^{(1)}, ..., x^{(N)}\}$.
- Likelihood of any particular sentence $x_{1:T}$ of length T

$$p(x_{1:T}|\theta) = \prod_{j=1}^{K} \pi_j^{1[x_1=j]} \prod_{t=2}^{T} \prod_{j=1}^{K} \prod_{k=1}^{K} A_{jk}^{1[x_t=k, x_{t-1}=j]}$$

Application: Markov Language Models

- We use MLE to estimate A from data $\mathcal{D} = \{x^{(1)}, ..., x^{(N)}\}$.
- Likelihood of any particular sentence $x_{1:T}$ of length T

$$p(x_{1:T}|\theta) = \prod_{j=1}^{K} \pi_j^{1[x_1=j]} \prod_{t=2}^{T} \prod_{j=1}^{K} \prod_{k=1}^{K} A_{jk}^{1[x_t=k, x_{t-1}=j]}$$

- Log-likelihood of a sentence $x^{(i)} = (x_1^{(i)}, ..., x_{T_i}^{(i)})$

$$\log p(\mathcal{D}|\theta) = \sum_{i=1}^{N} \log p(x^{(i)}|\theta) = \sum_j N_j^1 \log \pi_j + \sum_j \sum_k N_{jk} \log A_{jk}$$

where we define the counts

$$N_j^1 = \sum_{i=1}^{N} 1[x_{i1} = j], \qquad N_{jk} = \sum_{i=1}^{N} \sum_{t=1}^{T_i - 1} 1[x_{i,t} = j, x_{i,t+1} = k].$$

Application: Markov Language Models

- We use MLE to estimate A from data $\mathcal{D} = \{x^{(1)}, ..., x^{(N)}\}$.
- Likelihood of any particular sentence $x_{1:T}$ of length T

$$p(x_{1:T}|\theta) = \prod_{j=1}^{K} \pi_j^{1[x_1=j]} \prod_{t=2}^{T} \prod_{j=1}^{K} \prod_{k=1}^{K} A_{jk}^{1[x_t=k, x_{t-1}=j]}$$

- Log-likelihood of a sentence $x^{(i)} = (x_1^{(i)}, ..., x_{T_i}^{(i)})$

$$\log p(\mathcal{D}|\theta) = \sum_{i=1}^{N} \log p(x^{(i)}|\theta) = \sum_j N_j^1 \log \pi_j + \sum_j \sum_k N_{jk} \log A_{jk}$$

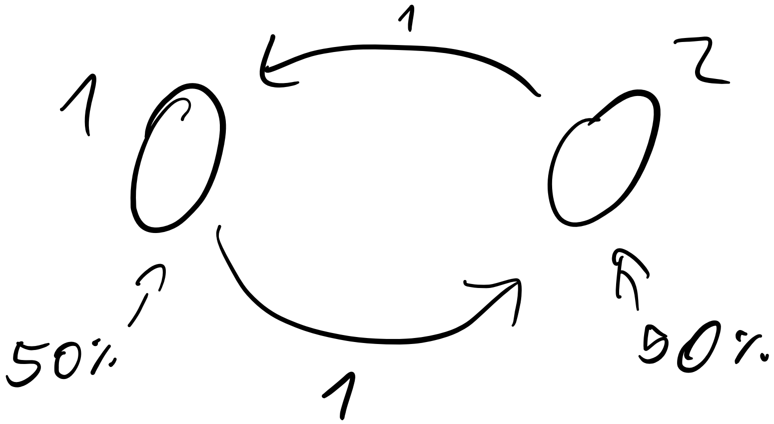
where we define the counts

$$N_j^1 = \sum_{i=1}^{N} 1[x_{i1} = j], \qquad N_{jk} = \sum_{i=1}^{N} \sum_{t=1}^{T_i - 1} 1[x_{i,t} = j, x_{i,t+1} = k].$$

- The MLE is given as $\hat{\pi}_j = \frac{N_j^1}{\sum_j N_j^1}$ $\quad \hat{A}_{jk} = \frac{N_{jk}}{\sum_k N_{jk}}$.

Stationary distribution of a Markov chain

- We are often interested in the long term distribution over states, which is known as the stationary distribution of the chain.



Stationary distribution of a Markov chain

- We are often interested in the long term distribution over states, which is known as the stationary distribution of the chain.
- Let A be the transition matrix, e.g. $p(x_{t+1} = j | x_t = i) = A_{ij}$ and $\pi_t(j) = p(x_t = j)$ be the probability of being in state j at time t. Thus the initial distribution is given by π_0 and

$$\pi_1(j) = \sum_i \pi_0(i) A_{ij}.$$

$\pi_0(x)$

Stationary distribution of a Markov chain

- We are often interested in the long term distribution over states, which is known as the stationary distribution of the chain.
- Let A be the transition matrix, e.g. $p(x_{t+1} = j | x_t = i) = A_{ij}$ and $\pi_t(j) = p(x_t = j)$ be the probability of being in state j at time t. Thus the initial distribution is given by π_0 and

$$\pi_1(j) = \sum_i \pi_0(i) A_{ij}.$$

- Assume that π_t is a row vector with entries $\pi_t(j)$. This vector is the distribution of x_t, e.g. $p(x_t = j) = \pi_t(j)$.

$$\pi_1 = \pi_0 A \quad \text{or more generally} \quad \pi_t = \pi_0 A^t.$$

$A(n) = A^n$

Stationary distribution of a Markov chain

- We are often interested in the long term distribution over states, which is known as the stationary distribution of the chain.
- Let A be the transition matrix, e.g. $p(x_{t+1} = j | x_t = i) = A_{ij}$ and $\pi_t(j) = p(x_t = j)$ be the probability of being in state j at time t. Thus the initial distribution is given by π_0 and

$$\pi_1(j) = \sum_i \pi_0(i) A_{ij}.$$

- Assume that π_t is a row vector with entries $\pi_t(j)$. This vector is the distribution of x_t, e.g. $p(x_t = j) = \pi_t(j)$.

$$\pi_1 = \pi_0 A \quad \text{or more generally} \quad \pi_t = \pi_0 A^t.$$

- Do this infinitely many steps, the distribution of x_t may converge

$$\pi = \pi A.$$

then we have reached the stationary distribution (aka the invariant distribution) of the Markov chain.

Stationary distribution

- We can find the stationary distribution of a Markov chain by solving the eigenvector equation

$$A^T v = v \quad \text{and set} \quad \pi = v^T.$$

v is the eigenvector of A^T with eigenvalue 1.

Stationary distribution

- We can find the stationary distribution of a Markov chain by solving the eigenvector equation

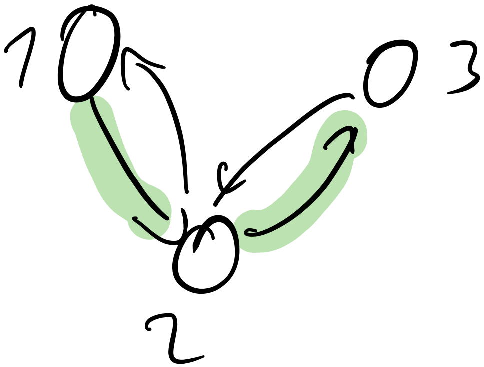
$$A^T v = v \quad \text{and set} \quad \pi = v^T.$$

v is the eigenvector of A^T with eigenvalue 1.

- Need to normalize!

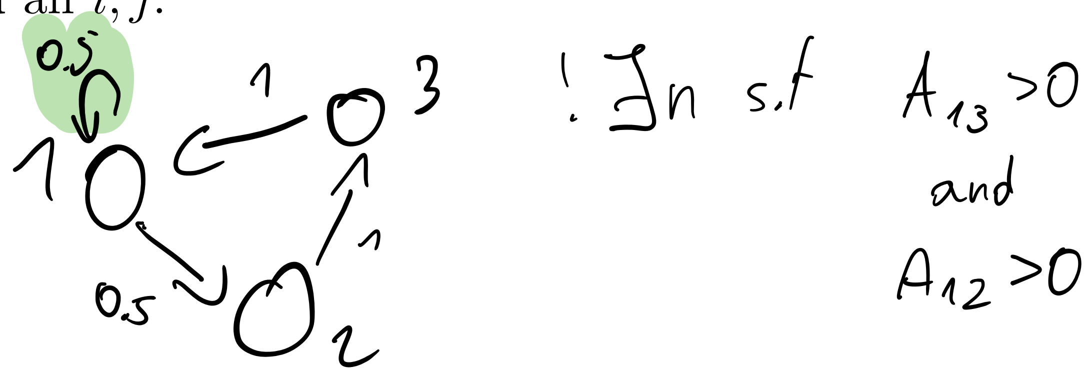
Detailed balance equations

- A MC is called **irreducible** if we can get from any state to any other state.



Detailed balance equations

- A MC is called **irreducible** if we can get from any state to any other state.
- A MC is called **regular** if the transition matrix satisfies $A^n_{ij} > 0$ for some n and all i, j.



Detailed balance equations

- A MC is called **irreducible** if we can get from any state to any other state.
- A MC is called **regular** if the transition matrix satisfies $A_{ij}^n > 0$ for some n and all i, j.
- A MC is **time reversible** if there exists a distribution π such that

$$\pi_i A_{ij} = \pi_j A_{ji}$$

This is called the detailed balance equations.

Detailed balance equations

- A MC is called **irreducible** if we can get from any state to any other state.
- A MC is called **regular** if the transition matrix satisfies $A^n_{ij} > 0$ for some n and all i, j.
- A MC is **time reversible** if there exists a distribution π such that

$$\pi_i A_{ij} = \pi_j A_{ji}$$

  This is called the detailed balance equations.

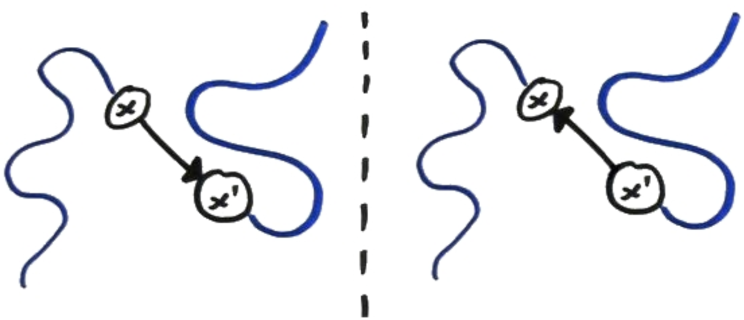
Detailed balance equations

- A MC is called **irreducible** if we can get from any state to any other state.
- A MC is called **regular** if the transition matrix satisfies $A_{ij}^n > 0$ for some n and all i, j.
- A MC is **time reversible** if there exists a distribution π such that

$$\pi_i A_{ij} = \pi_j A_{ji}$$

This is called the detailed balance equations.

Detailed balance means $\rightarrow x \rightarrow x'$ and $\rightarrow x' \rightarrow x$ are equally probable:



Detailed balance equations

Theorem

If a Markov chain with transition matrix A is regular and satisfies detailed balance wrt distribution π, then π is a stationary distribution.

Detailed balance equations

Theorem

If a Markov chain with transition matrix A is regular and satisfies detailed balance wrt distribution π, then π is a stationary distribution.

Proof:

$$\sum_i \pi_i A_{ij} = \sum_i \pi_j A_{ji} = \pi_j \sum_i A_{ji} = \pi_j \implies \pi = \pi A.$$

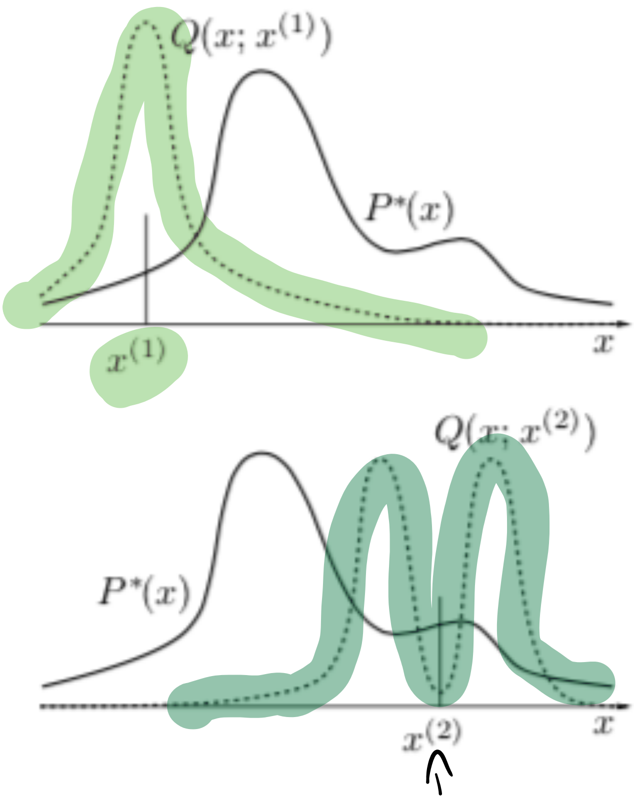
Metropolis-Hastings

well

Importance and rejection sampling work only if the proposal density $q(x)$ is similar to $p(x)$. In high dimensions, it is hard to find one such q.

Metropolis-Hastings

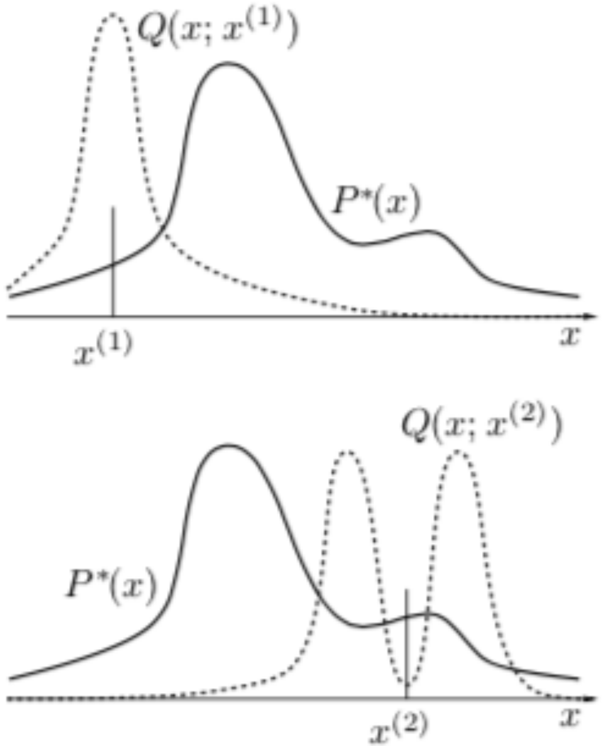
Importance and rejection sampling work only if the proposal density $q(x)$ is similar to $p(x)$. In high dimensions, it is hard to find one such q.



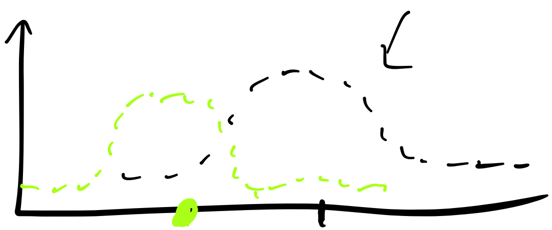
- The Metropolis–Hastings algorithm instead makes use of a proposal density q which depends on the current state $x^{(t)}$.

Metropolis-Hastings

Importance and rejection sampling work only if the proposal density $q(x)$ is similar to $p(x)$. In high dimensions, it is hard to find one such q.

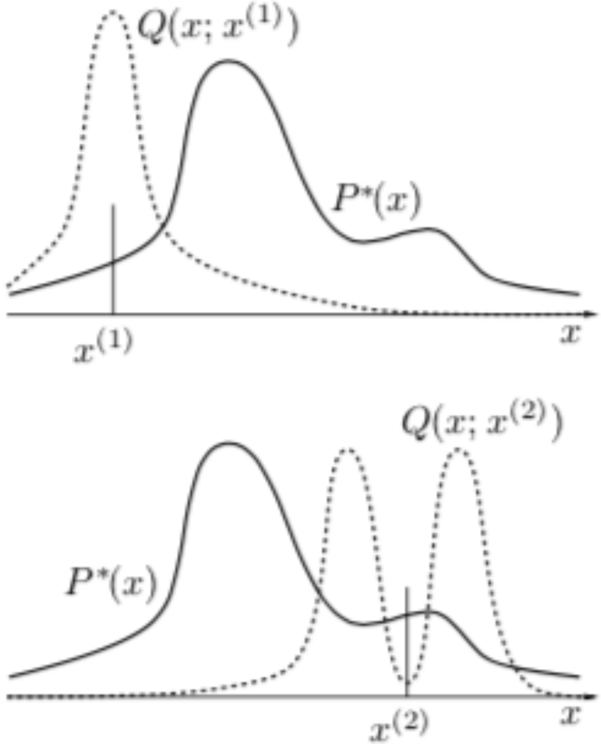


- The Metropolis–Hastings algorithm instead makes use of a proposal density q which depends on the current state $x^{(t)}$.
- The density $q(x'|x^{(t)})$ might be a simple distribution such as a Gaussian centered on the current $x^{(t)}$, but can be any density from which we can draw samples.



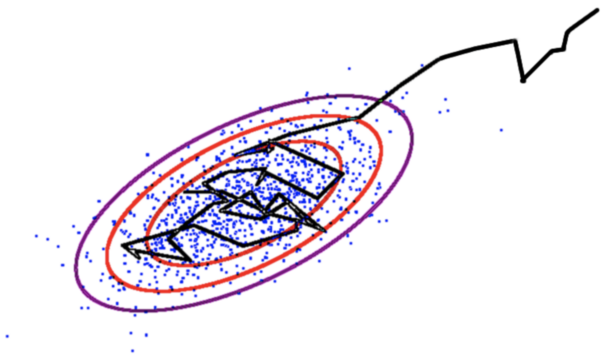
Metropolis-Hastings

Importance and rejection sampling work only if the proposal density $q(x)$ is similar to $p(x)$. In high dimensions, it is hard to find one such q.



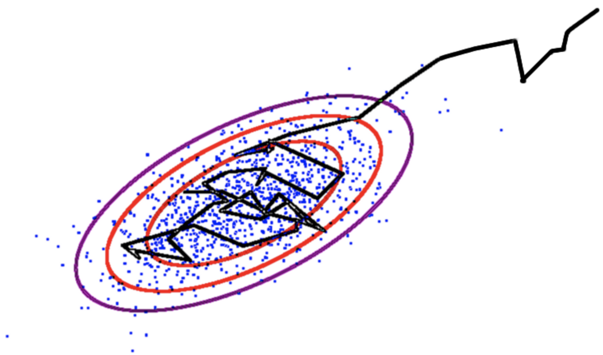
- The Metropolis–Hastings algorithm instead makes use of a proposal density q which depends on the current state $x^{(t)}$.
- The density $q(x'|x^{(t)})$ might be a simple distribution such as a Gaussian centered on the current $x^{(t)}$, but can be any density from which we can draw samples.
- In contrast to importance and rejection sampling, it is not necessary $q(x'|x^{(t)})$ to look at all similar to $p(x)$.

MCMC



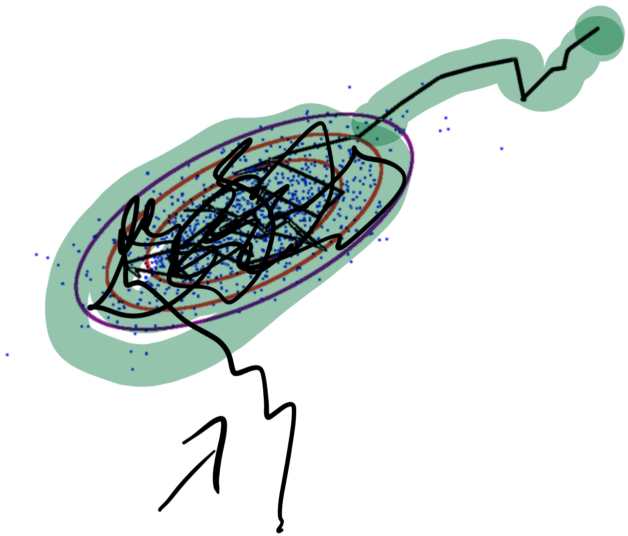
- In contrast to rejection sampling, where the accepted points $\{x^{(t)}\}$ are independent, MCMC methods generate a dependent sequence.

MCMC



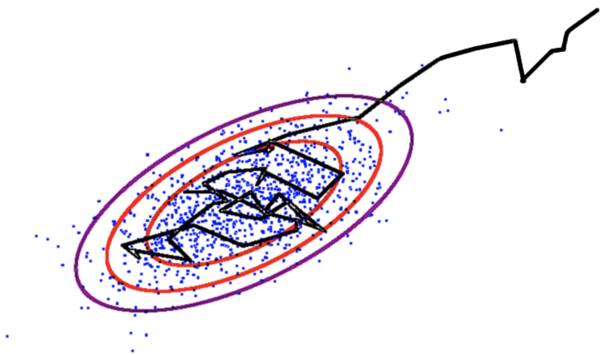
- In contrast to rejection sampling, where the accepted points $\{x^{(t)}\}$ are independent, MCMC methods generate a dependent sequence.
- Each sample $x^{(t)}$ has a probability distribution that depends on the previous value, $x^{(t-1)}$.

MCMC



- In contrast to rejection sampling, where the accepted points $\{x^{(t)}\}$ are independent, MCMC methods generate a dependent sequence.
- Each sample $x^{(t)}$ has a probability distribution that depends on the previous value, $x^{(t-1)}$.
- MCMC methods need to be run for a time in order to generate samples that are from the target distribution p.

MCMC



- In contrast to rejection sampling, where the accepted points $\{x^{(t)}\}$ are independent, MCMC methods generate a dependent sequence.
- Each sample $x^{(t)}$ has a probability distribution that depends on the previous value, $x^{(t-1)}$.
- MCMC methods need to be run for a time in order to generate samples that are from the target distribution p.

We can still do Monte Carlo estimaton for large enough T to estimate the mean of a test function ϕ:

$$\mathbb{E}_{x\sim p}[f(x)] \approx \frac{1}{T}\sum_{t=1}^{T} f(x^{(t)}).$$

Metropolis-Hastings algorithm

As before, we assume we can evaluate $\tilde{p}(x)$ for any x. The procedure is as follows:

- A tentative new state x' is generated from the proposal density $q(x'|x^{(t)})$. To decide whether to accept the new state, we compute

$$a = \frac{\tilde{p}(x')q(x^{(t)}|x')}{\tilde{p}(x^{(t)})q(x'|x^{(t)})}$$

  - If $a \geq 1$ then the new state is accepted.
  - Otherwise, the new state is accepted with probability a.
  - If accepted, set $x^{(t+1)} = x'$. Otherwise, set $x^{(t+1)} = x^{(t)}$.

Metropolis-Hastings algorithm

As before, we assume we can evaluate $\tilde{p}(x)$ for any x. The procedure is as follows:

- A tentative new state x' is generated from the proposal density $q(x'|x^{(t)})$. To decide whether to accept the new state, we compute

$$a = \frac{\tilde{p}(x')q(x^{(t)}|x')}{\tilde{p}(x^{(t)})q(x'|x^{(t)})}$$

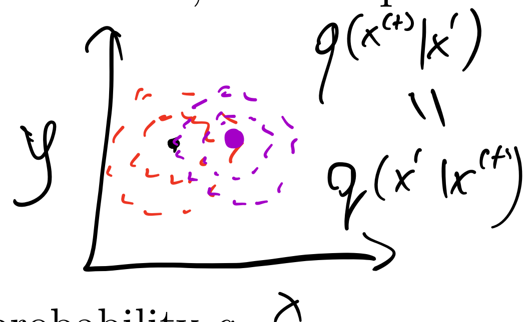
    - If $a \geq 1$ then the new state is accepted.
    - Otherwise, the new state is accepted with probability a.
    - If accepted, set $x^{(t+1)} = x'$. Otherwise, set $x^{(t+1)} = x^{(t)}$.
- This is a Markov chain with stationary distribution $\pi(x)$ is chosen to be the target distribution $p(x)$.

Metropolis-Hastings algorithm

As before, we assume we can evaluate $\tilde{p}(x)$ for any x. The procedure is as follows:

- A tentative new state x' is generated from the proposal density $q(x'|x^{(t)})$. To decide whether to accept the new state, we compute

$$a = \frac{\tilde{p}(x')q(x^{(t)}|x')}{\tilde{p}(x^{(t)})q(x'|x^{(t)})}$$



  - If $a \geq 1$ then the new state is accepted.
  - Otherwise, the new state is accepted with probability a.
  - If accepted, set $x^{(t+1)} = x'$. Otherwise, set $x^{(t+1)} = x^{(t)}$.
- This is a Markov chain with stationary distribution $\pi(x)$ is chosen to be the target distribution $p(x)$.
- The derivation of the algorithm starts with the condition of detailed balance.

Summary

- To sample from a distribution, we can design a Markov chain with its invariance distribution as the target (aka MCMC).

Summary

- To sample from a distribution, we can design a Markov chain with its invariance distribution as the target (aka MCMC).
- Metropolis-Hastings (MH) method can sample from high-dimensional targets.