Markov Chain Monte Carlo

Robert Winslow

GalvanizeU

June 22th, 2017

Review

Lab questions?

Learning objectives

- Know why we need to use MCMC – how do naive Monte Carlo and Importance Sampling fail? (Lecture)
- Understand how to compute the stationary distribution of a Markov Chain. (Lecture + Lab)
- Conceptually understand why Markov chains need to be ergodic for use in MCMC. (Lecture + Lab)
- Understand the Metropolis MCMC algorithm. (Lecture + Lab)
- Know what burn-in is and why it happens. (Lab)

Context

"The Best of the 20th Century: Editors Name Top 10 Algorithms

"1946: John von Neumann, Stan Ulam, and Nick Metropolis, all at the Los Alamos Scientific Laboratory, cook up the Metropolis algorithm, also known as the Monte Carlo method.

"The Metropolis algorithm aims to obtain approximate solutions to numerical problems with unmanageably many degrees of freedom and to combinatorial problems of factorial size, by mimicking a random process. Given the digital computer's reputation for deterministic calculation, it's fitting that one of its earliest applications was the generation of random numbers."

`SIAM News, Volume 33, Number 4, 2000`

Context

MCMC is crucial for making Bayesian methods practical.

The world is complicated $\rightarrow$ we want to use complicated models.

Probabilistic programming languages: BUGS, Stan, PyMC3...

Or in Tensorflow with Edward!

Exciting use case: *Bayesian deep learning*.

(Being able to examine the posterior distributions of neural net weights, instead of just their expectations.)

Review

What can you do when you cannot calculate the expectation?

Naive Monte Carlo estimation.

Take samples, then average them.

Simple, unbiased estimator.

But...

It can take a very long time.

Good first method.

Review

What can you do when you cannot even sample from your model?

Importance Sampling.

Use an approximating distribution, then correct for the fact that you are using the wrong distribution (weighting).

Why MCMC

But...

Choosing a good approximating distribution in high dimensions is challenging.

Curse of dimensionality.

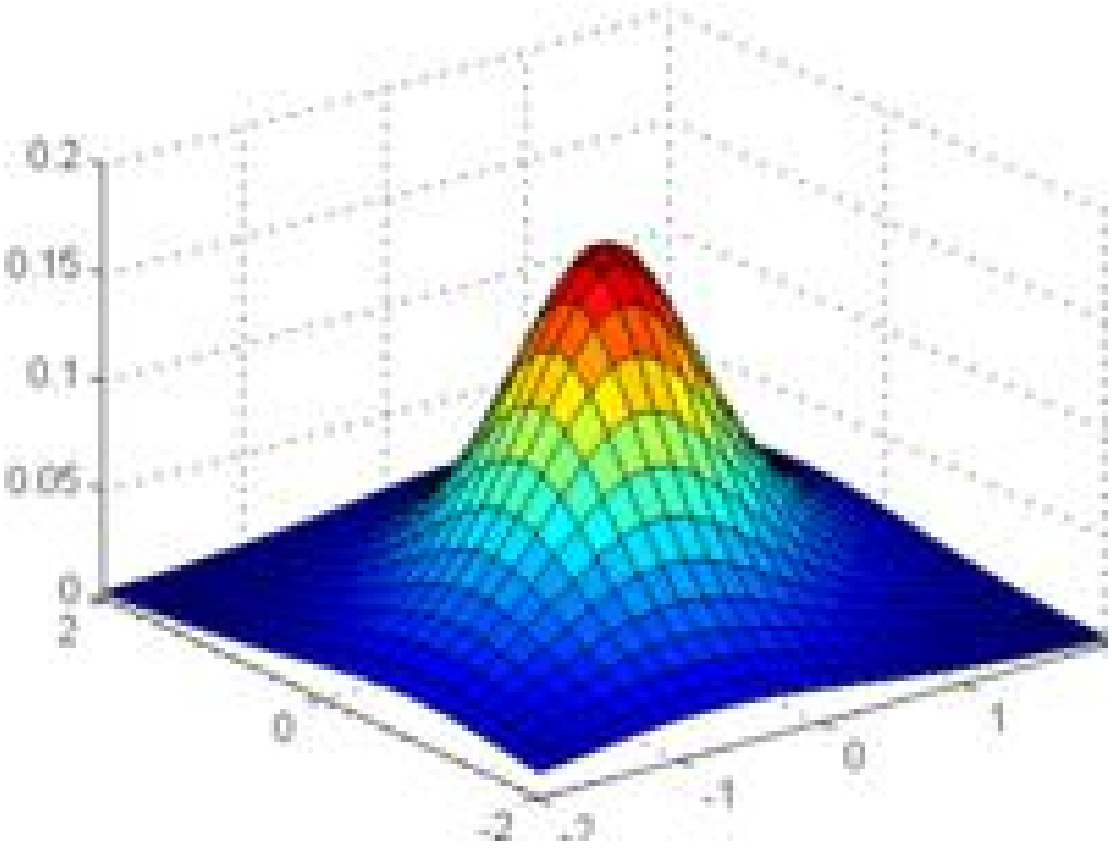
As dimensions increase, the less we can infer about the underlying distribution.

(We'd have to sample from it to find out!)

Therefore, we have little hope of picking a good *approximating* distribution for Importance Sampling.

Why MCMC

Example tolerable distribution in three dimensions

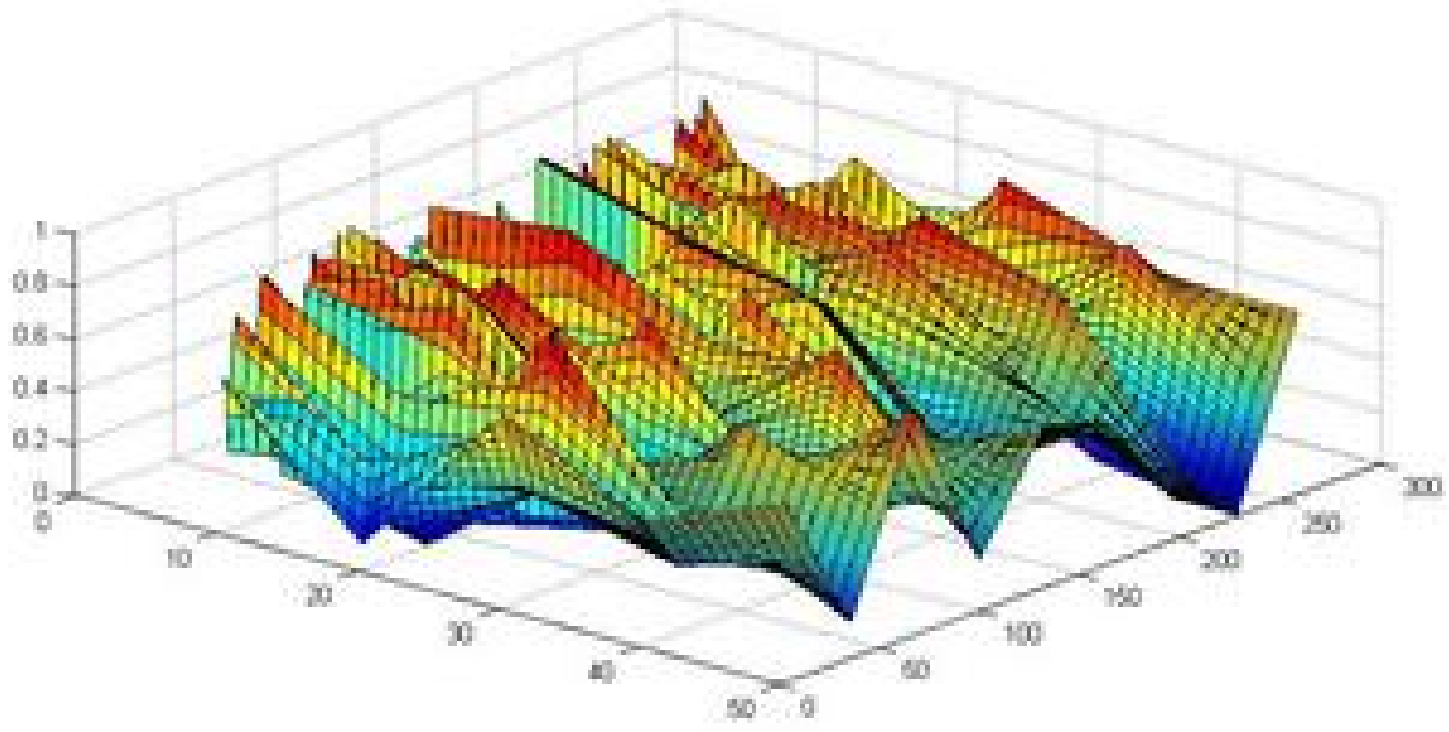


Could you use MCI for this?

Could you use Importance Sampling for this?

Why MCMC

Example difficult distribution in three dimensions



Could you use MCI for this?

Could you use Importance Sampling for this?

Why MCMC

MCMC to the rescue!

MCMC is a family of algorithms for sampling from P *according to its probability mass*.

Based on the theory of dynamical systems.

Intuition

Key ideas:

- Create a Markov chain to sample around high-probability areas of our distribution.
- Use ergodic theory to prove that this method converges to the expectation.

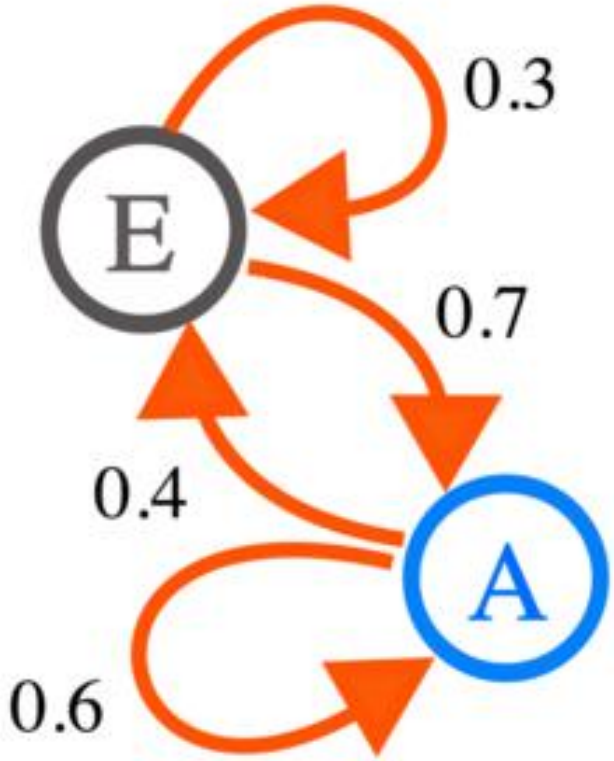
Markov chains

High-level: what is a Markov chain?

A way to evolve a system through time according to a set of states and probabilities.

Markov chains

Example:



What is the transition matrix?

$$\begin{bmatrix} P(A \to A) = ? & P(A \to E) = ? \\ P(E \to A) = ? & P(E \to E) = ? \end{bmatrix}$$

Markov chains

What is a transition matrix?

- Square
- Rows sum to 1
- All entries are nonnegative

(A transition matrix is a stochastic matrix.)

Is this a transition matrix?

$$\begin{bmatrix} 0.6 & 0.1 & 0.3 \\ 0.1 & 0.7 & 0.2 \\ 0.2 & 0.2 & 0.6 \end{bmatrix}$$

Markov chains

What is memorylessness?

The state of the world is *fully* described by the most recent history only.

$$P(x_t|x_{t-1}, x_{t-2}, \ldots, x_0) = P(x_t|x_{t-1})$$

Markov chains depend *only* on their most recent position to determine the next position.

Markov chains

We *run* a Markov chain by repeatedly *applying* the transition matrix to our probability vector.

$$x_t = x_{t-1} T$$

Conceptually, this is probabilistically moving between states in the transition diagram.

Markov chains

What is the stationary distribution?

If it exists, then the Markov chain *converges* to a steady-state set of probabilities.

$$x_1 = x_0 T$$

$$x_2 = x_1 T$$

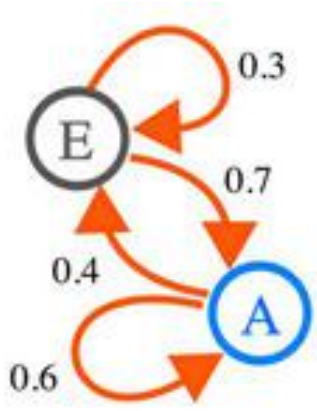
$$x_3 = x_2 T$$

$$\ldots$$

$$x_t = x_{t-1} T$$

Markov chains

Example:



$$\begin{bmatrix} 0.6 & 0.4 \\ 0.7 & 0.3 \end{bmatrix}$$

$[0.5, 0.5]$ $[0.65, 0.35]$ $[0.635, 0.365]$ $[0.6365, 0.3635]$
$[0.63635, 0.36365]$ $[0.636365, 0.363635]$

$[0.1, 0.9]$ $[0.69, 0.31]$ $[0.631, 0.369]$ $[0.6369, 0.3631]$
$[0.63631, 0.36369]$ $[0.636369, 0.363631]$ $[0.6363631, 0.3636369]$

Markov chains

Idea:

We want to use this state information to smartly walk through the probability mass.


DVD
VIDEO

Ergodic theory

With Monte Carlo estimation, we know the estimator is unbiased because samples are i.i.d.

MCI may be inefficient, but it does have that going for it.

Ergodic theory

With Markov chains, samples are not i.i.d.

Why?

Remember the definition:

$$P(x_t|x_{t-1}, x_{t-2}, ..., x_0) = P(x_t|x_{t-1})$$

Therefore, not i.i.d. State-dependent.

We call this *autocorrelation*.

Ergodic theory

How can we prove that Markov chains converge to our expectation?

Ergodic theory

Ergodic theory to the rescue!

The insight with MCMC is that if the transition matrix has certain properties, then it will converge.

Ergodic theory

Proof sketch.

We will state this, then define the terms.

If $x_0, x_1, x_2, ..., x_n$ is an *irreducible* and *time-homogenous* Markov chain, and converges to a *stationary distribution*, then the sample mean converges to the true mean:

$$\frac{1}{N}\sum_{i=1}^{N} f(x_i) \xrightarrow[n\to\infty]{} E[f(x)]$$

Further, if the Markov chain is *aperiodic*, then we can sample from it:

$$P(x_n = \alpha_n | x_0 = \alpha_0) \xrightarrow[n\to\infty]{} \pi(x)$$

We can start from any point and sample and *eventually* arrive at the true value.

Ergodic theory

Define: *Time-homogenous*

A Markov chain is time-homogenous if the transition matrix values do not depend on the time step.

Intuitively, this means that the probability of transitioning from one state to another is independent of how many samples we've made.

Ergodic theory

Define: *Irreducible*

A Markov chain is irreducible if there is a transition path from every state to every other state.

Intuitively, this means that you can get from any state to any other state (eventually).

Another way to think about it is that the Markov chain needs to be able to visit any part of the probability mass.

Ergodic theory

Define: *Stationary distribution*

A Markov chain has a stationary distribution if there is some state that is a fixed-point for the chain.

Intuitively, this means that there is a global optimum that can be found.

(But, don't think about this as optimization. It's sampling.)

Ergodic theory

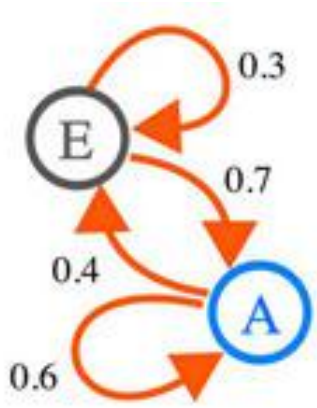
Define: *aperiodic*

This is the most abstract definition.

An irreducible Markov chain is aperiodic if states can be reached by paths of any length.

Ergodic theory

Example: *aperiodic*



For example, to go from A to E, we can construct arbitrary path lengths:

$$A \to E$$

$$A \to E \to E$$

$$A \to E \to A \to E$$

$$A \to E \to A \to E \to E$$

$$A \to E \to A \to E \to A \to E$$

Ergodic theory

Returning to our definition:

If $x_0, x_1, x_2, ..., x_n$ is an *irreducible*, *time-homogenous* Markov chain, and converges to a *stationary distribution*, then sample means converge to the true mean:

$$\frac{1}{N}\sum_{i=1}^{N} f(x_i) \xrightarrow[n\to\infty]{} E[f(x)]$$

Further, if the Markov chain is *aperiodic*, then we can sample from it:

$$P(x_n = \alpha_n | x_0 = \alpha_0) \xrightarrow[n\to\infty]{} \pi(x)$$

We can start from any point and sample and *eventually* arrive at the true value.

Application

Now we believe the fact that Markov chains can be used to construct good samples from distributions.

But we do not yet have a *procedure* to use.

The first such algorithm is the *Metropolis algorithm*.

Metropolis algorithm

Let $f(x)$ be a function proportional to our target distribution P.

- Choose an initial proposal point x_0, and choose a jumping probability density g. Typically this is $\mathcal{N}(0, 1)$.
- For each iteration t:
  - Append the current point to the list of samples.
  - Generate a candidate x_{t+1} by sampling from $g(x_{t+1}|x_t)$.
  - Calculate the *acceptance ratio* $\alpha = f(x_{t+1})/f(x_t)$
  - If $\alpha > 1$, then certainly accept the update to x_{t+1}.
  - Else, accept the update with probability α.

Metropolis algorithm

The Markov chains created by the Metropolis algorithm are ergodic.

Therefore the Markov chains are valid for MCMC and we will converge to our estimator.

You'll work with the Metropolis algorithm in lab.