Day3 : Introduction to Markov Chain Monte Carlo

[FastCampus] AI센터 베이지안 통계과정

강사: 전인수 (isjeon@vision.snu.ac.kr)

JUN 12, 2019

목차

- Monte Carlo Estimation
  - 1d sampling (discrete)
  - 1d sampling (continuous) – rejection sampling
- Markov Chain
- Gibbs Sampling
- Metropolis Hastings
- MCMC examples (with PyMC3)

Ch 1. Monte Carlo Estimation

Monte Carlo Markov Chain (MCMC)

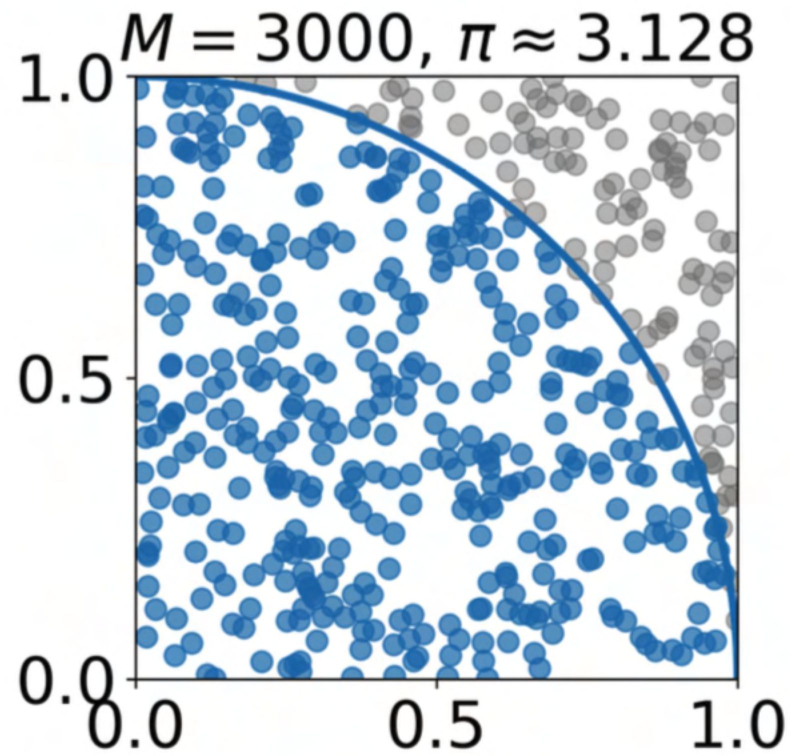
- 몬테카를로 기법(Monte Carlo Method)은 난수를 이용하여 함수의 값을 확률적으로 근사하는 알고리즘을 뜻하는 용어다.

- 일반적으로 평균값(expected value)을 근사하기 위해 사용되며, Bayesian Theory에서는 Posterior의 sampling을 얻기 위해 활용된다.

- Two common MCMC approaches:
  - Gibbs sampling – reducing multidimensional sampling to a sequence of 1d
  - Metropolis Hastings – rejection sampling for Markov Chains (gives more freedom)

Monte Carlo

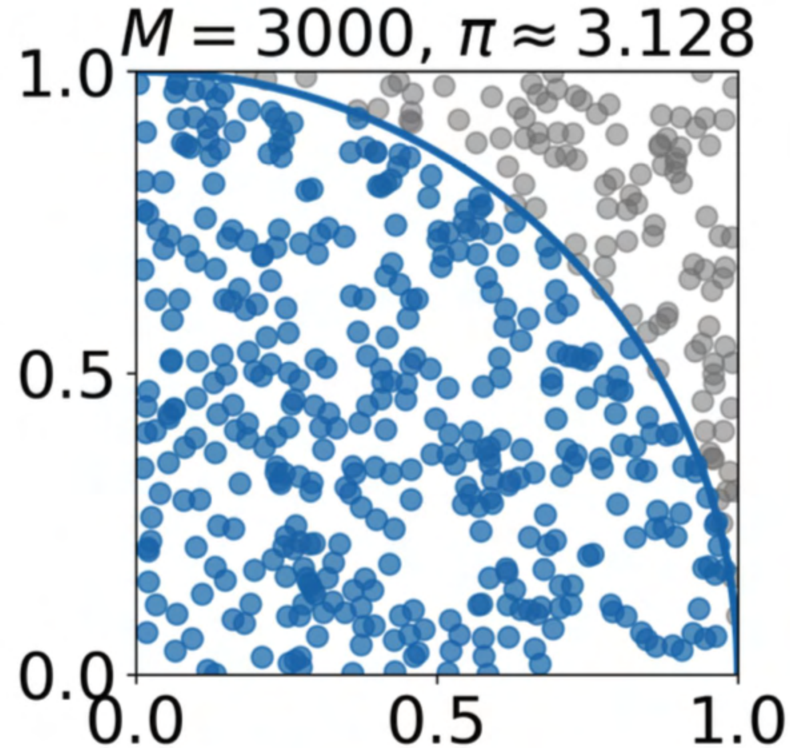
Estimate expected values by sampling



Monte Carlo

Estimate expected values by sampling

$$\frac{\pi}{4} = \mathbb{E}\left[x^2 + y^2 \leq 1\right]$$



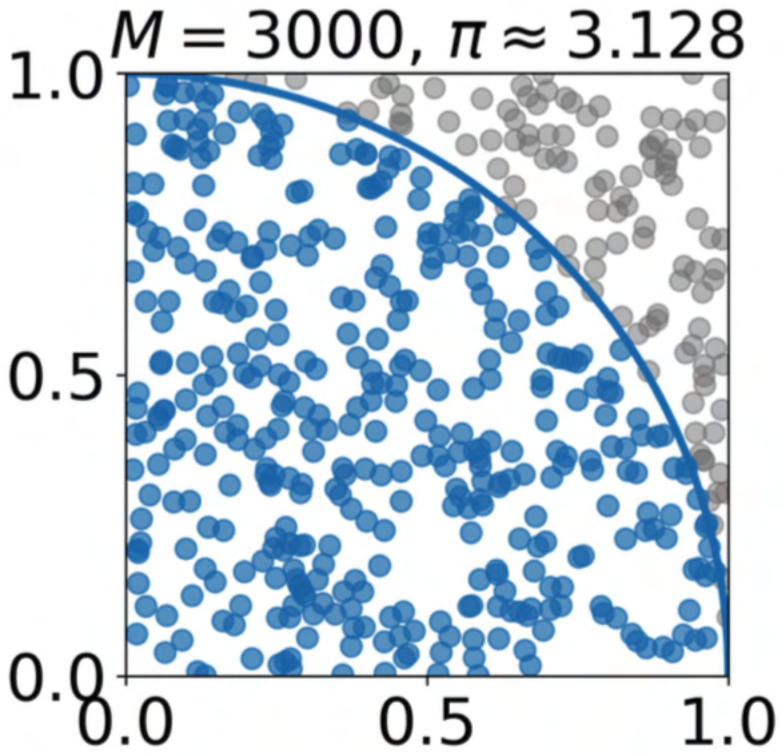
Monte Carlo

Estimate expected values by sampling

$$\frac{\pi}{4} = \mathbb{E}\left[x^2 + y^2 \leq 1\right]$$

$$\approx \frac{1}{M} \sum_{s=1}^{M} [x_s^2 + y_s^2 \leq 1]$$

$$x_s, y_s \sim \mathcal{U}(0, 1)$$



Monte Carlo

Estimate expected values by sampling

$$\mathbb{E}_{p(x)} f(x) \approx \frac{1}{M} \sum_{s=1}^{M} f(x_s)$$

$$x_s \sim p(x)$$

Monte Carlo Approximation is Unbiased

Monte Carlo

$$\mathbb{E}_{p(x)} f(x) \approx \frac{1}{M} \sum_{s=1}^{M} f(x_s)$$

$$x_s \sim p(x)$$

Unbiased estimate (larger M => better accuracy)

$$\mathbb{E}_{p(x)} \frac{1}{M} \sum_{s=1}^{M} f(x_s) = \mathbb{E}_{p(x)} f(x)$$

Monte Carlo

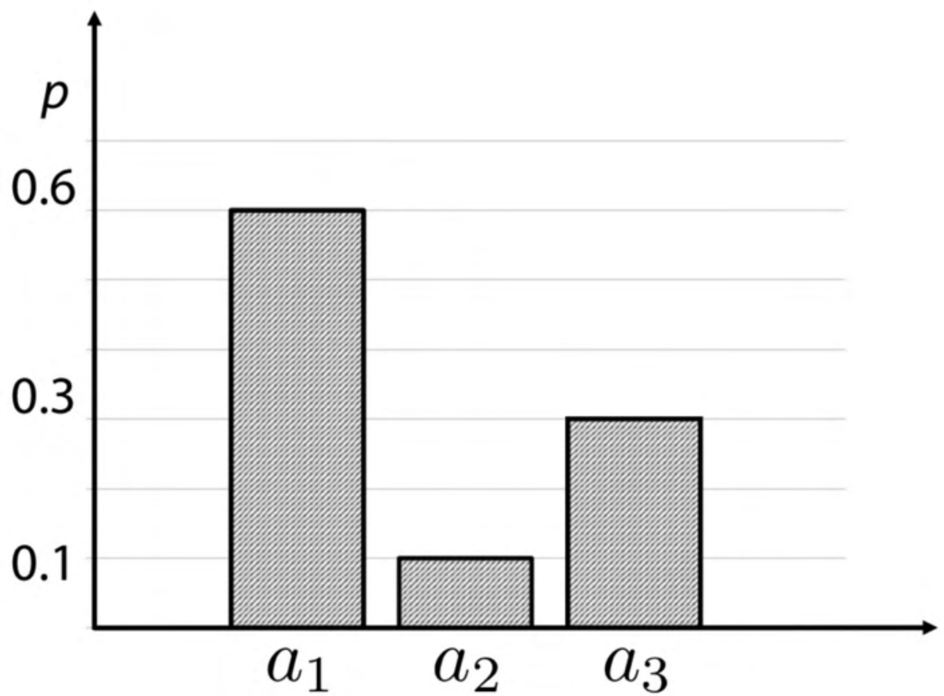
Why do we need to estimate expected values?

- Bayesian Analysis에서는 일반적으로 Sample들을 이용해서, Posterior Distribution을 분석하거나 Predictive Distribution을 계산하는데 활용된다.

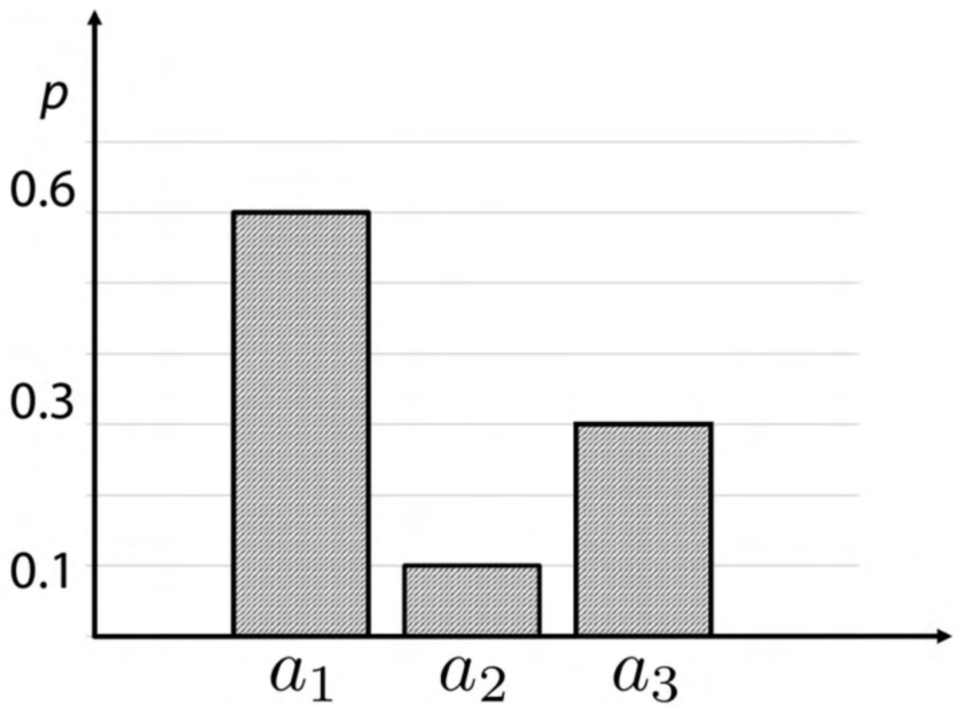
$$
\begin{aligned}
& p(y \mid x, Y_{\text{train}}, X_{\text{train}}) \\
& = \int p(y \mid x, w) p(w \mid Y_{\text{train}}, X_{\text{train}}) dw \\
& = \mathbb{E}_{p(w \mid Y_{\text{train}}, X_{\text{train}})} p(y \mid x, w)
\end{aligned}
$$

$$
p(w \mid Y_{\text{train}}, X_{\text{train}}) = \frac{p(Y_{\text{train}} \mid X_{\text{train}}, w) p(w)}{Z}
$$

1d sampling (discrete)

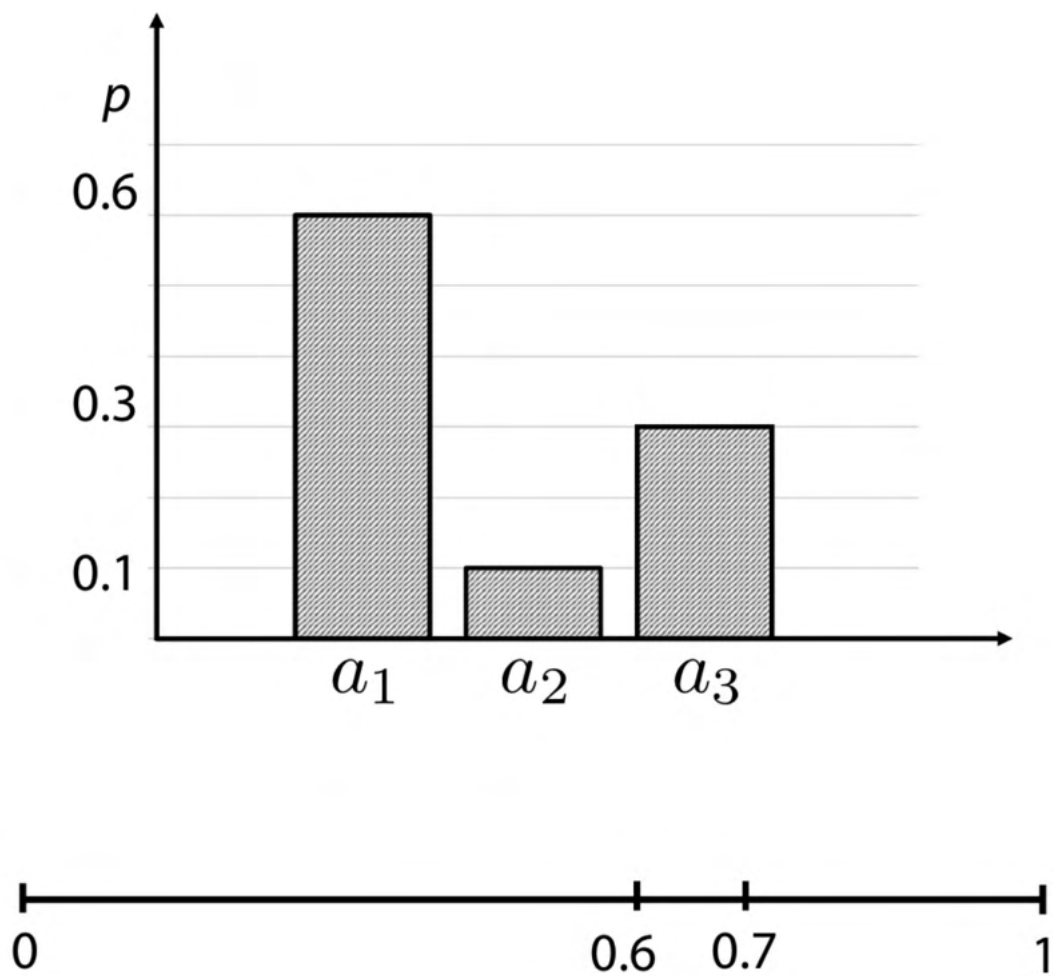


1d sampling (discrete)

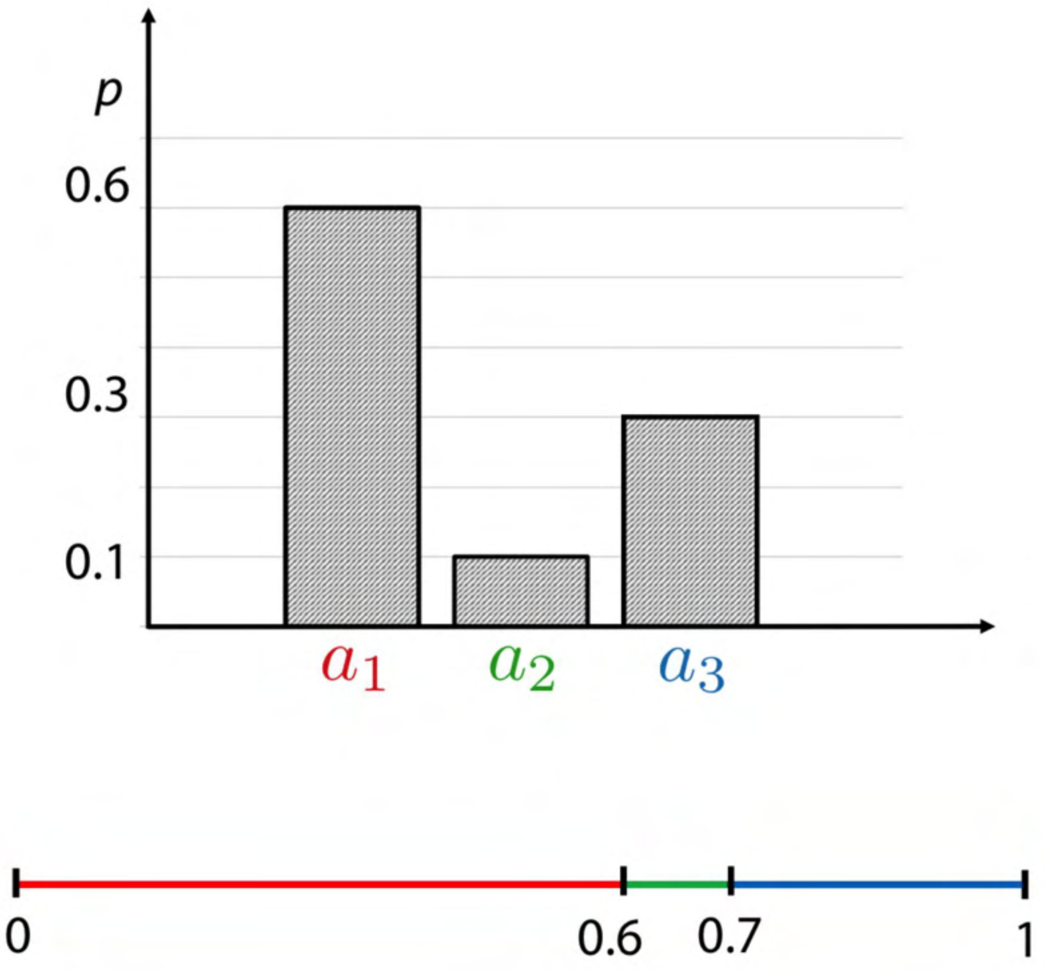


We can always sample from uniform $\mathcal{U}[0,1]$

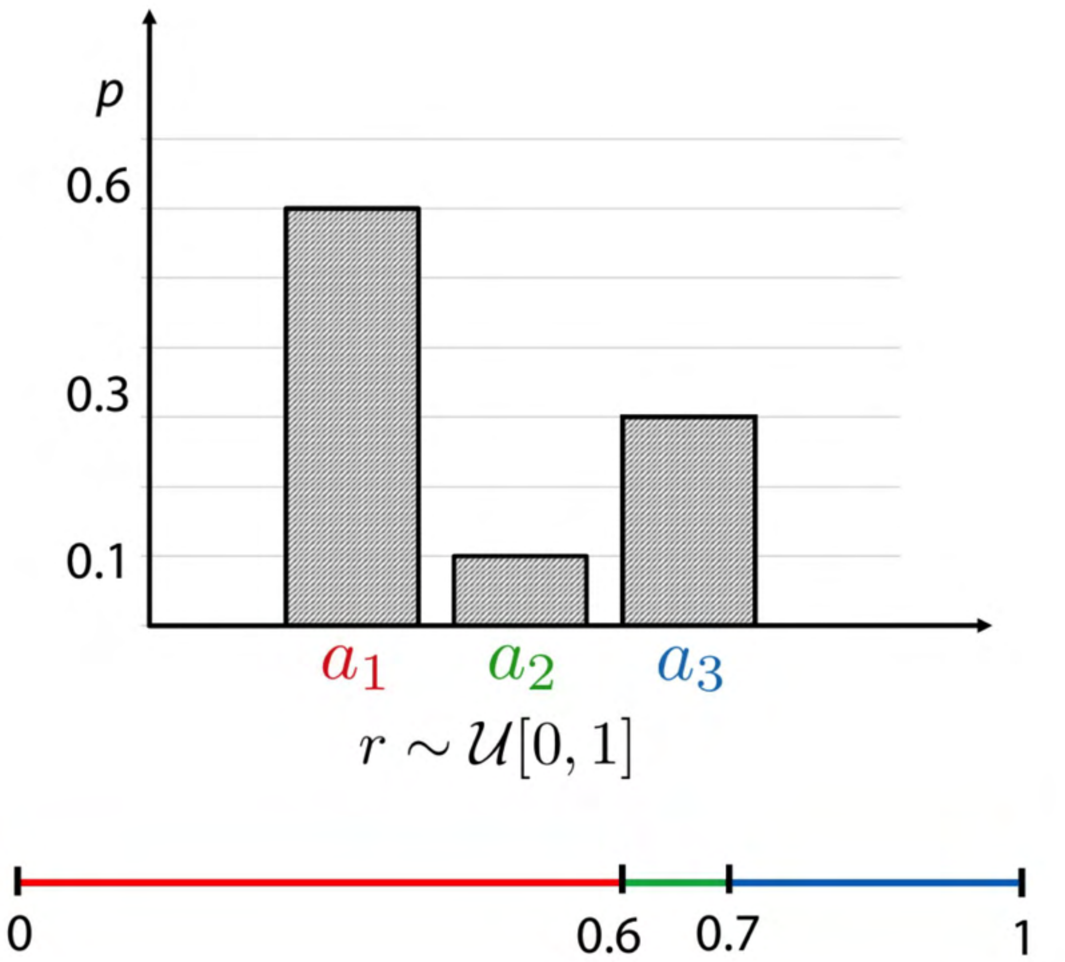
1d sampling (discrete)



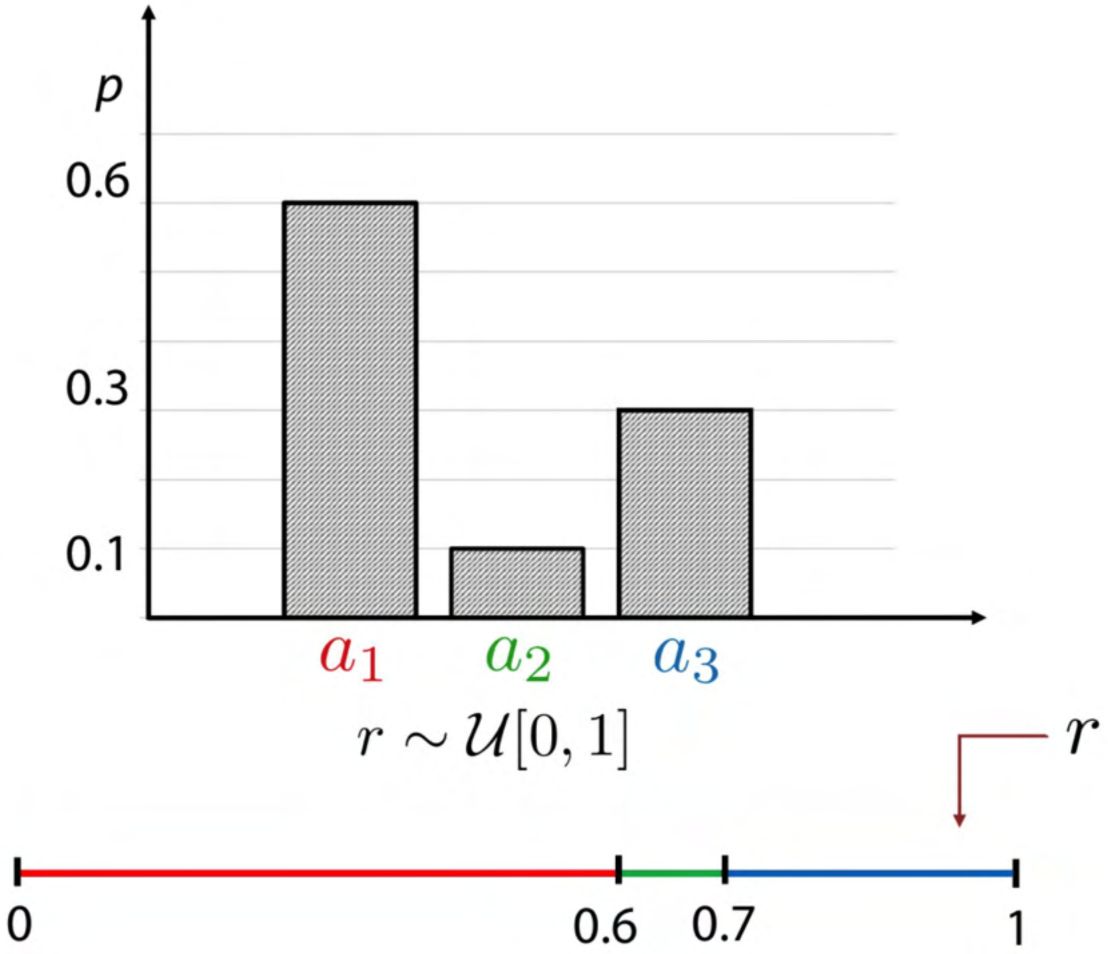
1d sampling (discrete)



1d sampling (discrete)



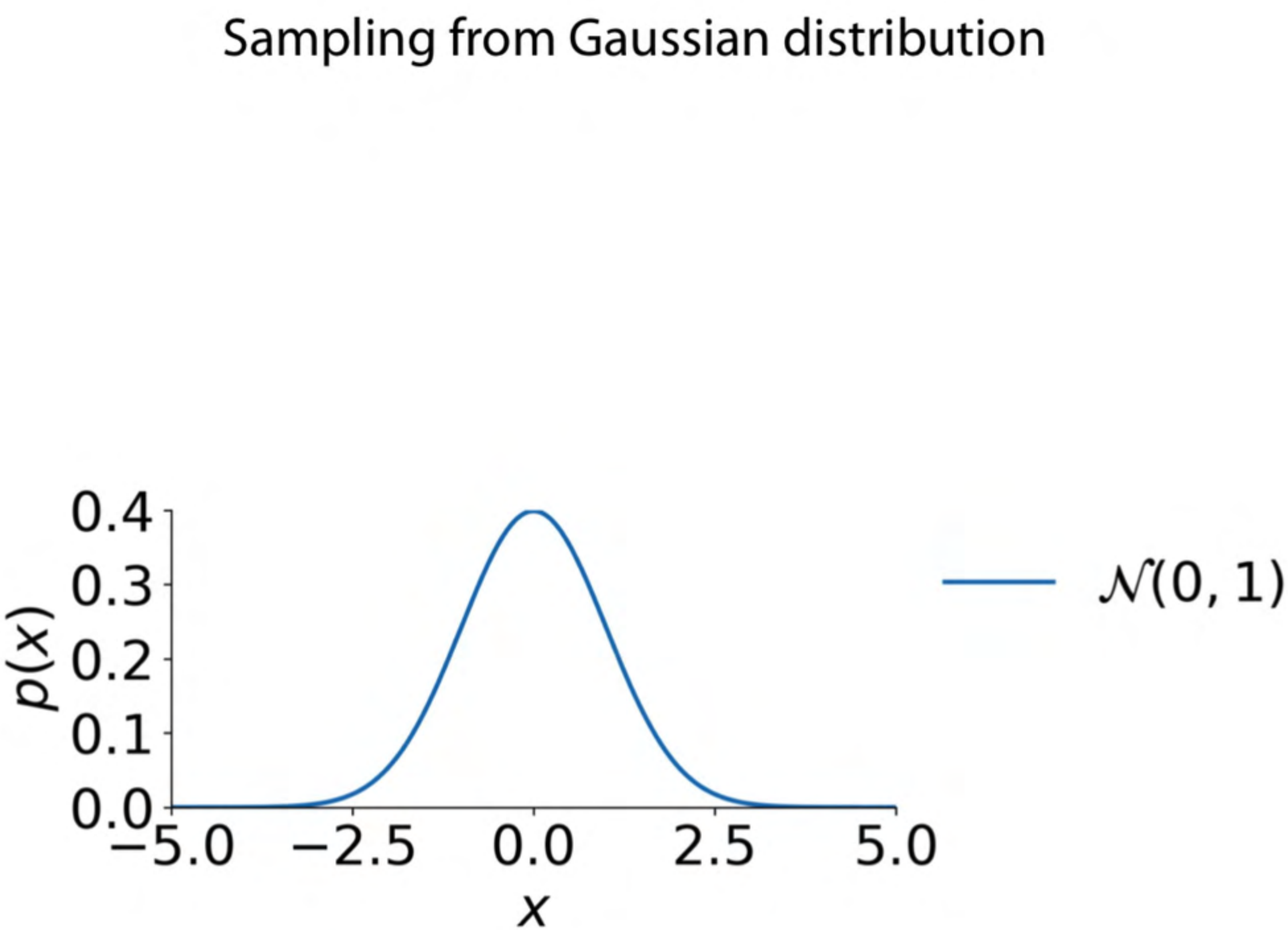
1d sampling (discrete)



1d sampling (discrete) - Summary

- 차원이 낮은 discrete distribution으로 부 터 sampling을 하는 방법은 아주 쉽다
  - At least then number of values is < 100 000
- 고차원인 경우 훨씬 많은 sample이 필요하다.

1d sampling (continuous) - Gaussian

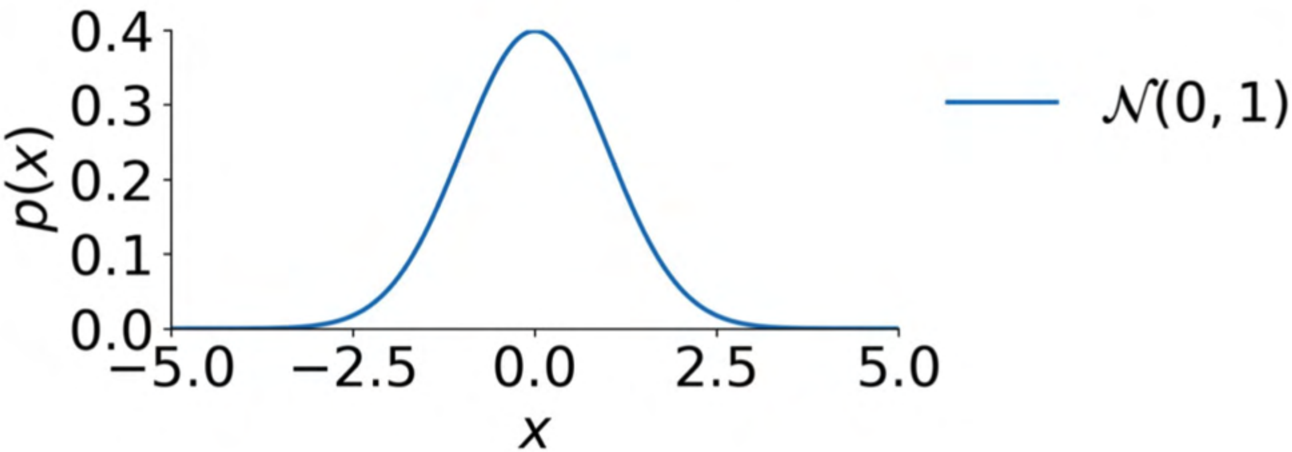


1d sampling (continuous) - Gaussian

Sampling from Gaussian distribution

$$z = \sum_{i=1}^{12} x_i - 6, \quad x_i \sim \mathcal{U}[0,1]$$

$$p(z) \approx \mathcal{N}(0,1)$$

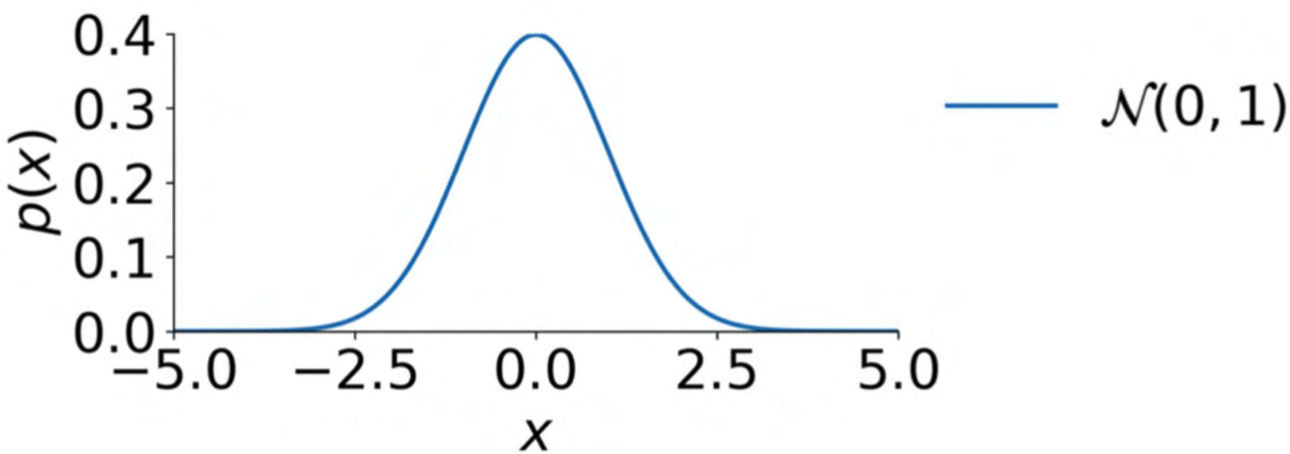


1d sampling (continuous) - Gaussian

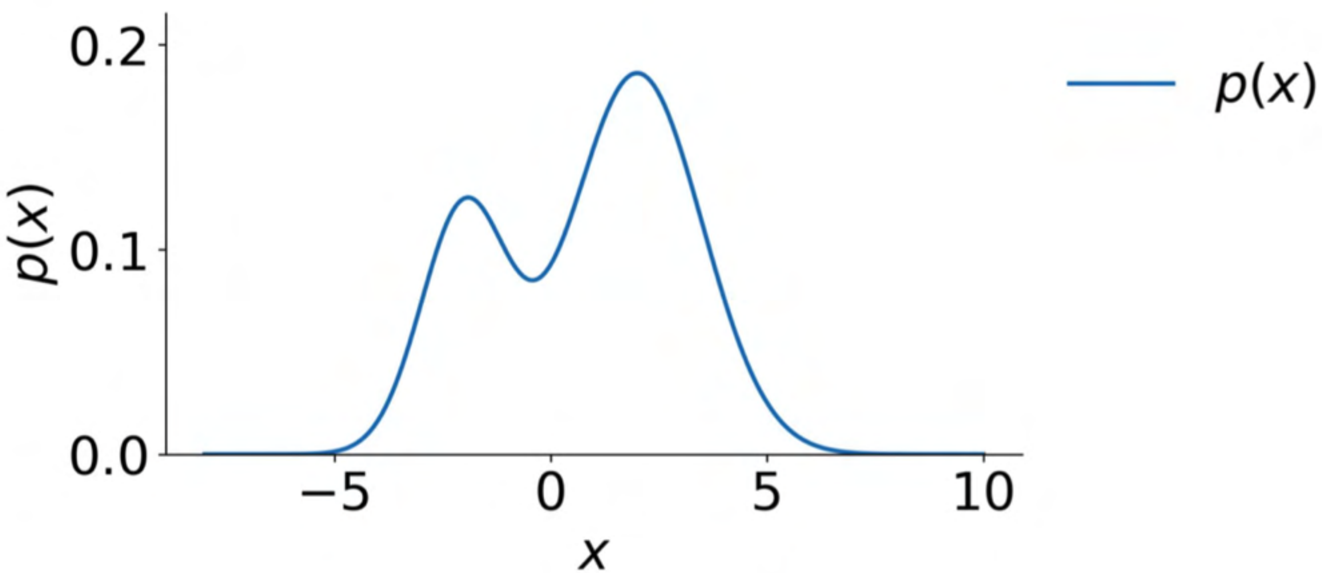
Sampling from Gaussian distribution

Or call library function ☺

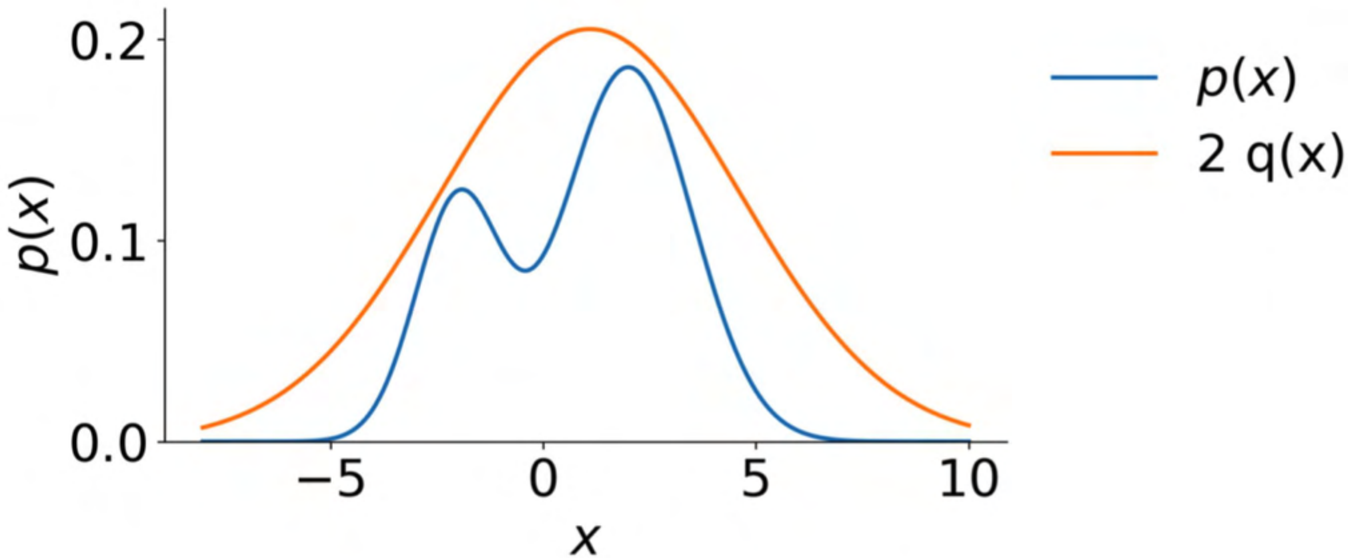
z = numpy.random.randn()



1d sampling (continuous) - General



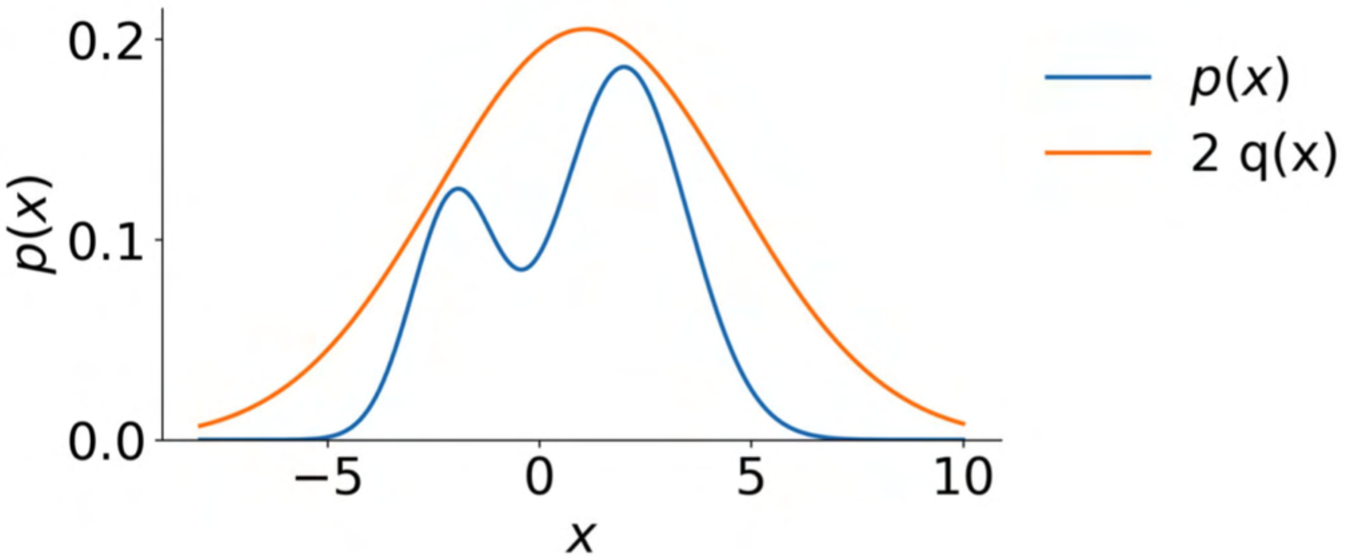
Rejection sampling



$$q(x) = \mathcal{N}(1, 3^2)$$

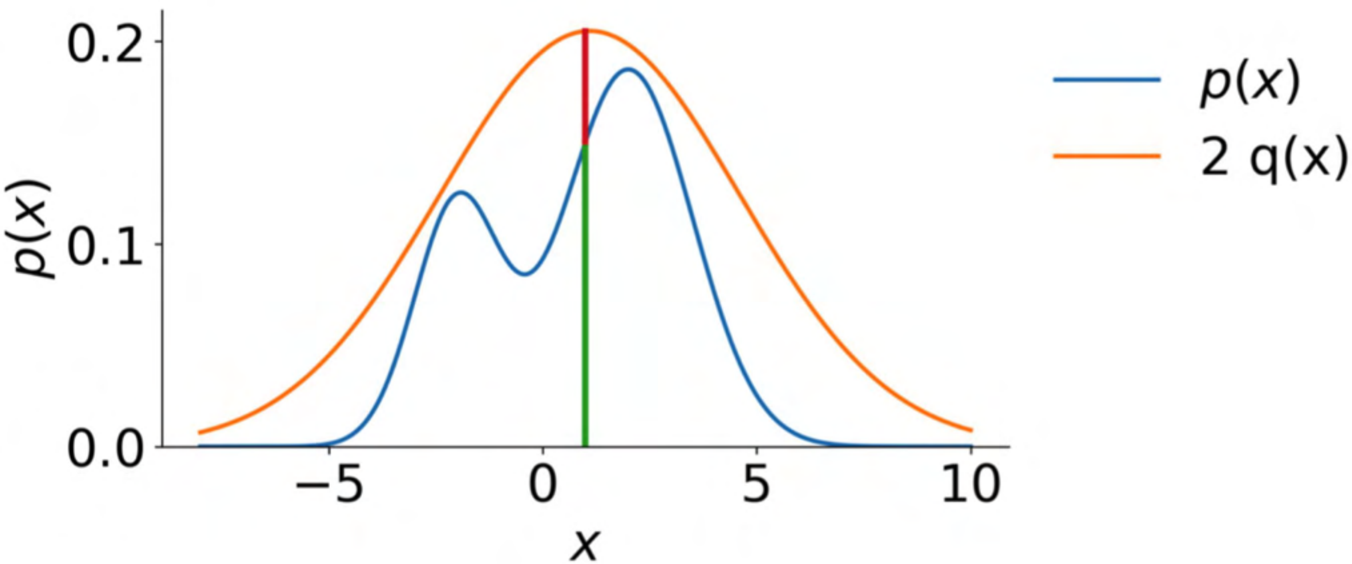
$$p(x) \leq 2q(x)$$

Rejection sampling



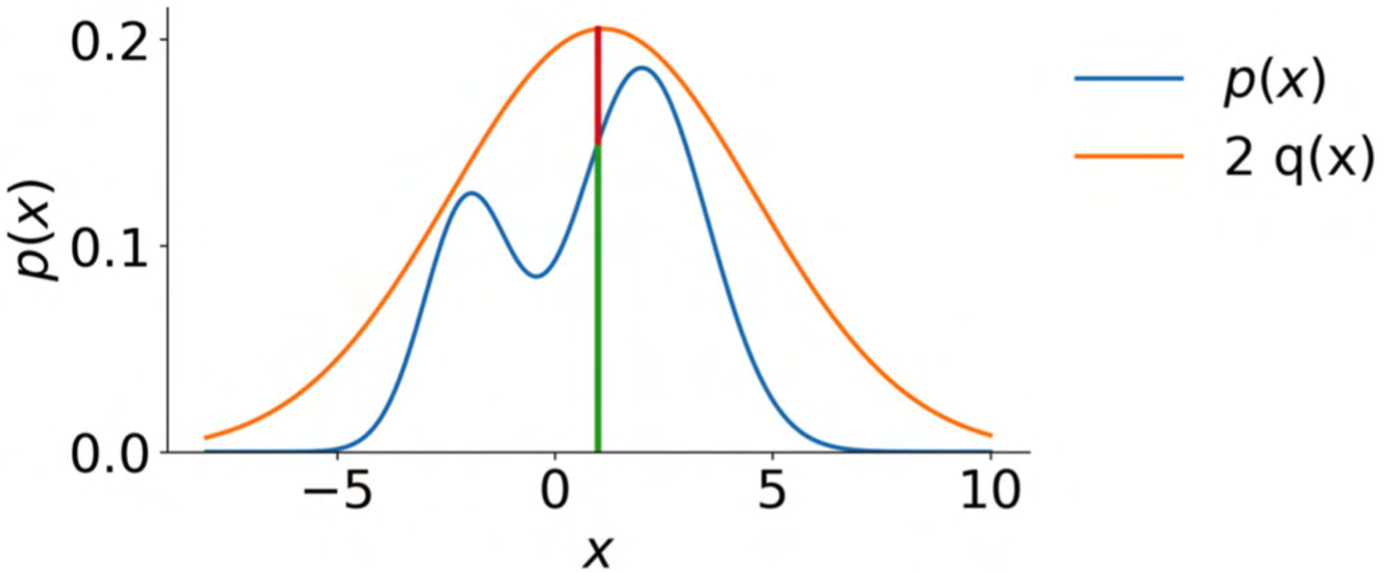
$$\widetilde{x} \sim q(x)$$

Rejection sampling



$$\widetilde{x} \sim q(x)$$

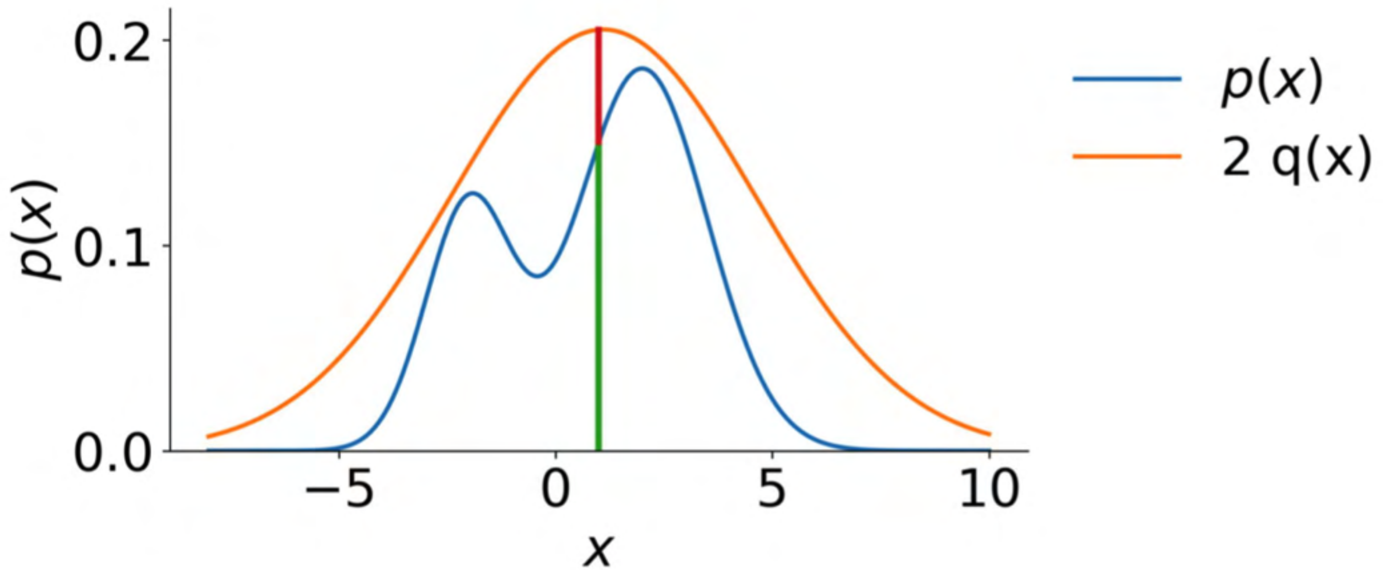
Rejection sampling



$\widetilde{x} \sim q(x)$ $\qquad$ $y \sim \mathcal{U}[0, 2q(\widetilde{x})]$

Accept $\widetilde{x}$ with probability $\frac{p(x)}{2q(x)}$

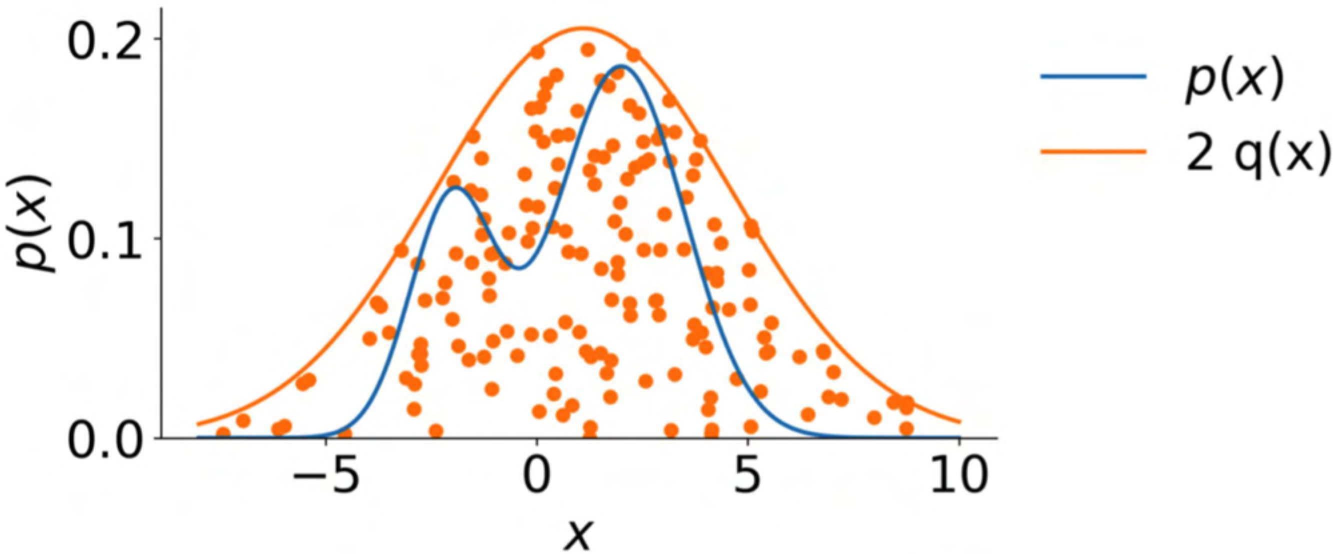
Rejection sampling



$$\widetilde{x} \sim q(x) \qquad\qquad y \sim \mathcal{U}[0, 2q(\widetilde{x})]$$

Accept $\widetilde{x}$ with probability $\dfrac{p(x)}{2q(x)}$: if $y \leq p(x)$

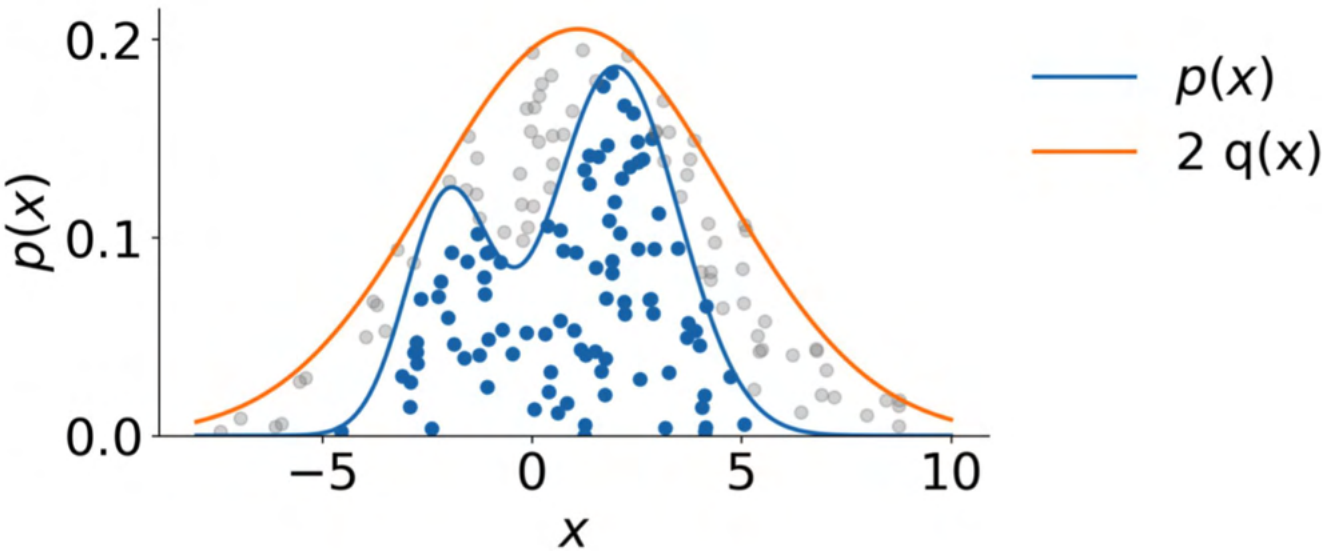
Rejection sampling



$$\widetilde{x} \sim q(x) \qquad\qquad y \sim \mathcal{U}[0, 2q(\widetilde{x})]$$

Accept $\widetilde{x}$ with probability $\dfrac{p(x)}{2q(x)}$: if $y \leq p(x)$

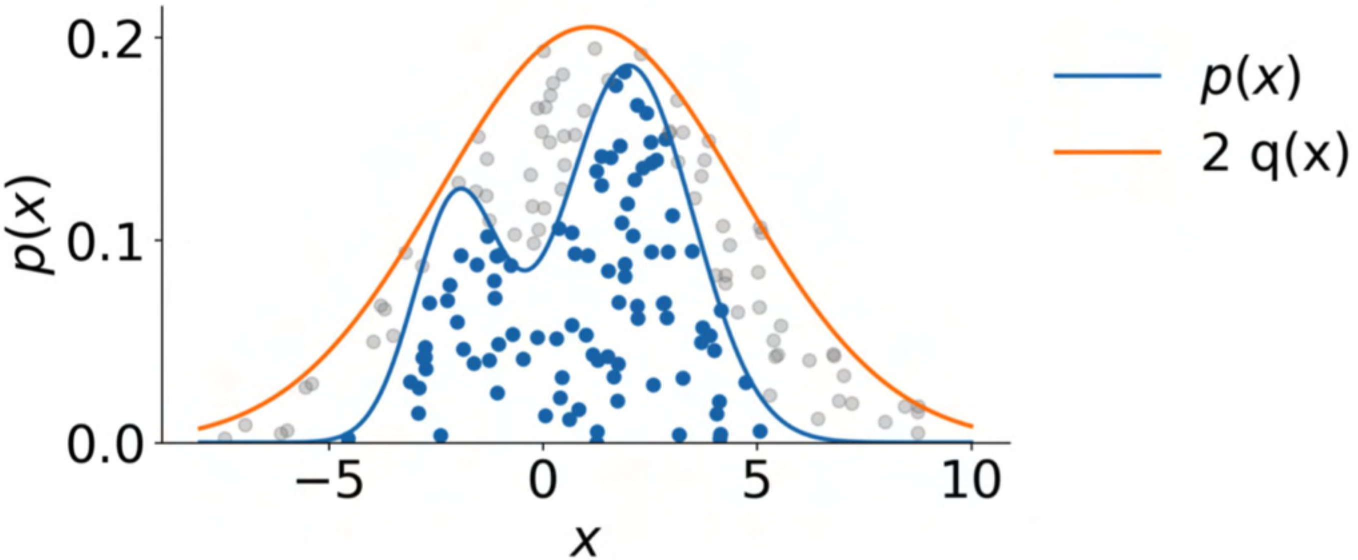
Rejection sampling



$$\widetilde{x} \sim q(x) \qquad\qquad y \sim \mathcal{U}[0, 2q(\widetilde{x})]$$

Accept $\widetilde{x}$ with probability $\dfrac{p(x)}{2q(x)}$: if $y \leq p(x)$

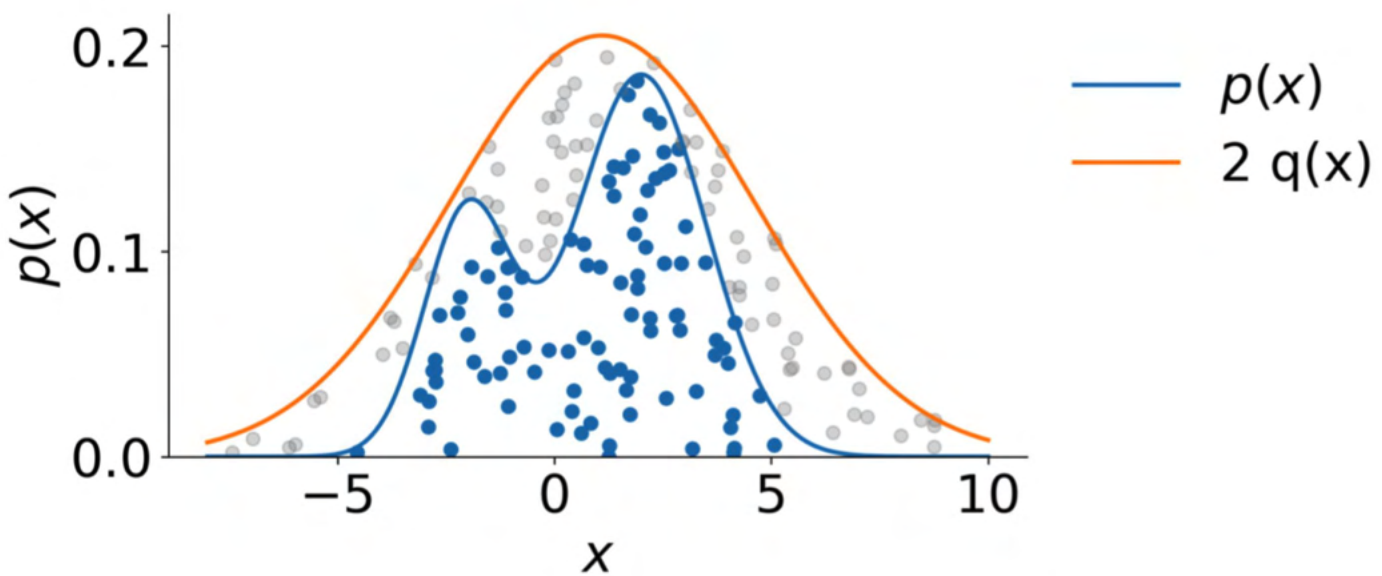
Rejection sampling



$$p(x) \leq Mq(x)$$

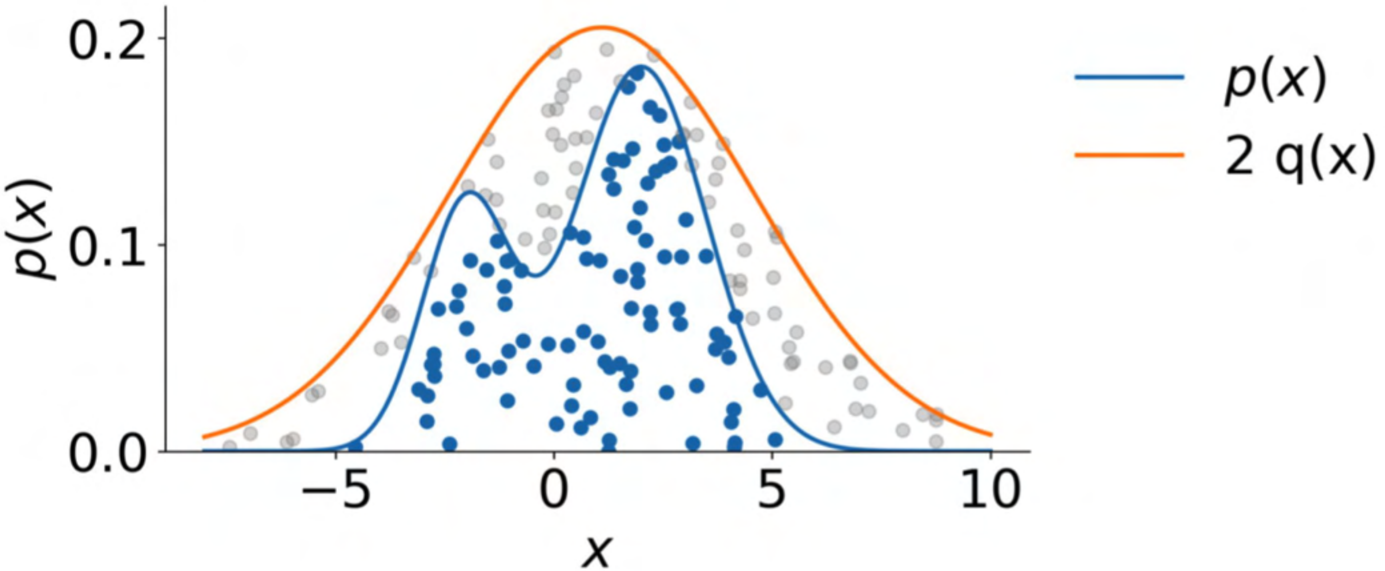
Accepts $\frac{1}{M}$ points on average

Rejection sampling



$$\frac{\widehat{p}(x)}{Z} \leq Mq(x)$$

Rejection sampling



$$\widehat{p}(x) \leq \underbrace{ZM}_{\widetilde{M}} q(x)$$

Monte Carlo Sampling – Summary

Pros:

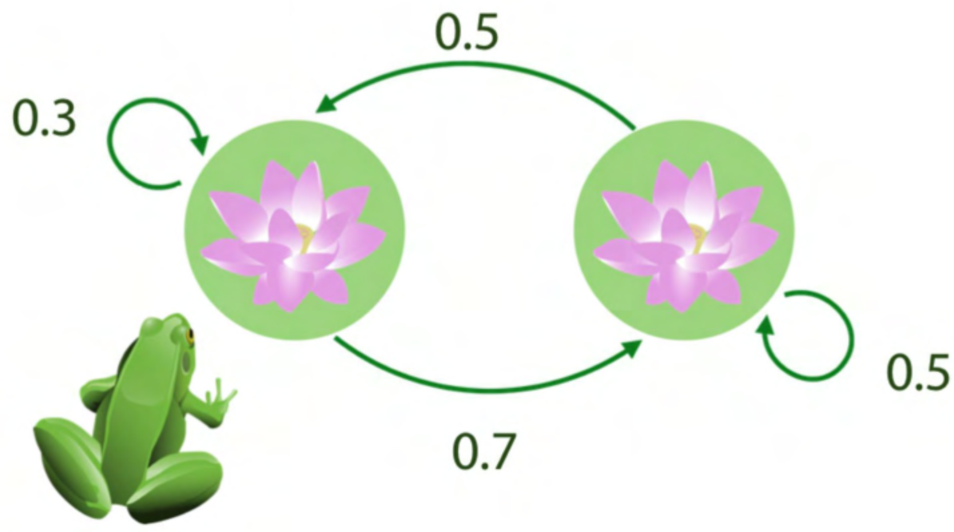
- Works for most distributions (even unnormalized)

Cons:

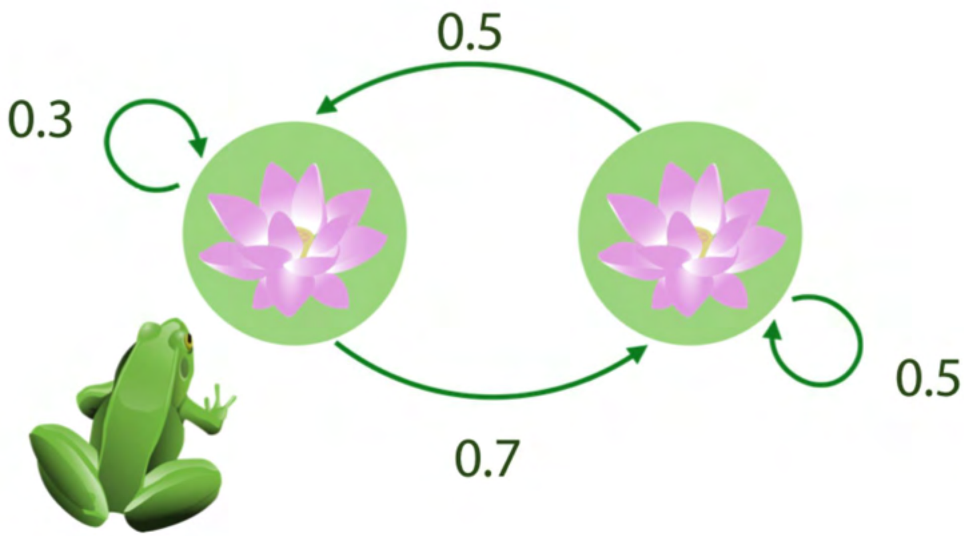
- If q and p are too different (M is large), rejects most of the points
- M is large for d-dimensional distributions

Ch 2. Markov Chain

Markov Chains



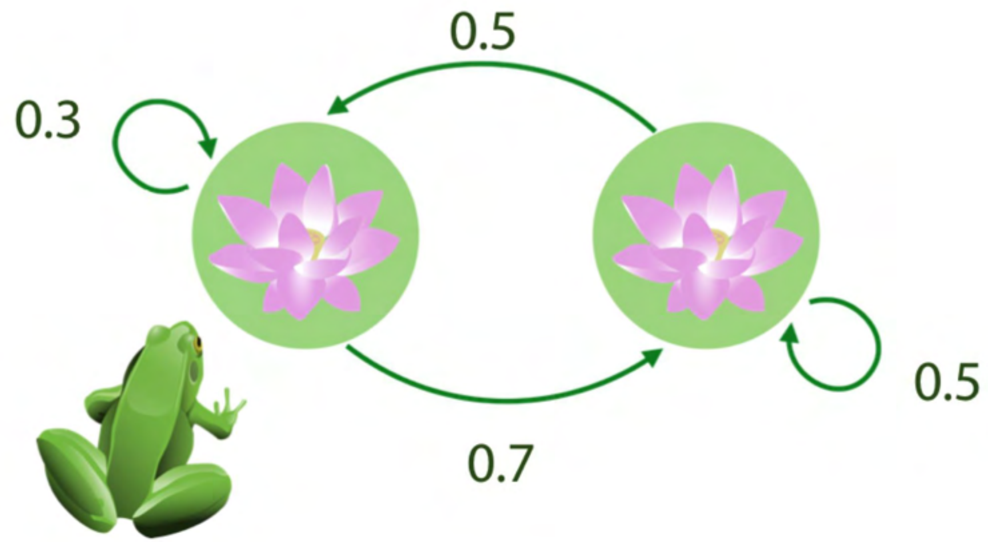
Markov Chains



$$T(\mathrm{L} \to \mathrm{L}) = 0.3 \qquad T(\mathrm{R} \to \mathrm{L}) = 0.5$$

$$T(\mathrm{L} \to \mathrm{R}) = 0.7 \qquad T(\mathrm{R} \to \mathrm{R}) = 0.5$$

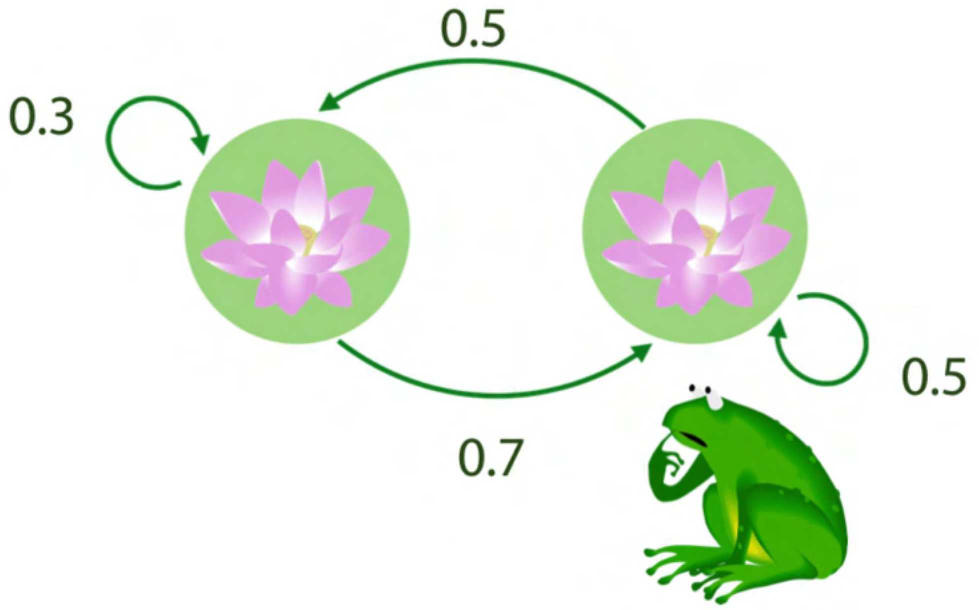
Markov Chains



Timestamp: **1**

Log: **L**

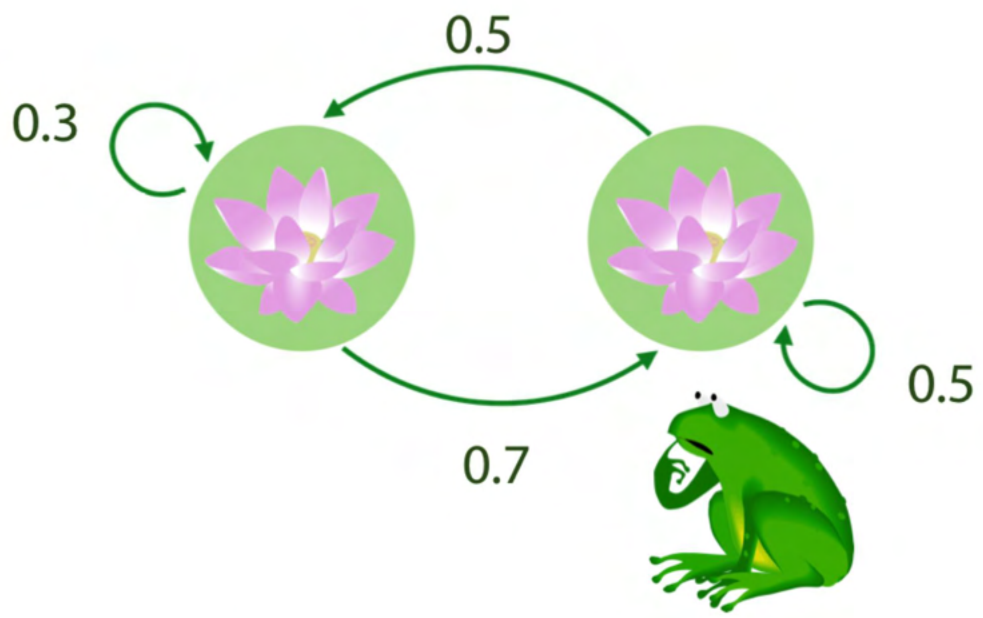
Markov Chains



Timestamp: **2**

Log: **L R**

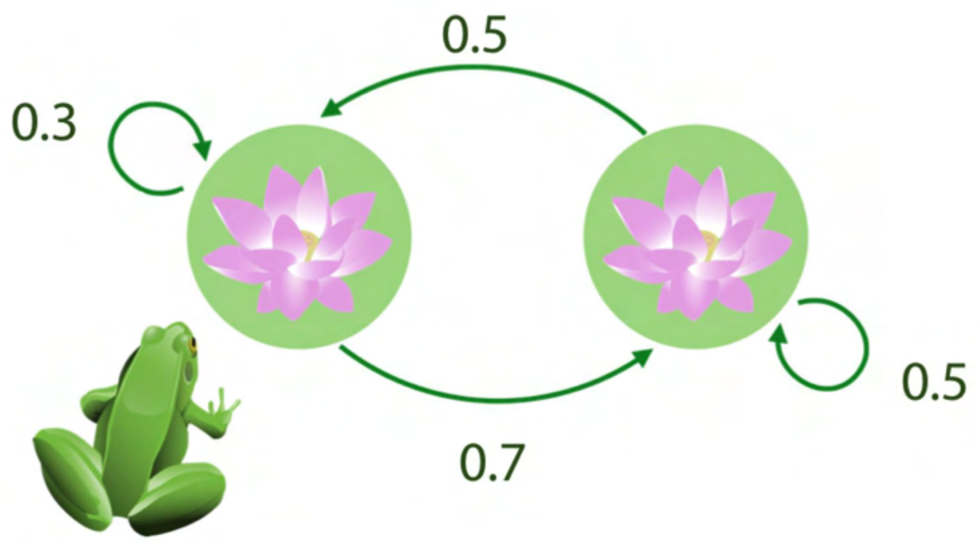
Markov Chains



Timestamp: **3**

Log: **L R R**

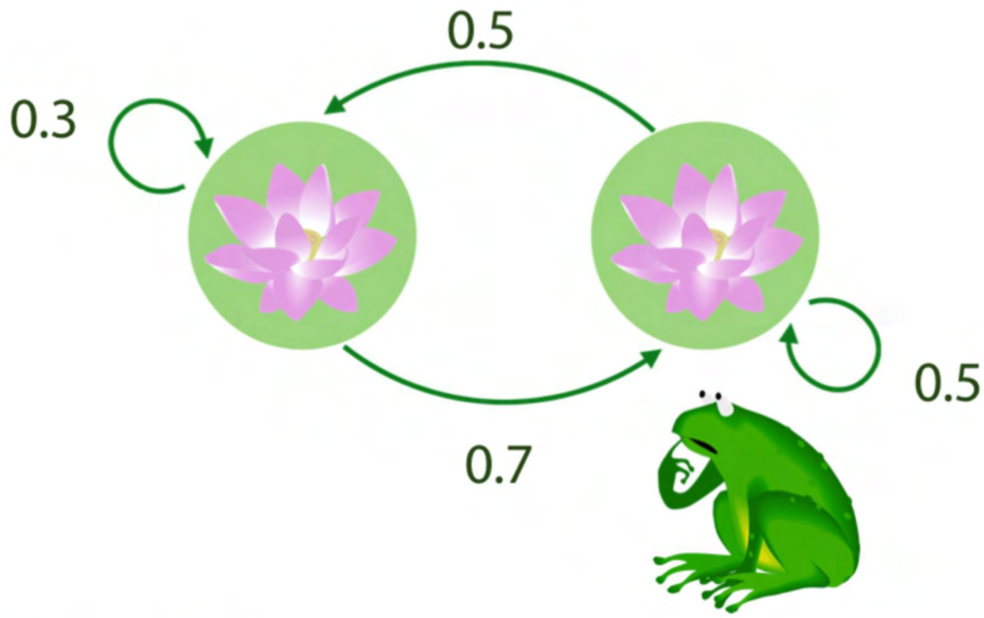
Markov Chains



Timestamp: **4**

Log: **L R R L**

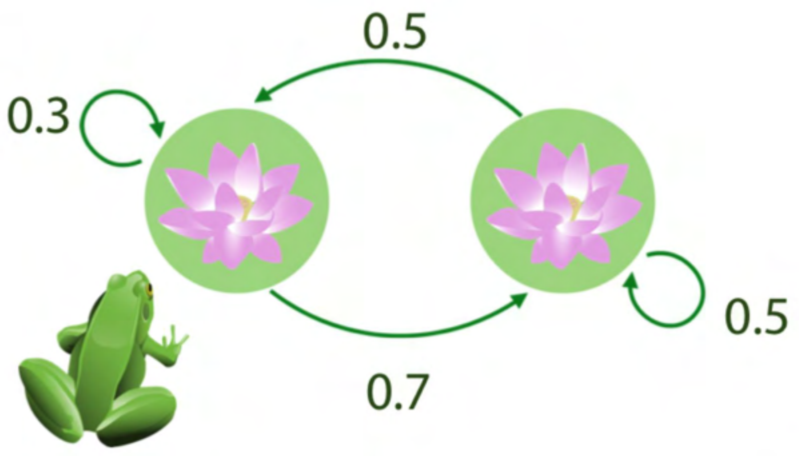
Markov Chains



Timestamp: **5**

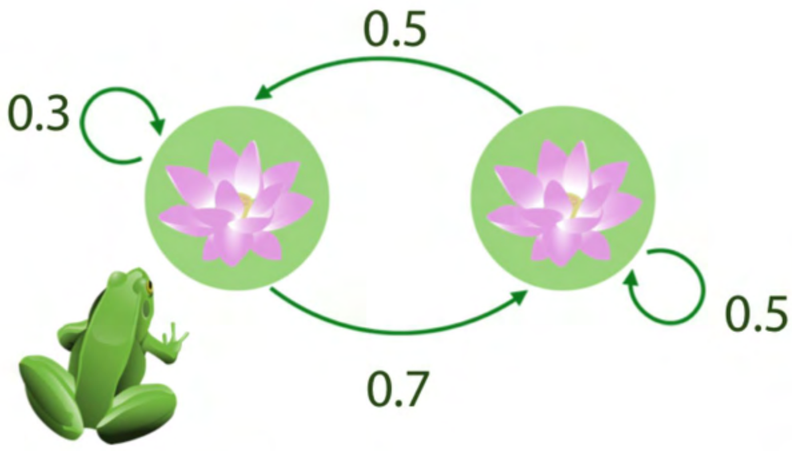
Log: **L R R L R**

Markov Chains



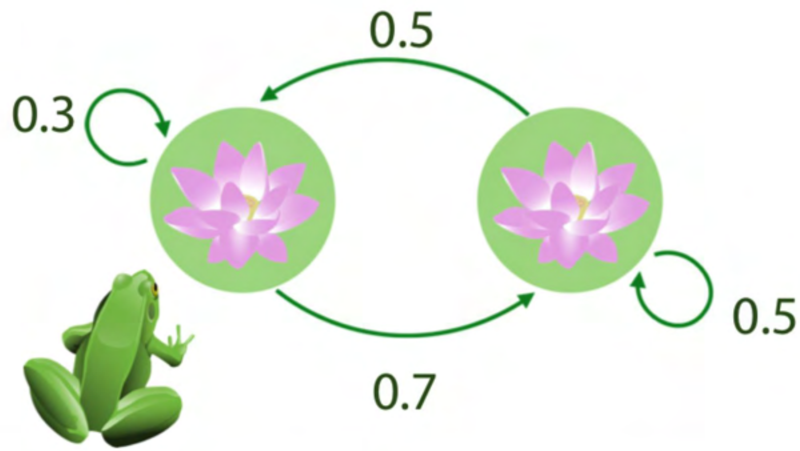
	p(Left)	p(Right)
x^1	1	0

Markov Chains



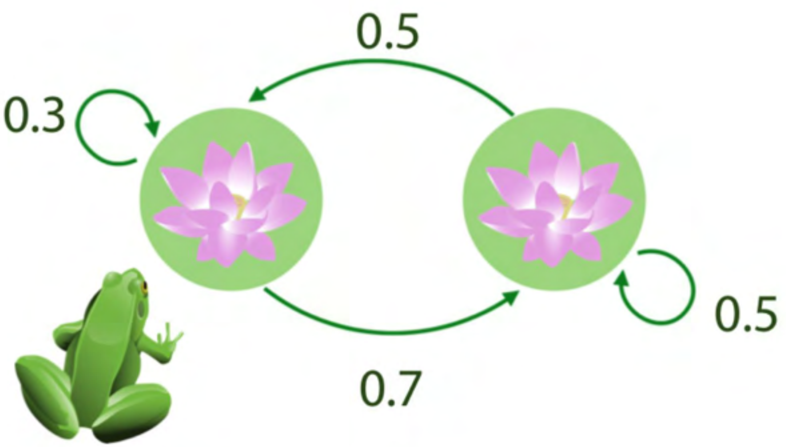
	p(Left)	p(Right)
x^1	1	0
x^2	0.3	0.7

Markov Chains



	p(Left)	p(Right)
x^1	1	0
x^2	0.3	0.7
x^3		

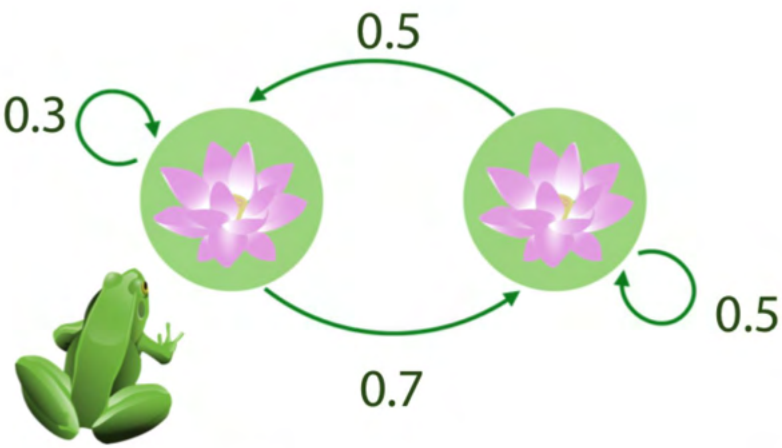
Markov Chains



	p(Left)	p(Right)
x^1	1	0
x^2	0.3	0.7
x^3		

$$p(x^3) = p(x^3 \mid x^2 = \mathrm{L})p(x^2 = \mathrm{L}) + p(x^3 \mid x^2 = \mathrm{R})p(x^2 = \mathrm{R})$$

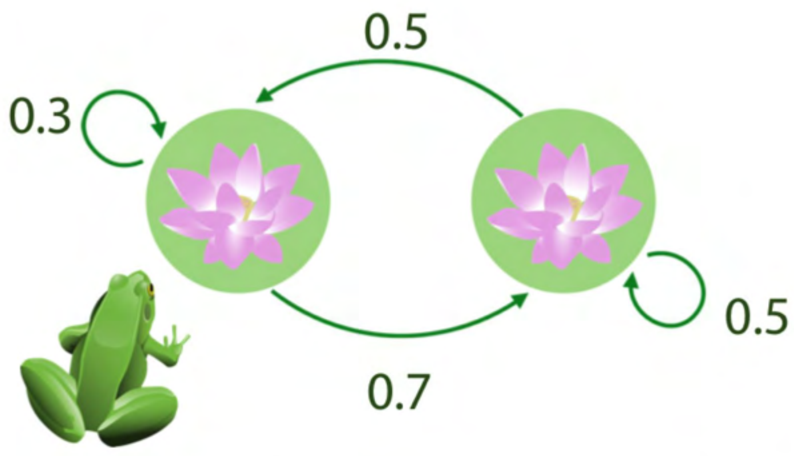
Markov Chains



	p(Left)	p(Right)
x^1	1	0
x^2	0.3	0.7
x^3	$0.3^2 + 0.7 \cdot 0.5$	

$$p(x^3) = p(x^3 \mid x^2 = \mathrm{L})p(x^2 = \mathrm{L}) + p(x^3 \mid x^2 = \mathrm{R})p(x^2 = \mathrm{R})$$

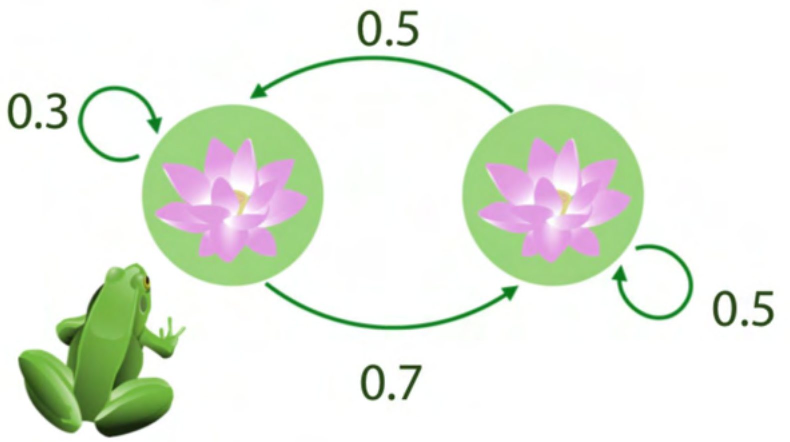
Markov Chains



	p(Left)	p(Right)
x^1	1	0
x^2	0.3	0.7
x^3	$0.3^2 + 0.7 \cdot 0.5$	$0.3 \cdot 0.7 + 0.7 \cdot 0.5$

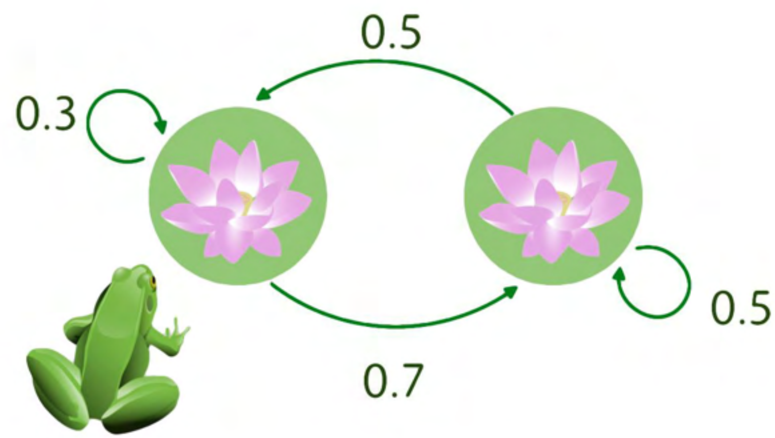
$$p(x^3) = p(x^3 \mid x^2 = \mathrm{L})p(x^2 = \mathrm{L}) + p(x^3 \mid x^2 = \mathrm{R})p(x^2 = \mathrm{R})$$

Markov Chains



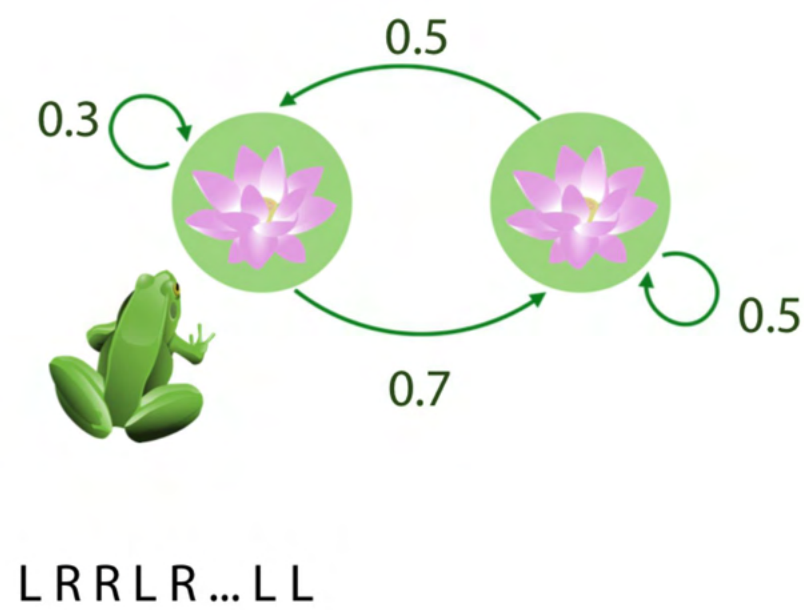
	p(Left)	p(Right)
x^1	1	0
x^2	0.3	0.7
x^3	0.44	0.56

Markov Chains

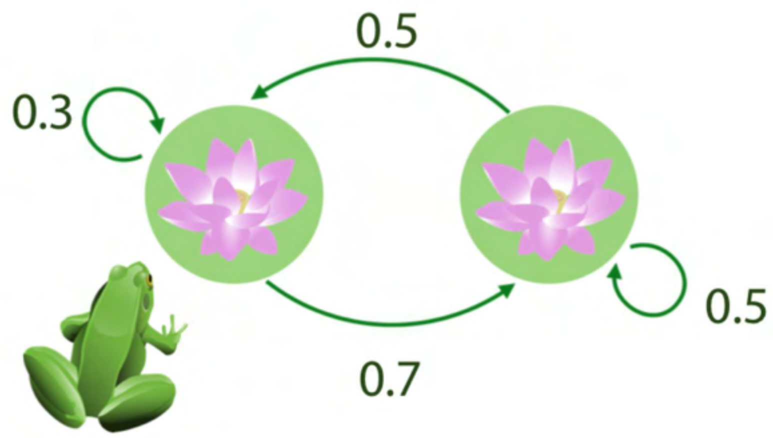


	p(Left)	p(Right)
x^1	1	0
x^2	0.3	0.7
x^3	0.44	0.56
$\cdots$	$\cdots$	$\cdots$
	≈ 0.42	≈ 0.58

Markov Chains

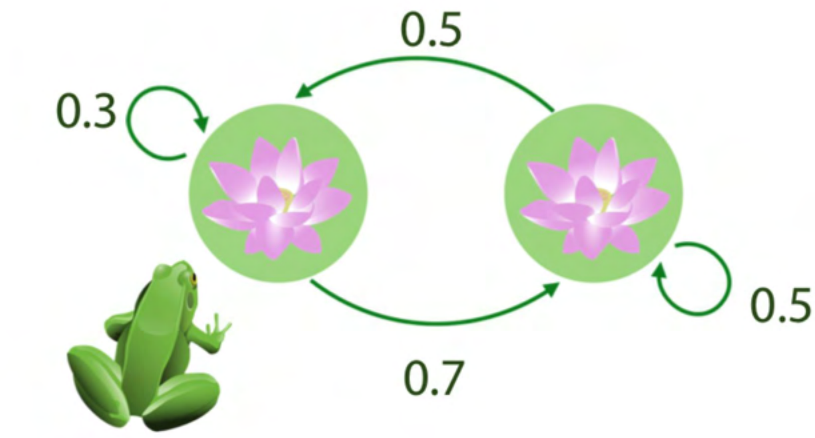


Markov Chains



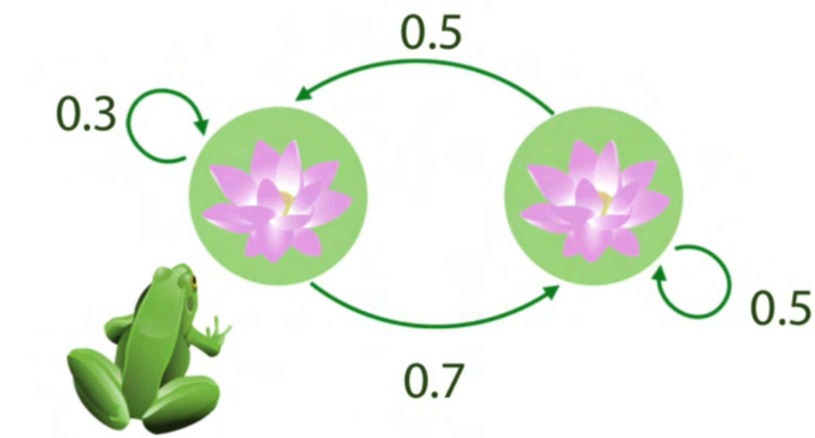
L R R L R … L **L**

Markov Chains



L R R L R ... L **L**
L R R L R ... L R

Markov Chains

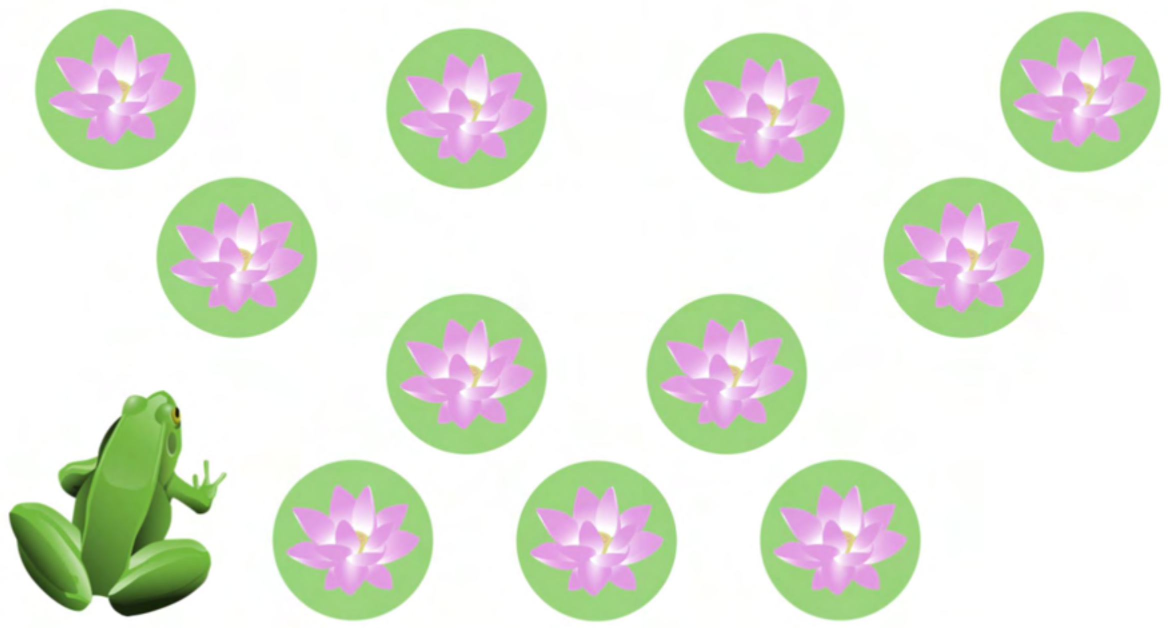


L R R L R ... L **L**
L R R L R ... L **R**
L R L R R ... R **R**
L R R L L ... L **R**
L L R L R ... R **L**

$$p(L) \approx 0.42$$

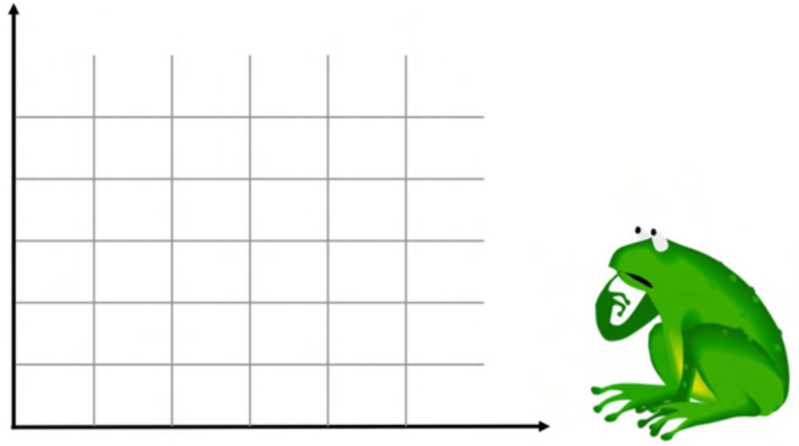
$$p(R) \approx 0.58$$

Markov Chains



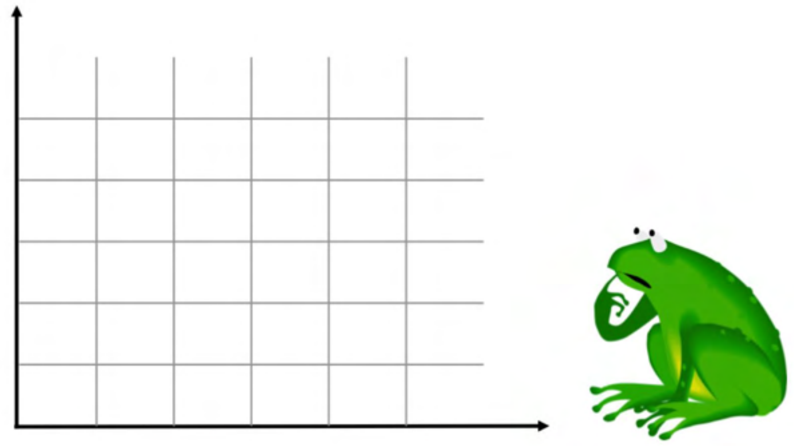
But what if there are 10 lilies? Or a billion?

Markov Chains



But what if there are 10 lilies? Or a billion?
Or maybe frog position is continuous?

Markov Chains



But what if there are 10 lilies? Or a billion?
Or maybe frog position is continuous?
You can still sample!

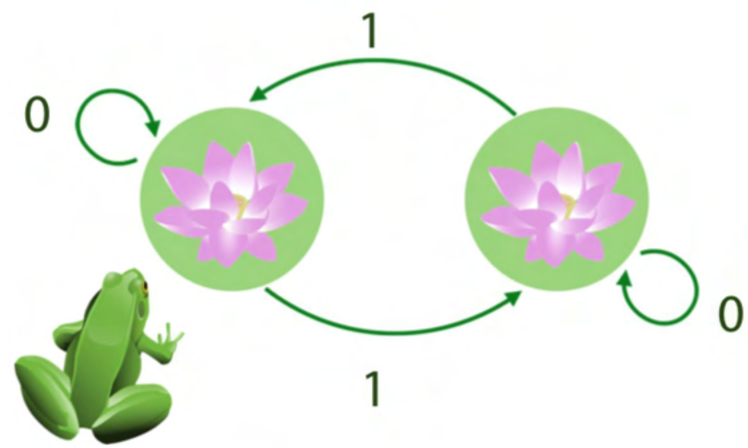
How to use Markov Chains?

- We want to sample from $p(x)$
- Build a Markov chain that converge to $p(x)$
- Start from any x^0
- For k = 0, 1, …

$$x^{k+1} \sim T(x^k \rightarrow x^{k+1})$$

- Eventually x^k will look like samples from $p(x)$

Do Markov Chains always converge?



	p(Left)	p(Right)
x^1	1	0
x^2	0	1
x^3	1	0
$\dots$	$\dots$	$\dots$

Does not converge

Stationary Distribution

Definition:

A distribution π is called stationary if

$$\pi(x') = \sum_{x} T(x \to x')\pi(x)$$

Convergence Theorem

Theorem:

If $T(x \to x') > 0$ for all x, x' then exists unique π:

$$\pi(x') = \sum_x T(x \to x')\pi(x)$$

And Markov chain converges to π from any starting point

Ch 3. Gibbs Sampling

Gibbs Sampling

$$p(x_1, x_2, x_3) = \frac{\widehat{p}(x_1, x_2, x_3)}{Z}$$

Gibbs Sampling

$$p(x_1, x_2, x_3) = \frac{\widehat{p}(x_1, x_2, x_3)}{Z}$$

Start with (x_1^0, x_2^0, x_3^0), e. g. $(0, 0, 0)$

Gibbs Sampling

$$p(x_1, x_2, x_3) = \frac{\widehat{p}(x_1, x_2, x_3)}{Z}$$

Start with (x_1^0, x_2^0, x_3^0), e. g. $(0, 0, 0)$

$$x_1^1 \sim p(x_1 \mid x_2 = x_2^0, x_3 = x_3^0)$$

Gibbs Sampling

$$p(x_1, x_2, x_3) = \frac{\widehat{p}(x_1, x_2, x_3)}{Z}$$

Start with (x_1^0, x_2^0, x_3^0), e. g. $(0, 0, 0)$

$$x_1^1 \sim p(x_1 \mid x_2 = x_2^0, x_3 = x_3^0)$$

$$= \frac{\widehat{p}(x_1, x_2^0, x_3^0)}{Z_1}$$

Gibbs Sampling

$$p(x_1, x_2, x_3) = \frac{\widehat{p}(x_1, x_2, x_3)}{Z}$$

Start with (x_1^0, x_2^0, x_3^0), e. g. $(0, 0, 0)$

$$x_1^1 \sim p(x_1 \mid x_2 = x_2^0, x_3 = x_3^0)$$

$$x_2^1 \sim p(x_2 \mid x_1 = x_1^1, x_3 = x_3^0)$$

Gibbs Sampling

$$p(x_1, x_2, x_3) = \frac{\widehat{p}(x_1, x_2, x_3)}{Z}$$

Start with (x_1^0, x_2^0, x_3^0), e. g. $(0, 0, 0)$

$$x_1^1 \sim p(x_1 \mid x_2 = x_2^0, x_3 = x_3^0)$$

$$x_2^1 \sim p(x_2 \mid x_1 = x_1^1, x_3 = x_3^0)$$

$$x_3^1 \sim p(x_3 \mid x_1 = x_1^1, x_2 = x_2^1)$$

Gibbs Sampling

$$p(x_1, x_2, x_3) = \frac{\widehat{p}(x_1, x_2, x_3)}{Z}$$

Start with (x_1^0, x_2^0, x_3^0), e. g. $(0, 0, 0)$

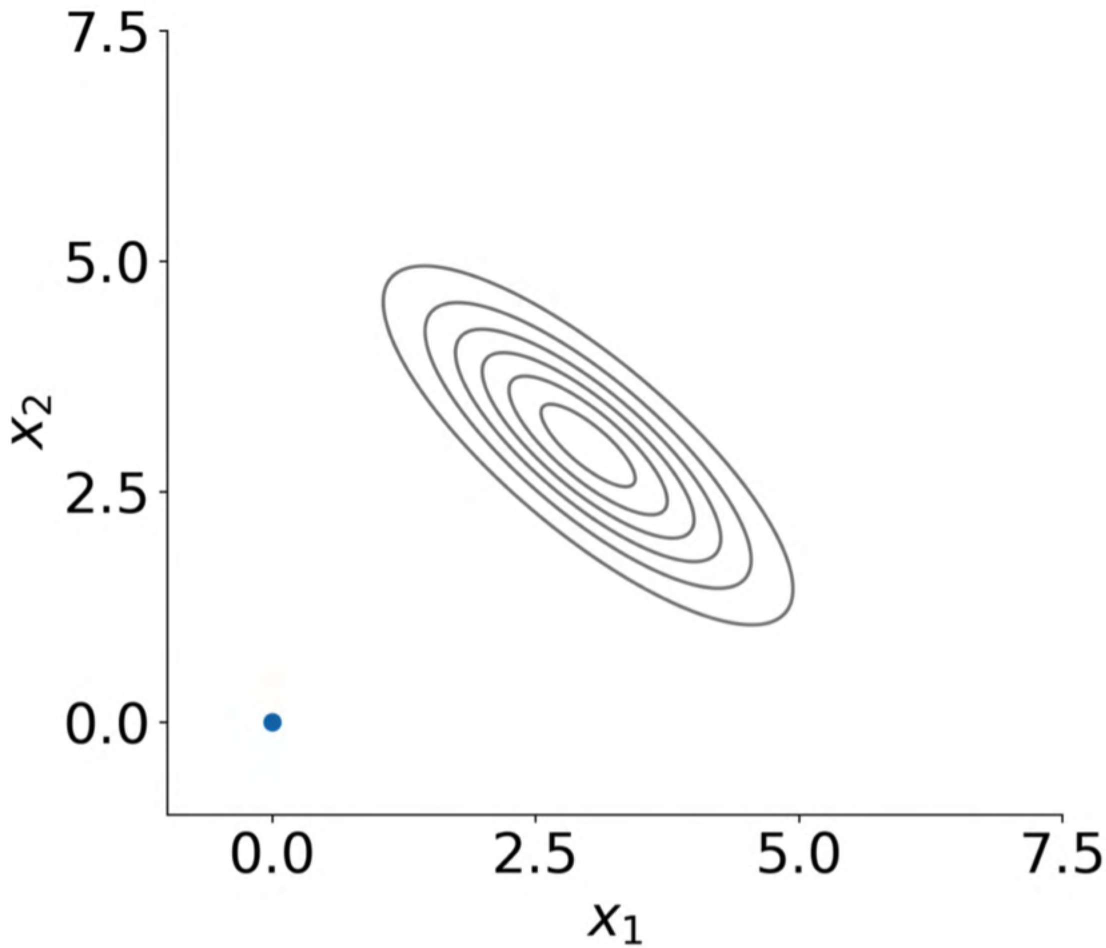
For k = 0, 1, …

$$x_1^{k+1} \sim p(x_1 \mid x_2 = x_2^k, x_3 = x_3^k)$$

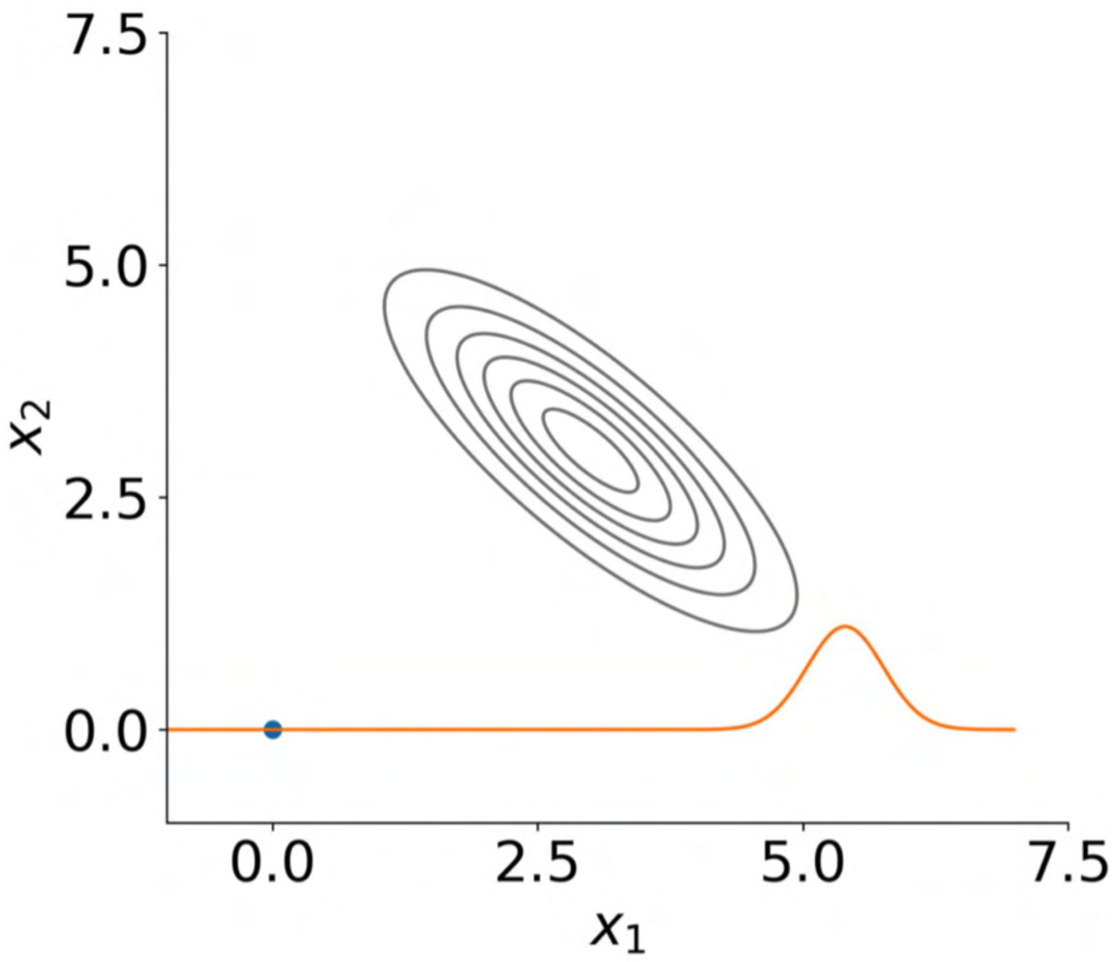
$$x_2^{k+1} \sim p(x_2 \mid x_1 = x_1^{k+1}, x_3 = x_3^k)$$

$$x_3^{k+1} \sim p(x_3 \mid x_1 = x_1^{k+1}, x_2 = x_2^{k+1})$$

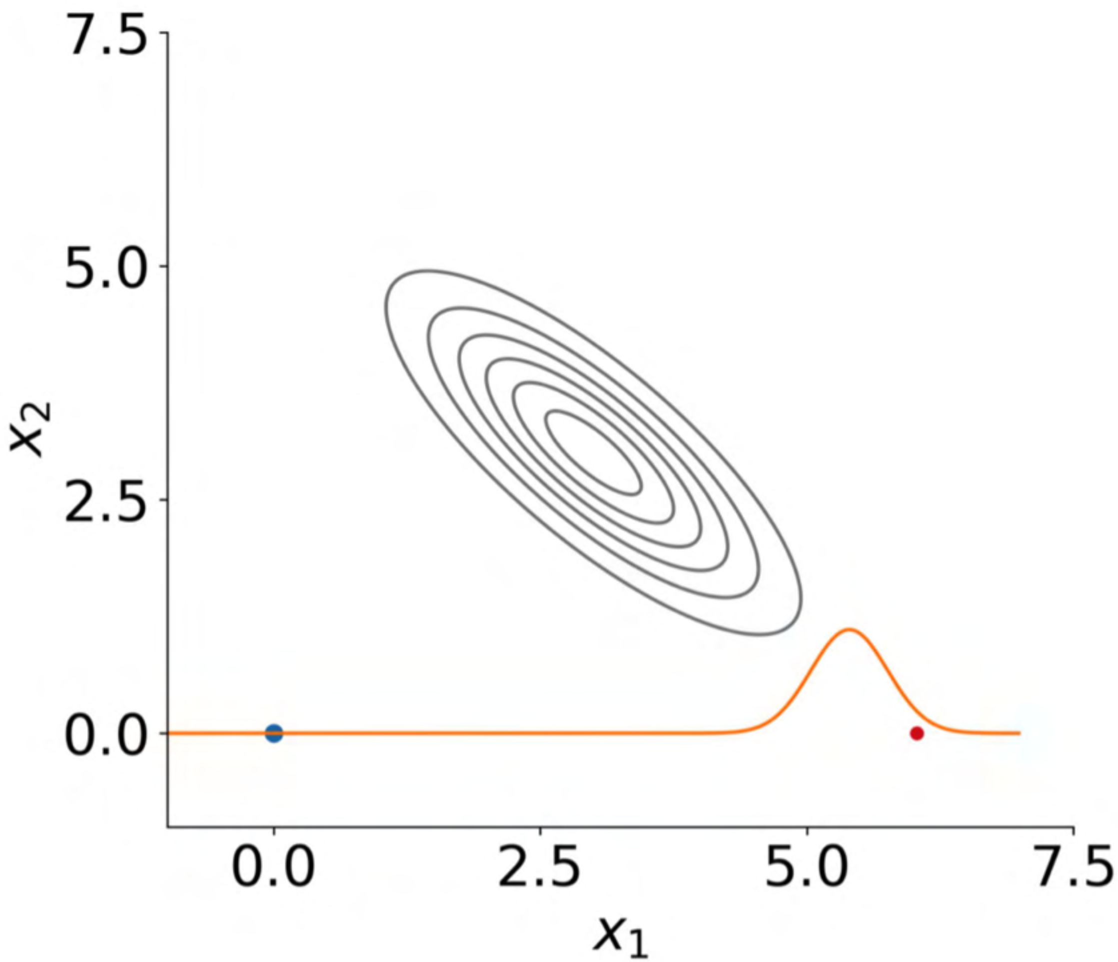
Gibbs Sampling - Demo



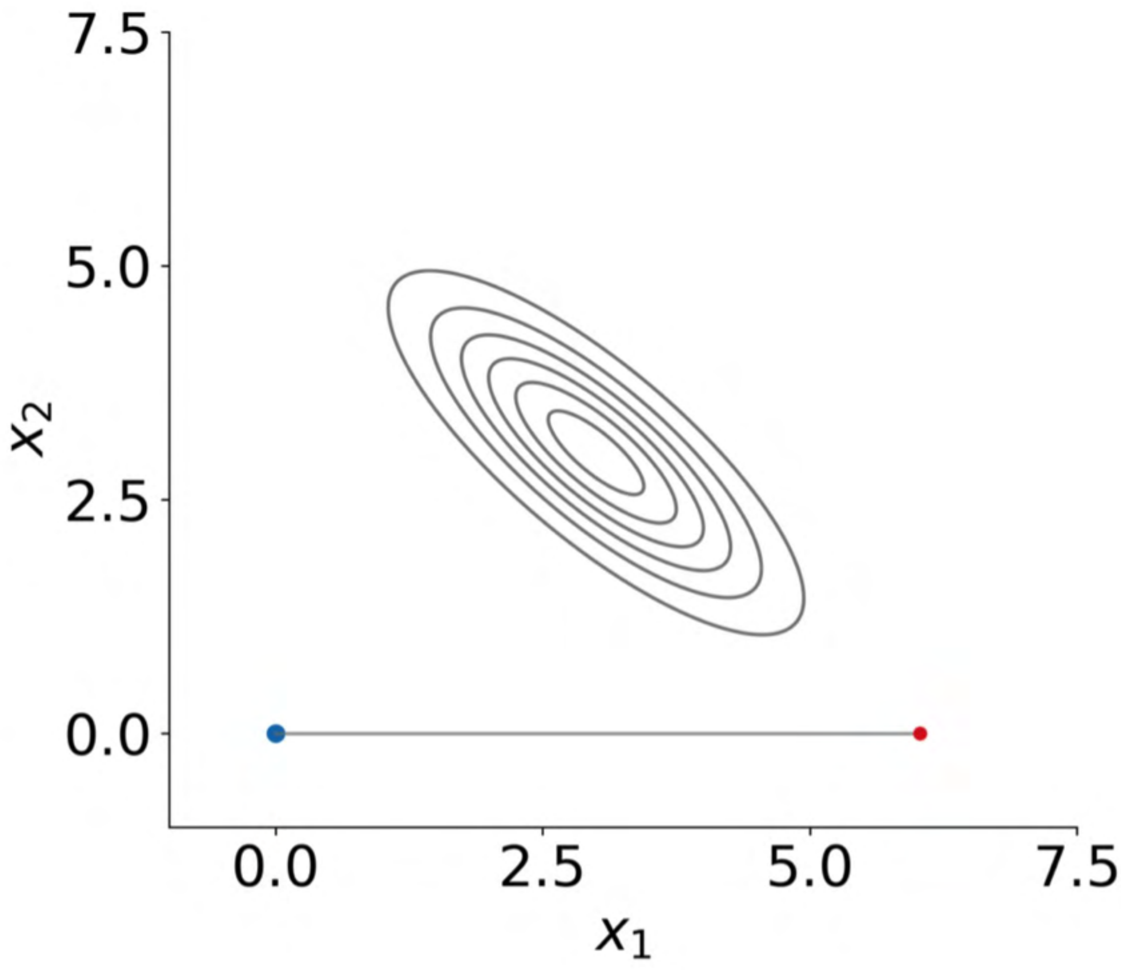
Gibbs Sampling - Demo



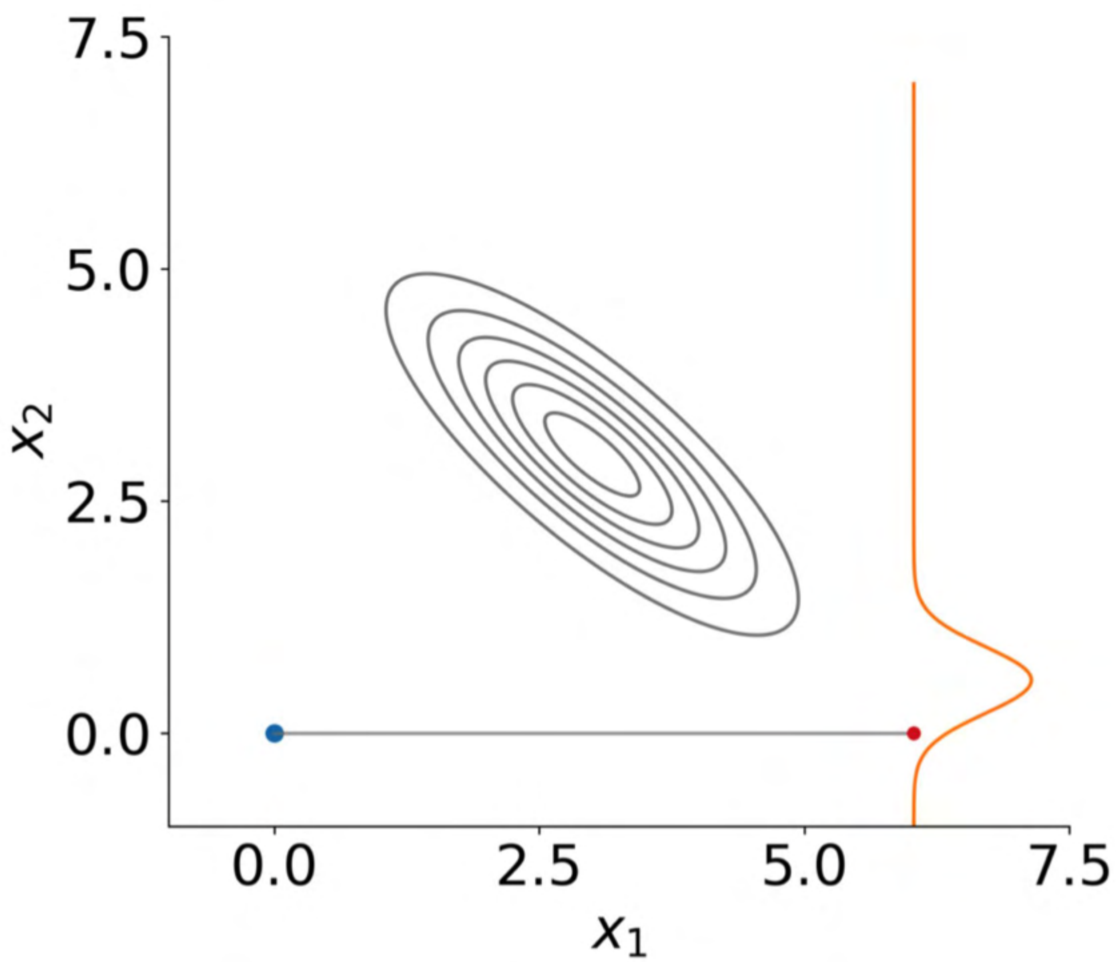
Gibbs Sampling - Demo



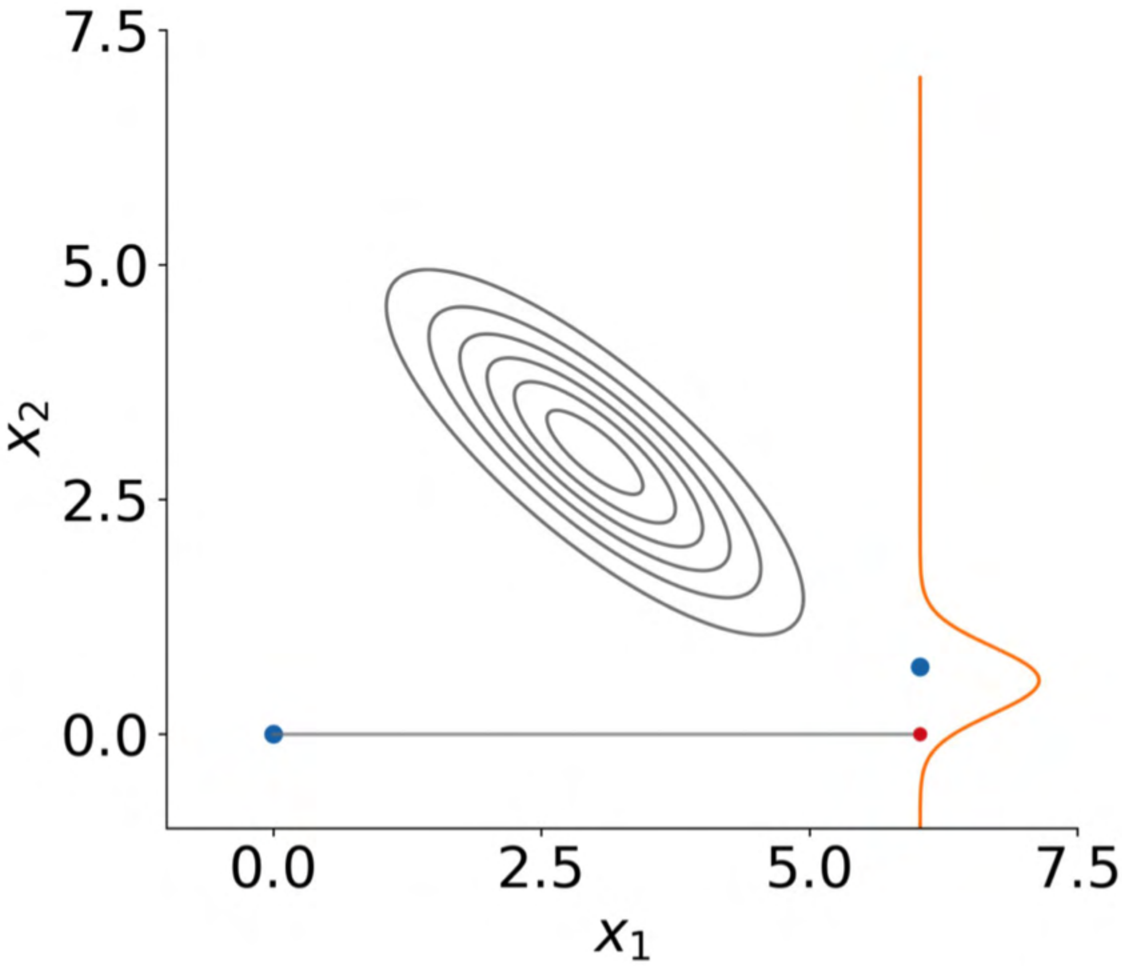
Gibbs Sampling - Demo



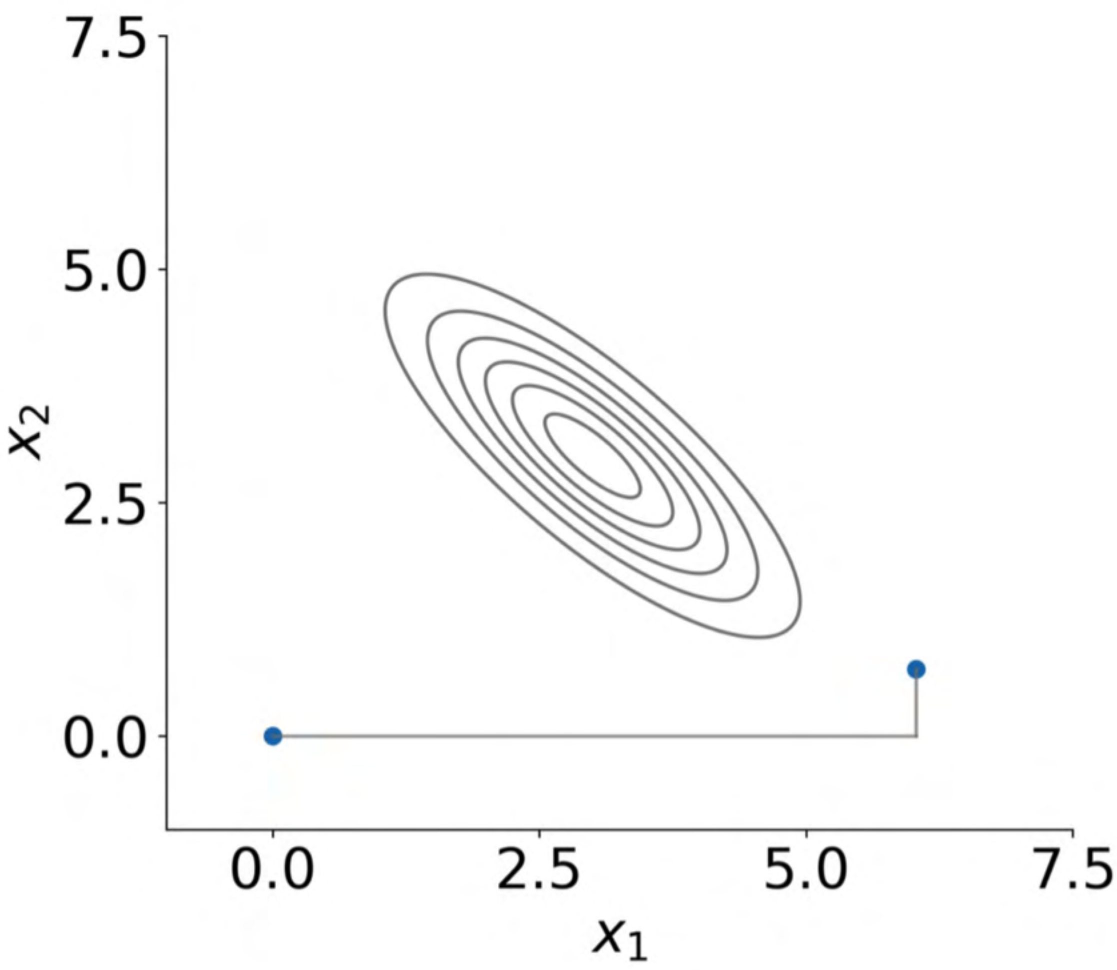
Gibbs Sampling - Demo



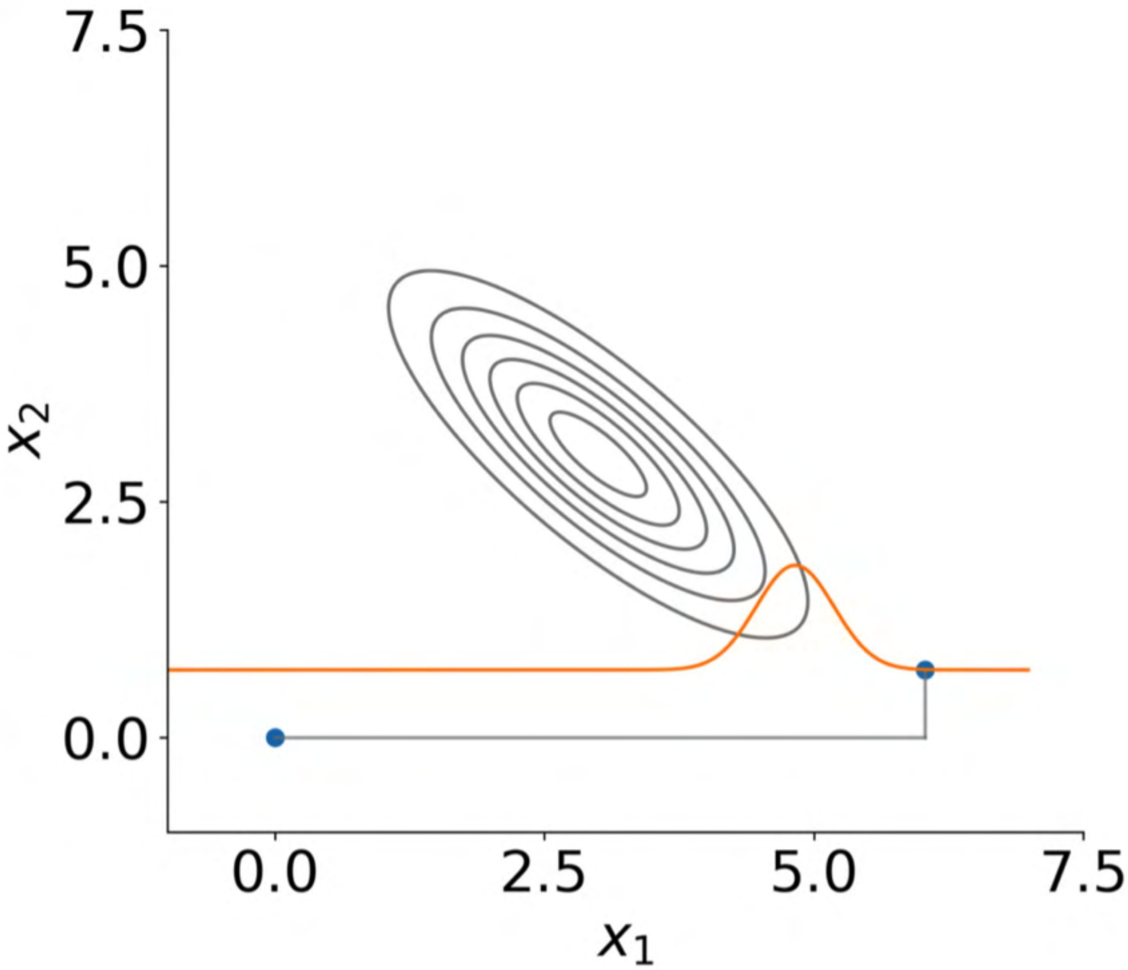
Gibbs Sampling - Demo



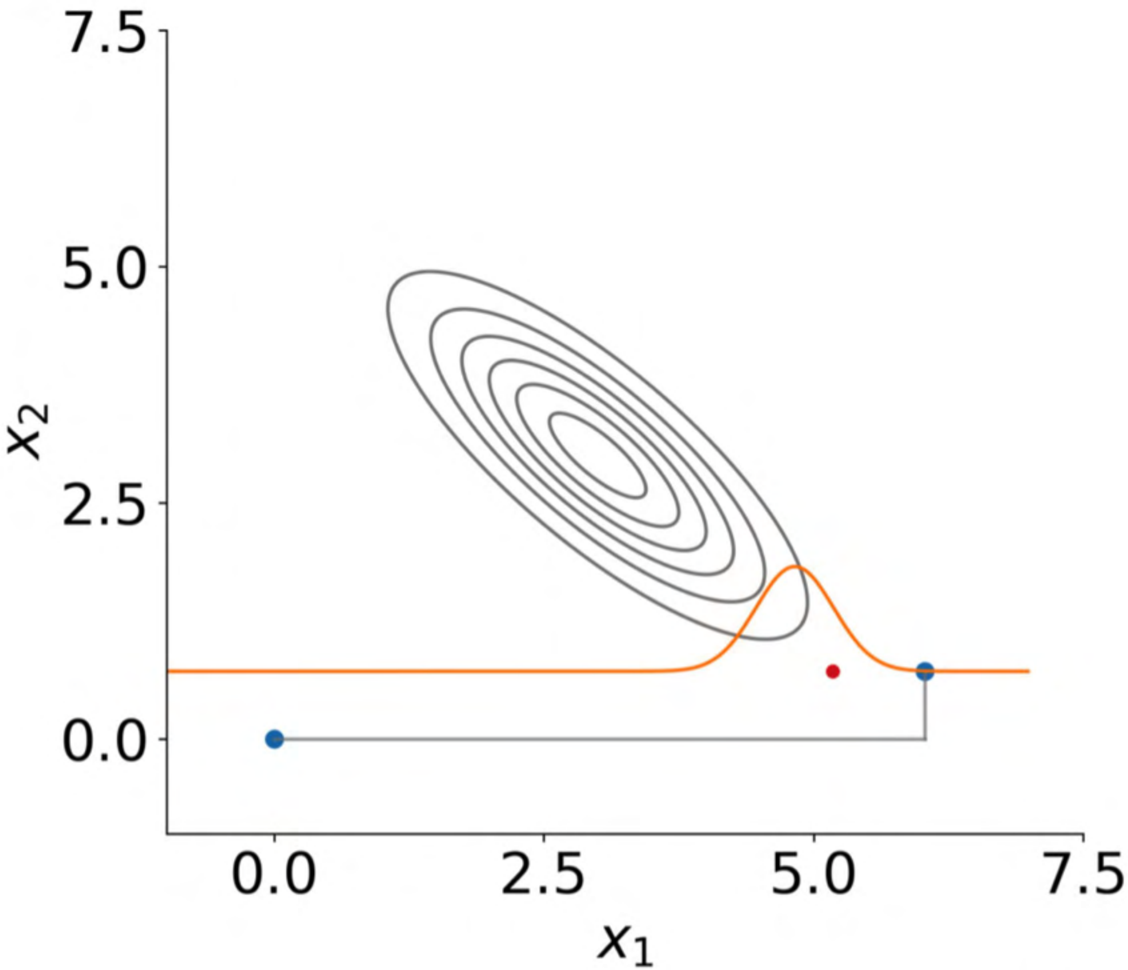
Gibbs Sampling - Demo



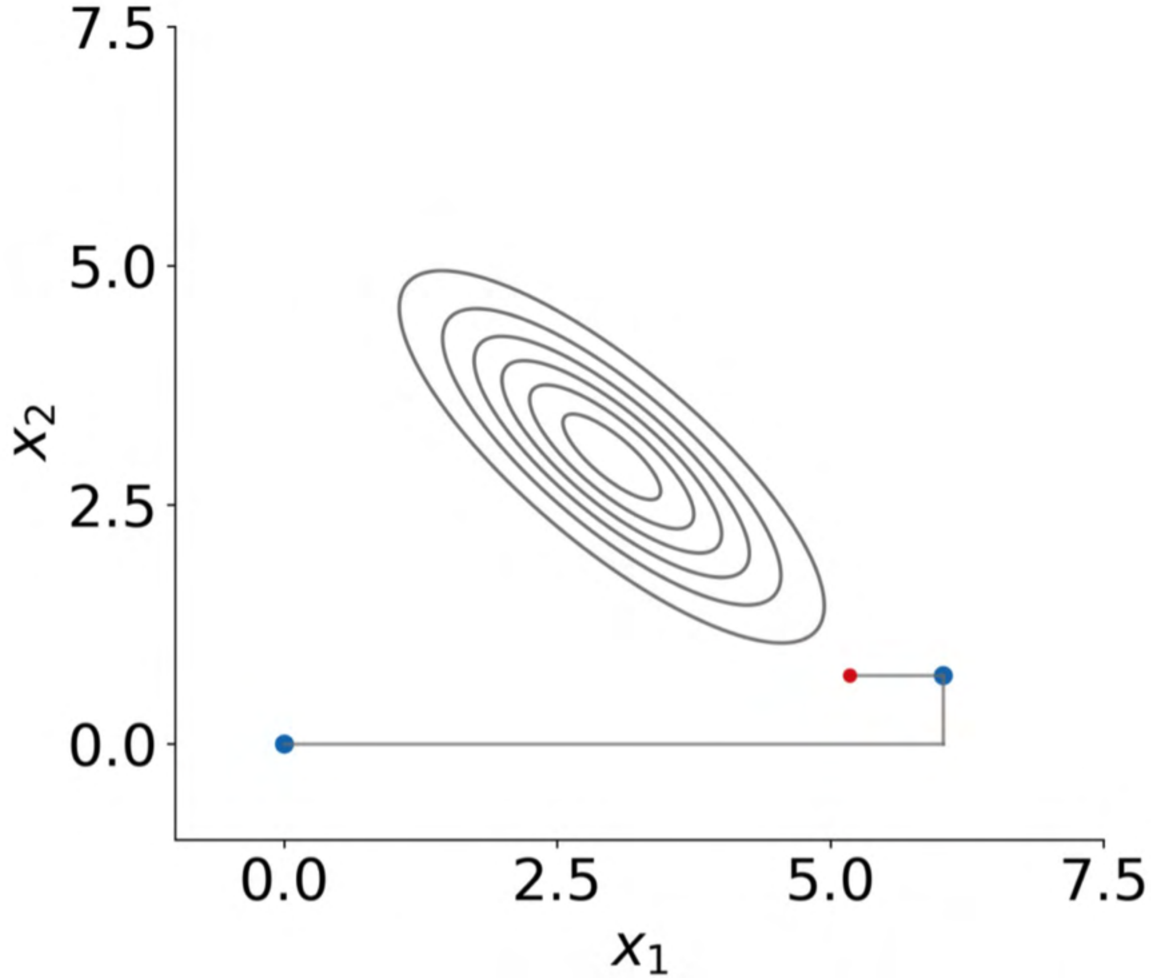
Gibbs Sampling - Demo



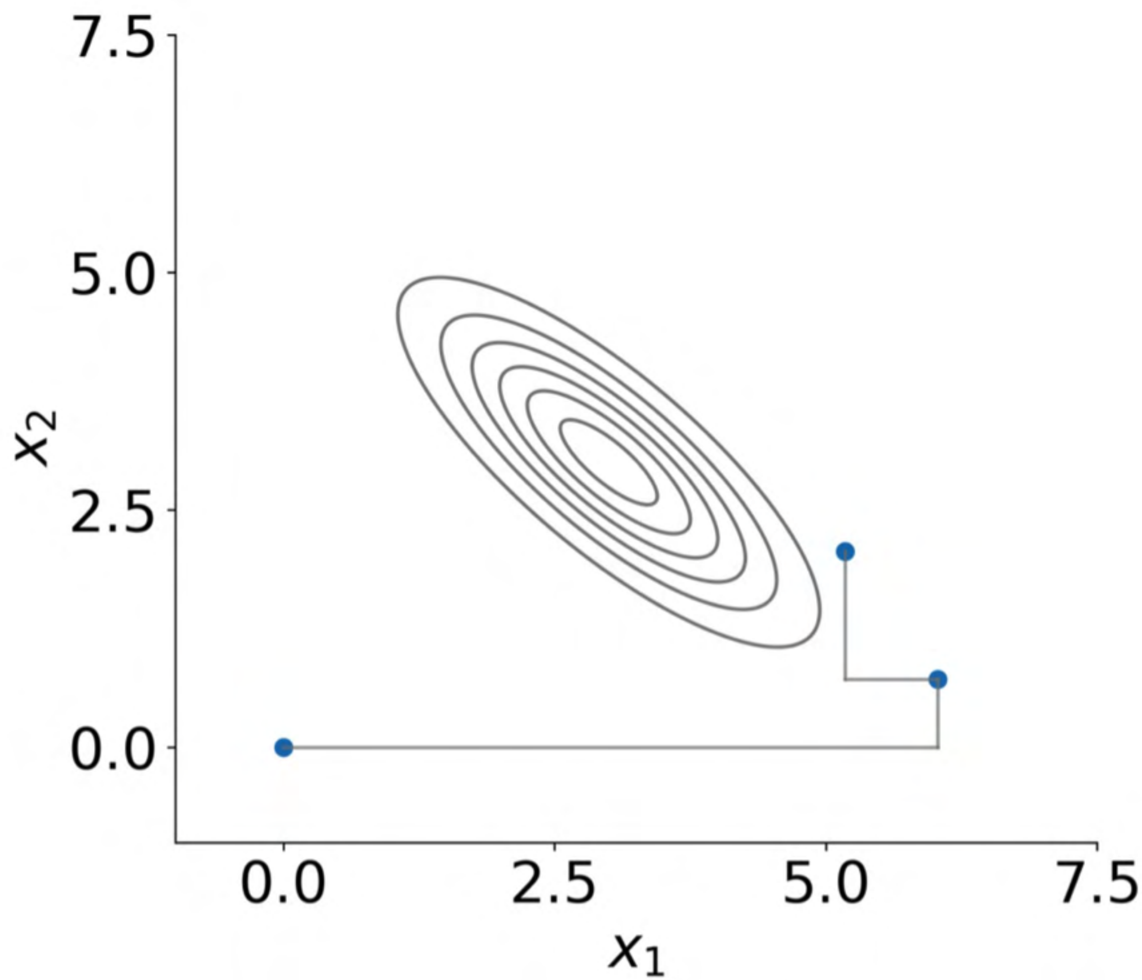
Gibbs Sampling - Demo



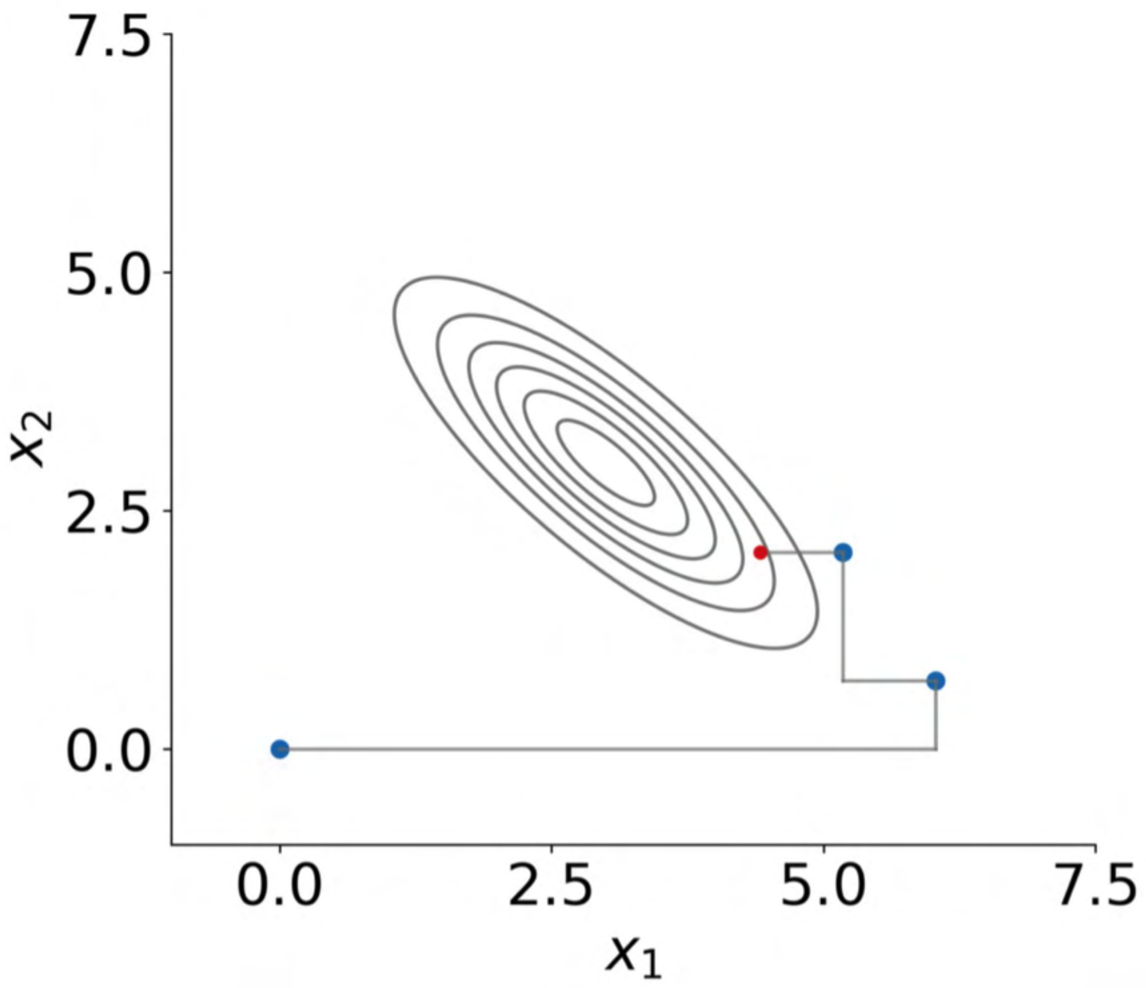
Gibbs Sampling - Demo



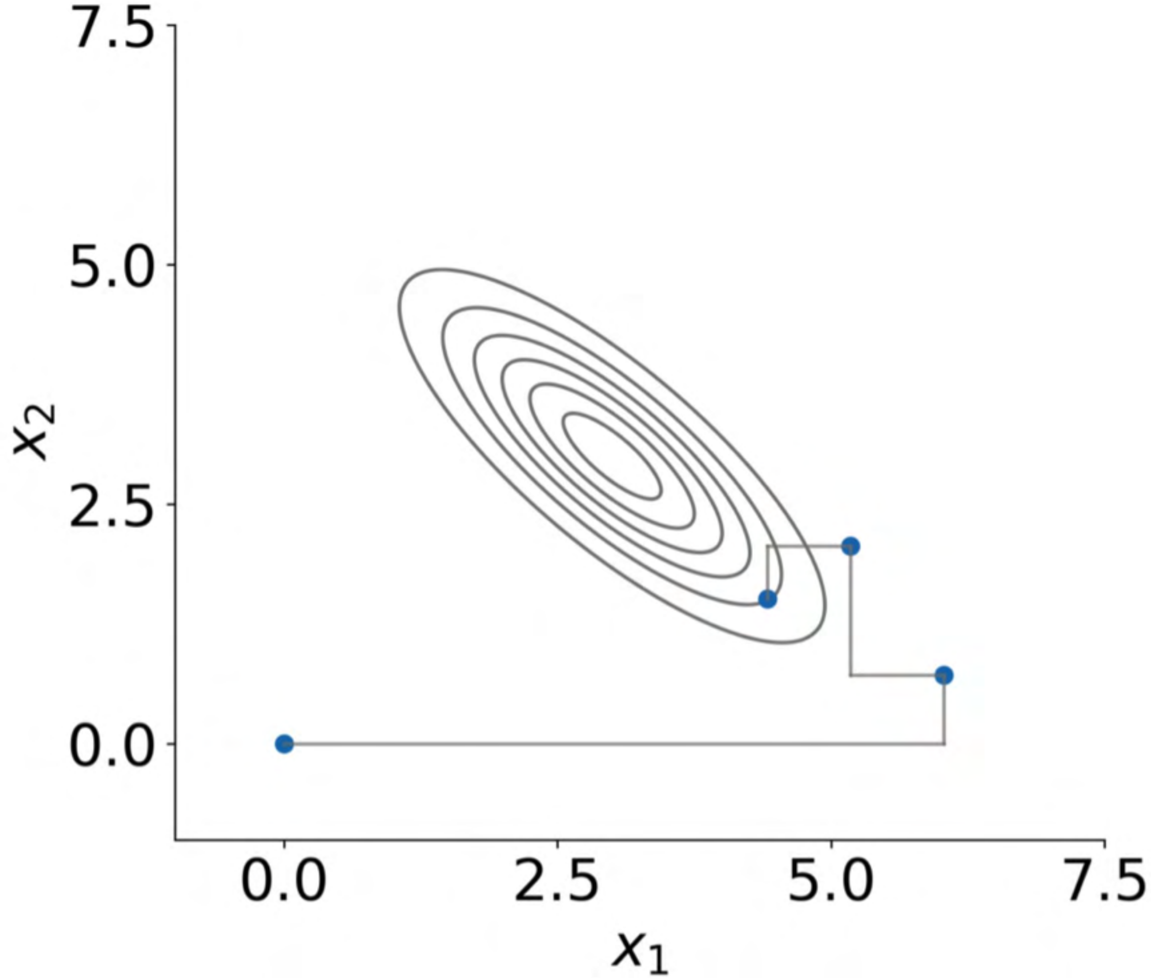
Gibbs Sampling - Demo



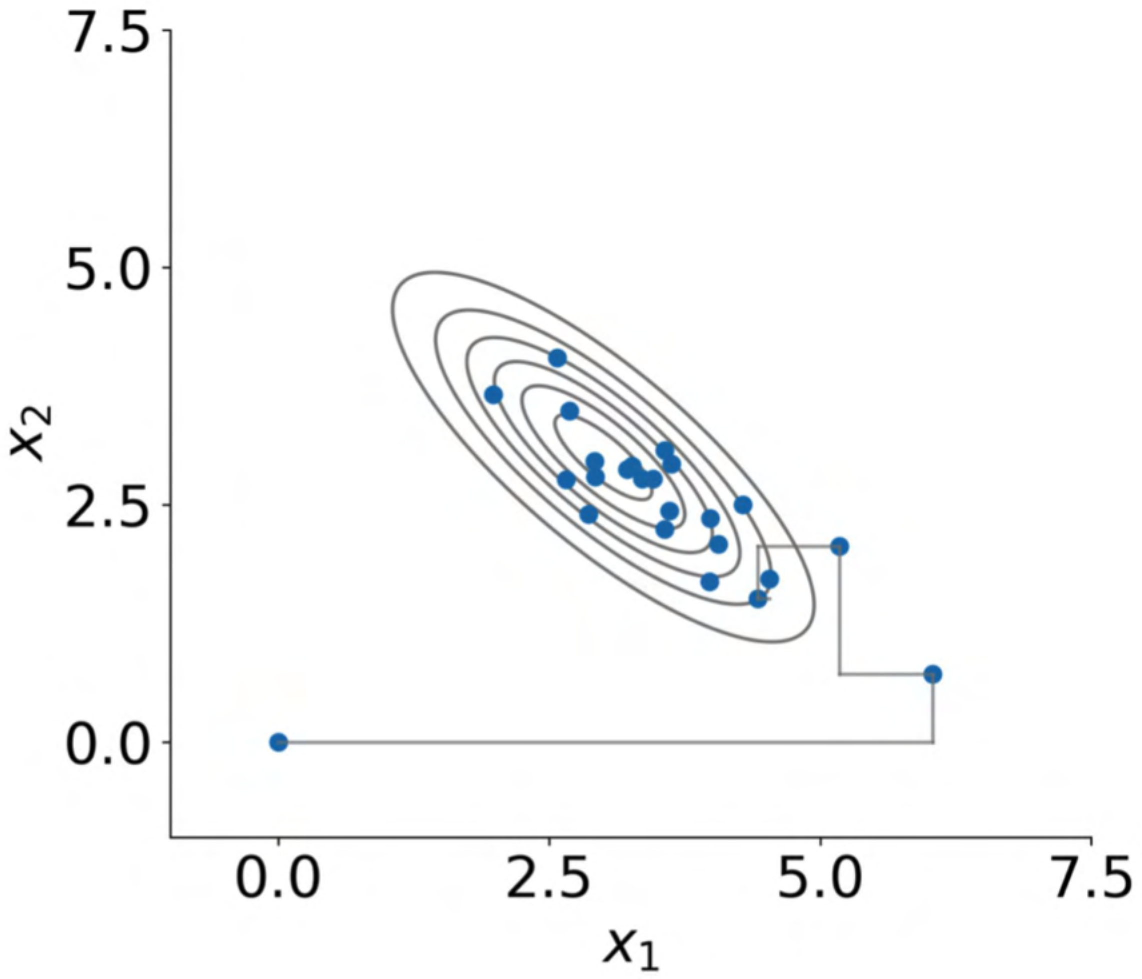
Gibbs Sampling - Demo



Gibbs Sampling - Demo



Gibbs Sampling - Demo



Gibbs Sampling - Summary

Pros:

- Reduce multidimensional sampling to sequence of 1d samplings
- A few lines of code

Cons:

- Highly correlated samples
  - Samples are similar to each others
- Slow Convergence (Mixing)
- Not Parallel

Ch 5. Metropolis-Hastings

Metropolis-Hastings

- Sometimes Gibbs samples are too correlated
- Apply Rejection Sampling to Markov Chains

Metropolis-Hastings

For k = 1, 2, …

- Sample x' from a wrong $Q(x^k \to x')$

Metropolis-Hastings

For k = 1, 2, …

- Sample x' from a wrong $Q(x^k \to x')$
- Accept proposal x' with probability $A(x^k \to x')$
- Otherwise stay at x^k

$$x^{k+1} = x^k$$

Metropolis-Hastings

For k = 1, 2, …

- Sample x' from a wrong $Q(x^k \to x')$
- Accept proposal x' with probability $A(x^k \to x')$
- Otherwise stay at x^k

$$x^{k+1} = x^k$$

$$T(x \to x') = Q(x \to x')A(x \to x') \quad \text{for all } x \neq x'$$

Metropolis-Hastings

For k = 1, 2, …

- Sample x' from a wrong $Q(x^k \to x')$
- Accept proposal x' with probability $A(x^k \to x')$
- Otherwise stay at x^k

$$x^{k+1} = x^k$$

$$T(x \to x') = Q(x \to x')A(x \to x') \quad \text{for all } x \neq x'$$

$$T(x \to x) = Q(x \to x)A(x \to x) + \sum_{x' \neq x} Q(x \to x')(1 - A(x \to x'))$$

Metropolis-Hastings

For k = 1, 2, …

- Sample x' from a wrong $Q(x^k \to x')$
- Accept proposal x' with probability $A(x^k \to x')$
- Otherwise stay at x^k

$$x^{k+1} = x^k$$

$$T(x \to x') = Q(x \to x')A(x \to x') \quad \text{for all } x \neq x'$$

$$T(x \to x) = Q(x \to x)A(x \to x) + \sum_{x' \neq x} Q(x \to x')(1 - A(x \to x'))$$

How to choose A: $\pi(x') = \sum_x \pi(x)T(x \to x')$

Detailed Balance

- Sufficient Condition for Stationary Distribution

If $$\pi(x)T(x \to x') = \pi(x')T(x' \to x)$$

Then $$\pi(x') = \sum_{x} \pi(x)T(x \to x')$$

Detailed Balance

- Sufficient Condition for Stationary Distribution

If $\pi(x)T(x \to x') = \pi(x')T(x' \to x)$

Then $\pi(x') = \sum_x \pi(x)T(x \to x')$

Proof

$$\sum_x \pi(x)T(x \to x') = \sum_x \pi(x')T(x' \to x)$$

$$= \pi(x') \sum_x T(x' \to x)$$

$$= \pi(x')$$

Choosing a Critic (Accepting Prob.)

For k = 1, 2, …

- Sample x' from a wrong $Q(x^k \to x')$
- Accept proposal x' with probability $A(x^k \to x')$
- Otherwise stay at x^k

$$x^{k+1} = x^k$$

$T(x \to x') = Q(x \to x')A(x \to x')$ for all $x \neq x'$

$$T(x' \to x') = Q(x' \to x')$$

How to choose A:

$$\pi(x') = \sum_x \pi(x) T(x \to x')$$

Choosing a Critic (Accepting Prob.)

For k = 1, 2, …

- Sample x' from a wrong $Q(x^k \to x')$
- Accept proposal x' with probability $A(x^k \to x')$
- Otherwise stay at x^k

$$x^{k+1} = x^k$$

$$T(x \to x') = Q(x \to x')A(x \to x') \text{ for all } x \neq x'$$

$$T(x' \to x') = Q(x' \to x')$$

How to choose A:

$$\pi(x)T(x \to x') = \pi(x')T(x' \to x)$$

Metropolis Hastings

For k = 1, 2, …

- Sample x' from a wrong $Q(x^k \to x')$
- Accept proposal x' with probability $A(x^k \to x')$
- Otherwise stay at x^k

$$x^{k+1} = x^k$$

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

Metropolis Hastings

For k = 1, 2, …

- Sample x' from a wrong $Q(x^k \to x')$
- Accept proposal x' with probability $A(x^k \to x')$
- Otherwise stay at x^k

$$x^{k+1} = x^k$$

$$A(x \to x') = \min\left(1, \frac{\widehat{\pi}(x')Q(x' \to x)}{\widehat{\pi}(x)Q(x \to x')}\right)$$

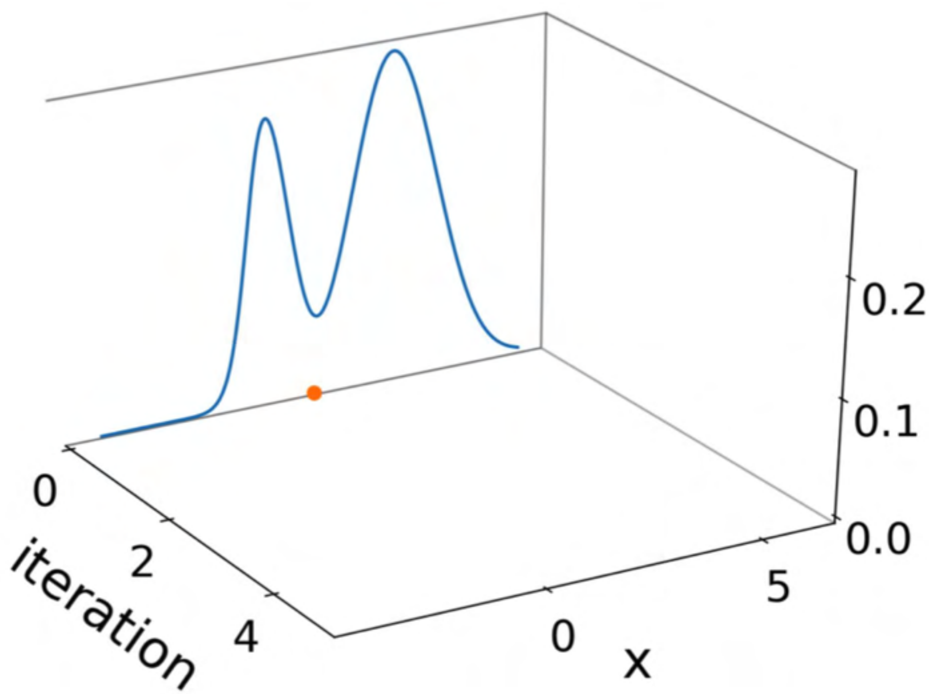
Choice of Q (Proposal Distribution)

$$Q(x \to x') > 0$$

Opposing forces:

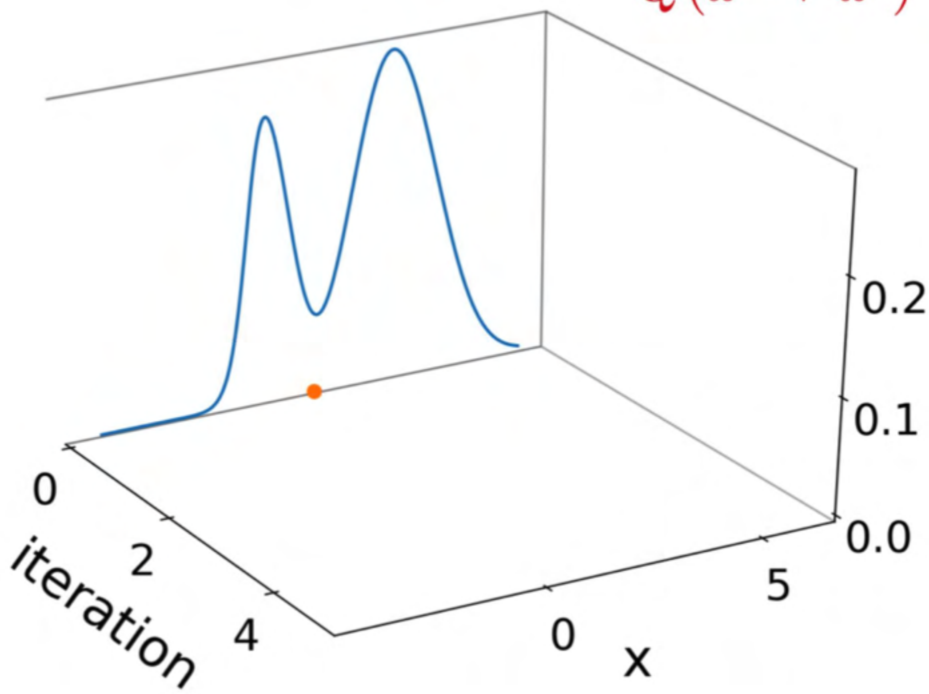
- Q should spread out, to improve mixing and reduce correlation
- But then acceptance probability is often low

Metropolis Hastings - Demo



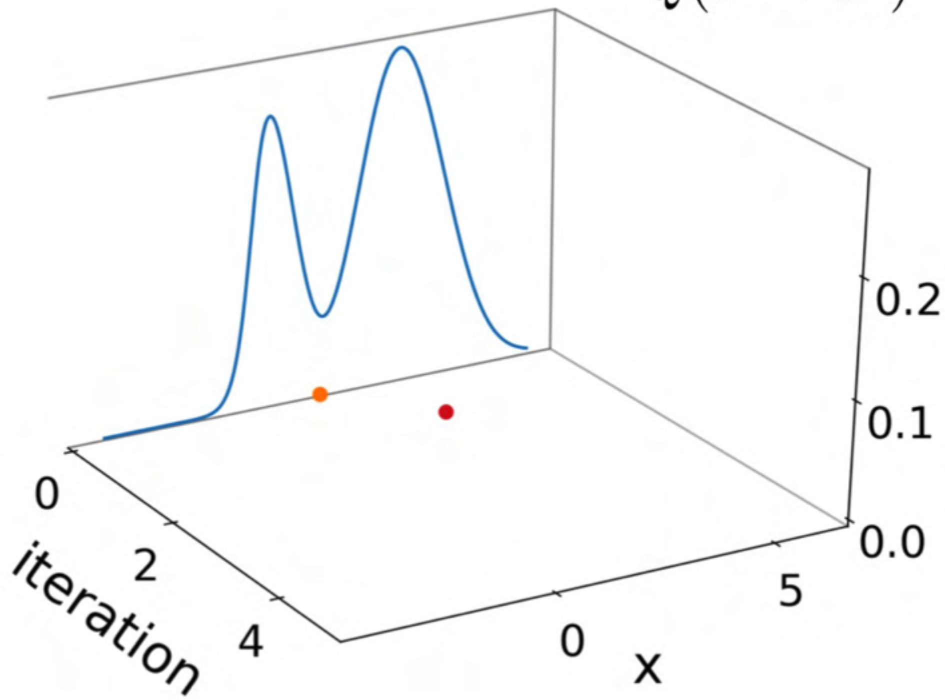
Metropolis Hastings - Demo

$$Q(x \to x') = \mathcal{N}(x, 1)$$



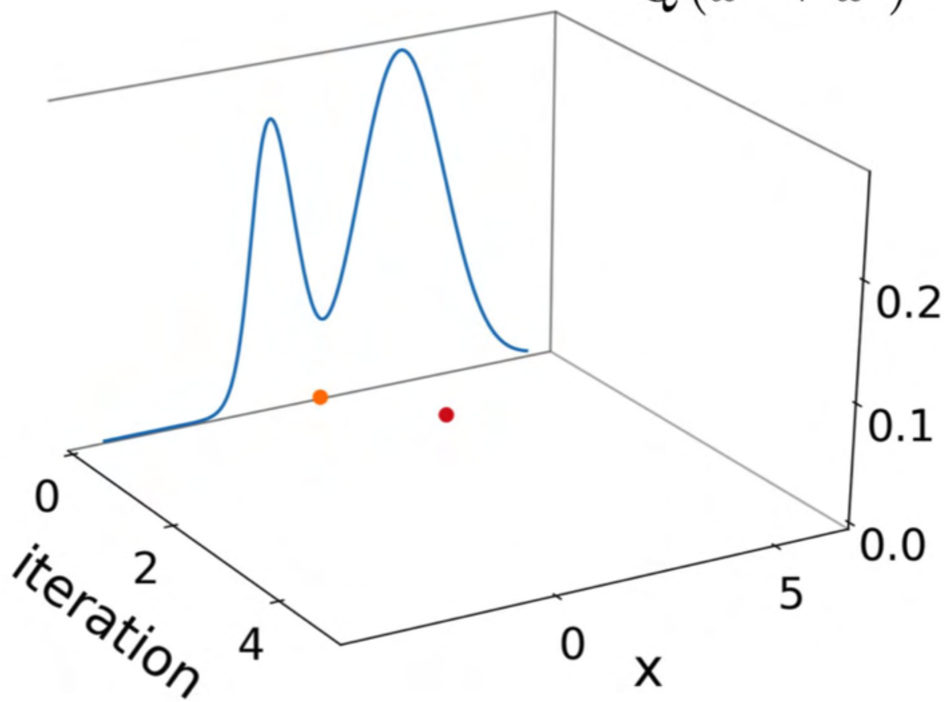
Metropolis Hastings - Demo

$Q(x \to x') = \mathcal{N}(x, 1)$



Metropolis Hastings - Demo

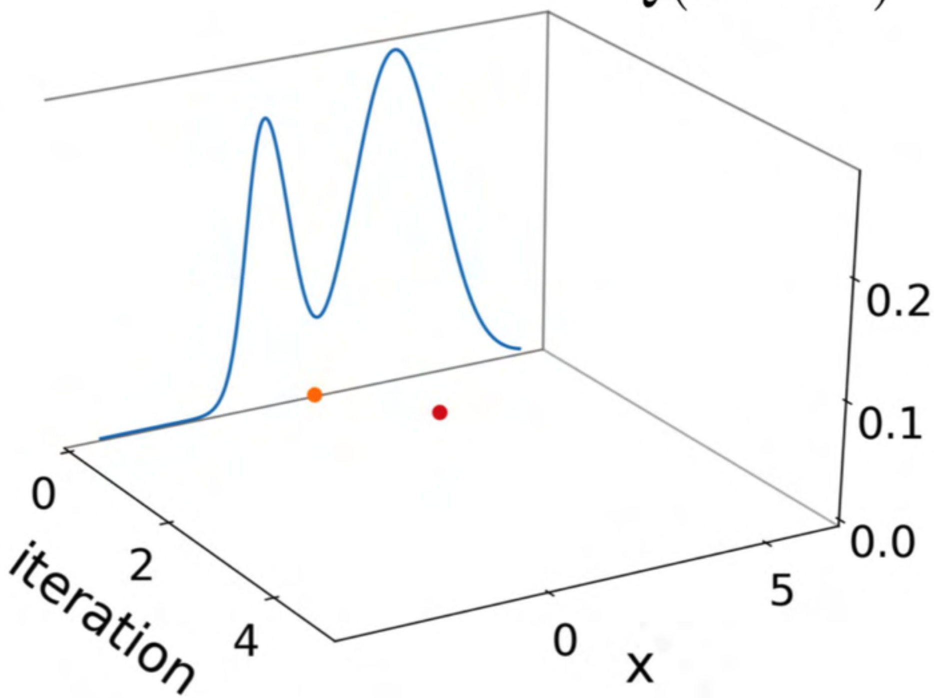
$$Q(x \to x') = \mathcal{N}(x, 1)$$



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

Metropolis Hastings - Demo

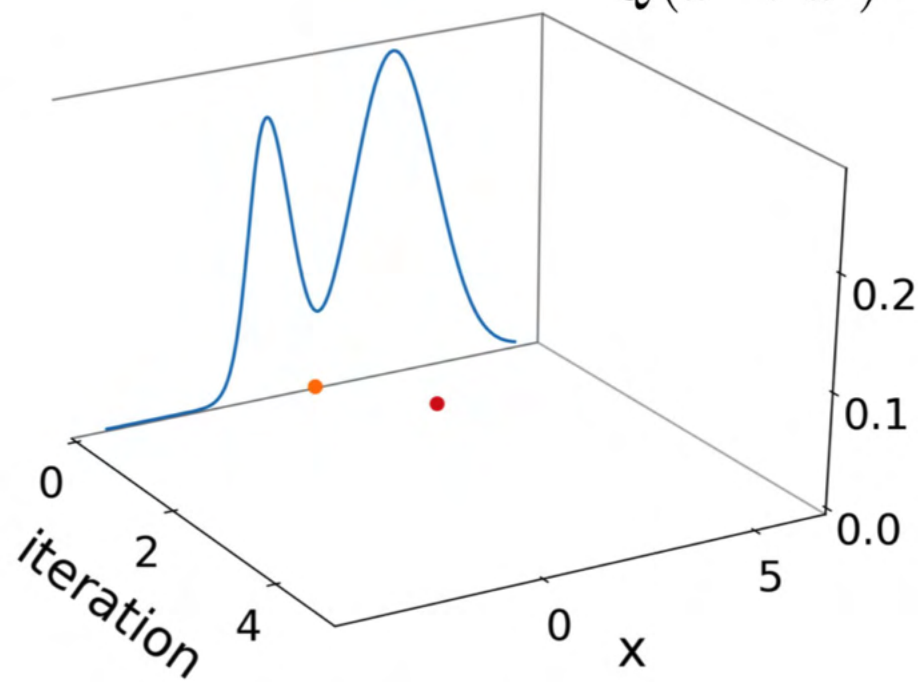
$$Q(x \to x') = \mathcal{N}(x, 1)$$



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

Metropolis Hastings - Demo

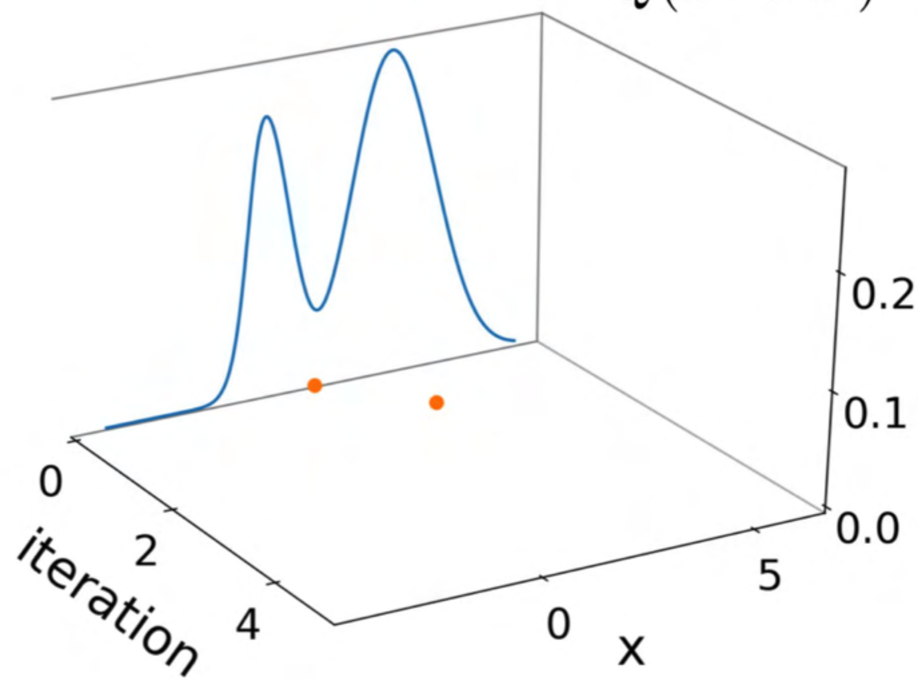
$$Q(x \to x') = \mathcal{N}(x, 1)$$



$$A(x \to x') = \min\left(1, \frac{0.27}{0.07}\right) = \min(1, 3.87)$$

Metropolis Hastings - Demo

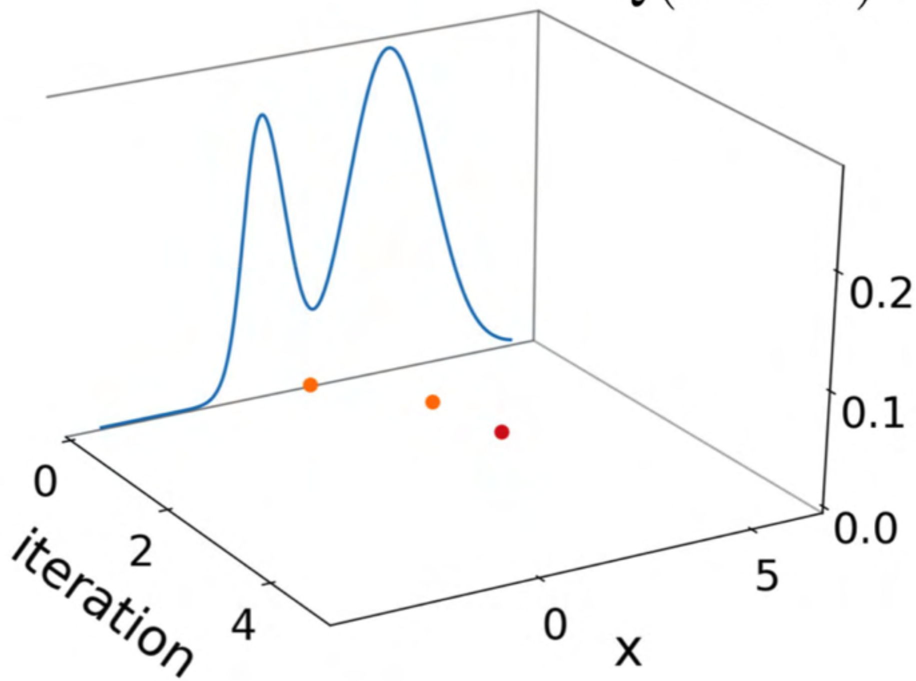
$$Q(x \to x') = \mathcal{N}(x, 1)$$



$$A(x \to x') = \min\left(1, \frac{0.27}{0.07}\right) = \min(1, 3.87)$$

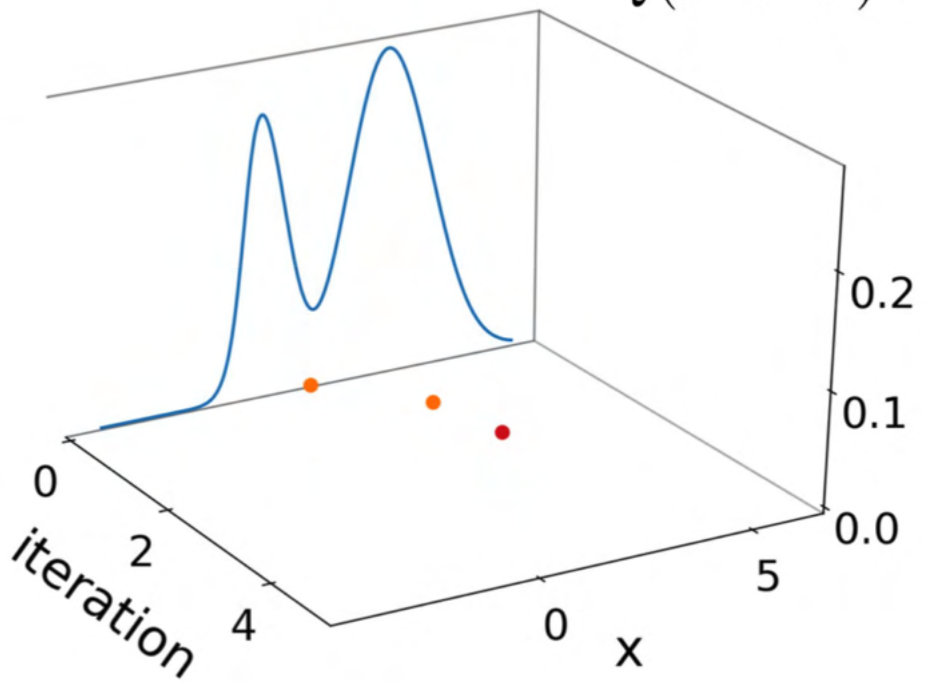
Metropolis Hastings - Demo

$$Q(x \to x') = \mathcal{N}(x, 1)$$



Metropolis Hastings - Demo

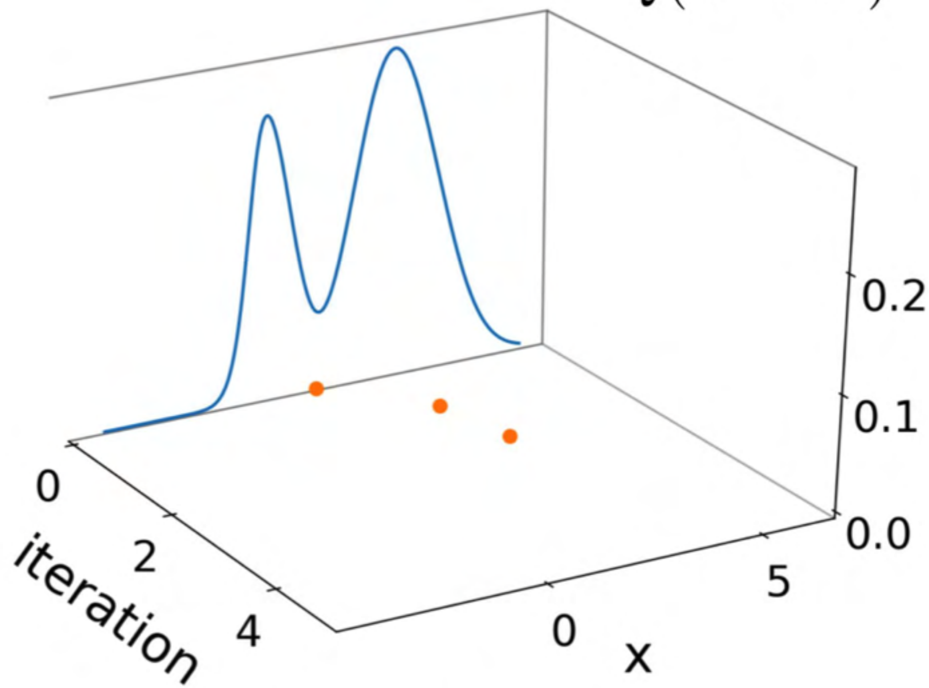
$$Q(x \to x') = \mathcal{N}(x, 1)$$



$$A(x \to x') = \min\left(1, \frac{0.28}{0.27}\right) = \min(1, 1.01)$$

Metropolis Hastings - Demo

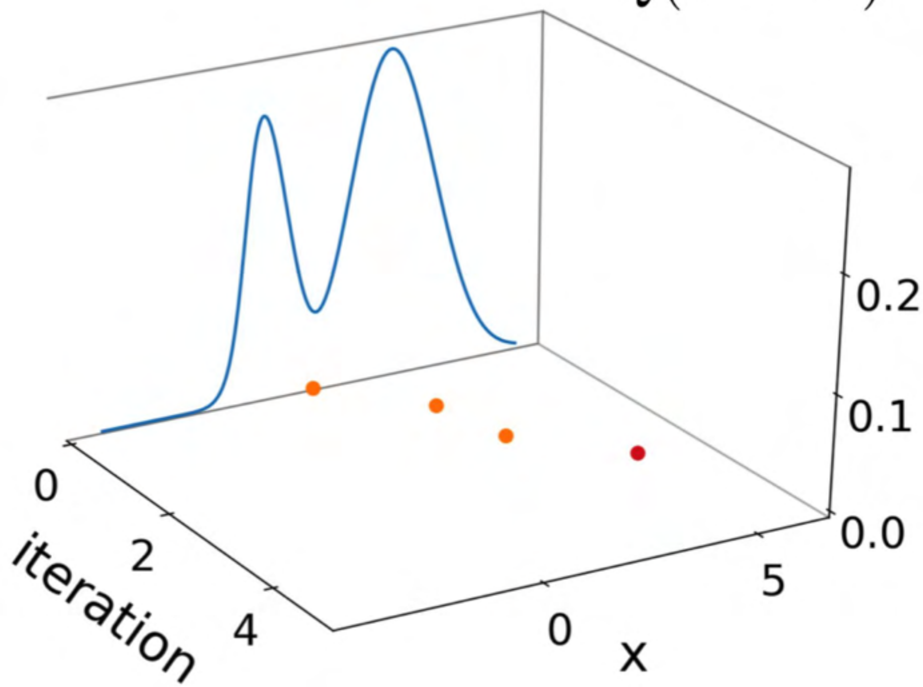
$$Q(x \to x') = \mathcal{N}(x, 1)$$



$$A(x \to x') = \min\left(1, \frac{0.28}{0.27}\right) = \min(1, 1.01)$$

Metropolis Hastings - Demo

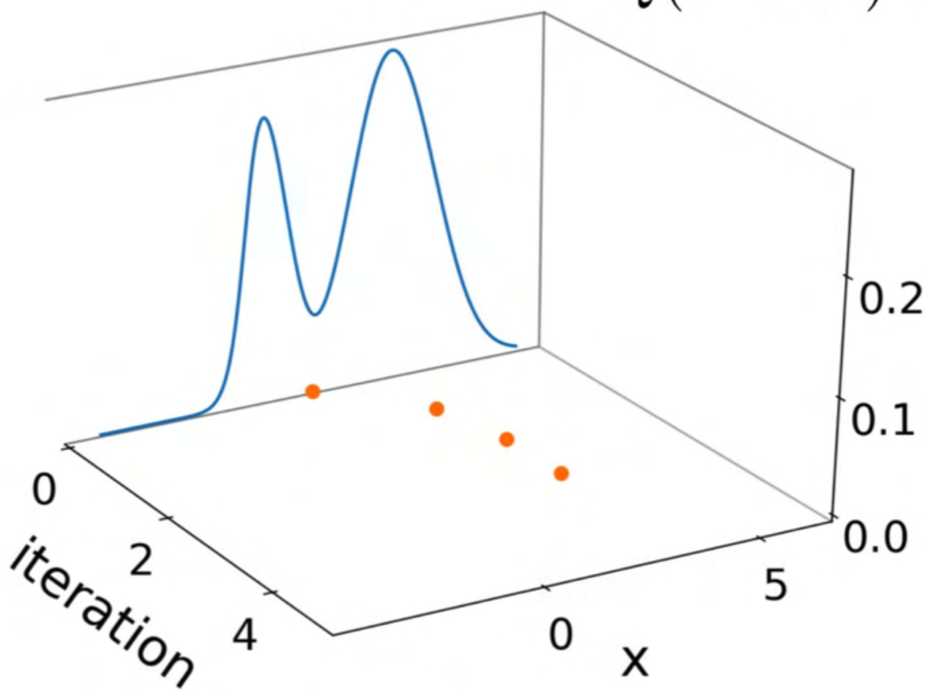
$$Q(x \to x') = \mathcal{N}(x, 1)$$



$$A(x \to x') = \min\left(1, \frac{0.04}{0.28}\right) = \min(1, 0.13)$$

Metropolis Hastings - Demo

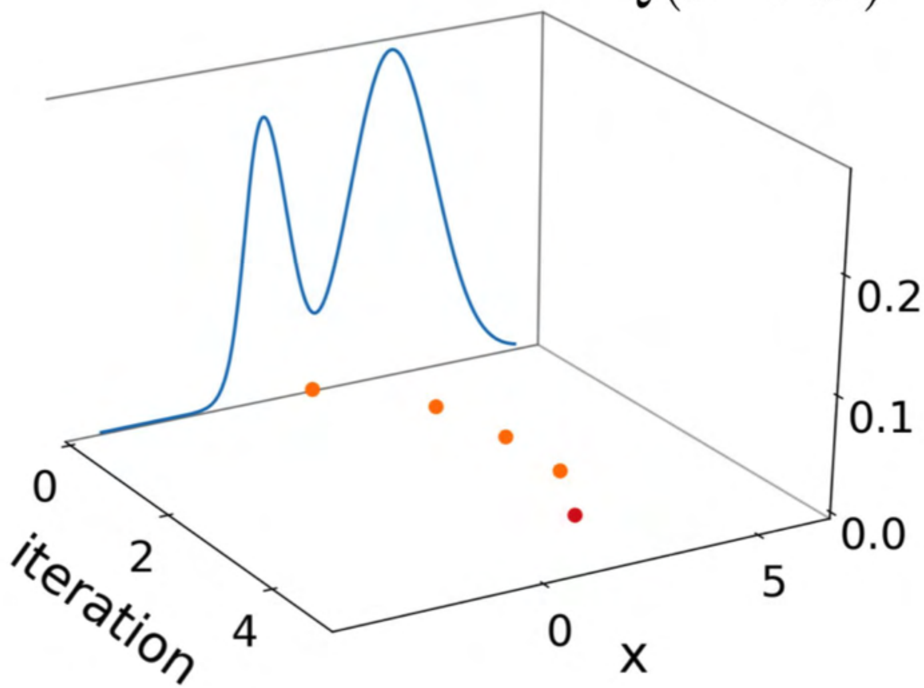
$$Q(x \to x') = \mathcal{N}(x, 1)$$



$$A(x \to x') = \min\left(1, \frac{0.04}{0.28}\right) = \min(1, 0.13)$$

Metropolis Hastings - Demo

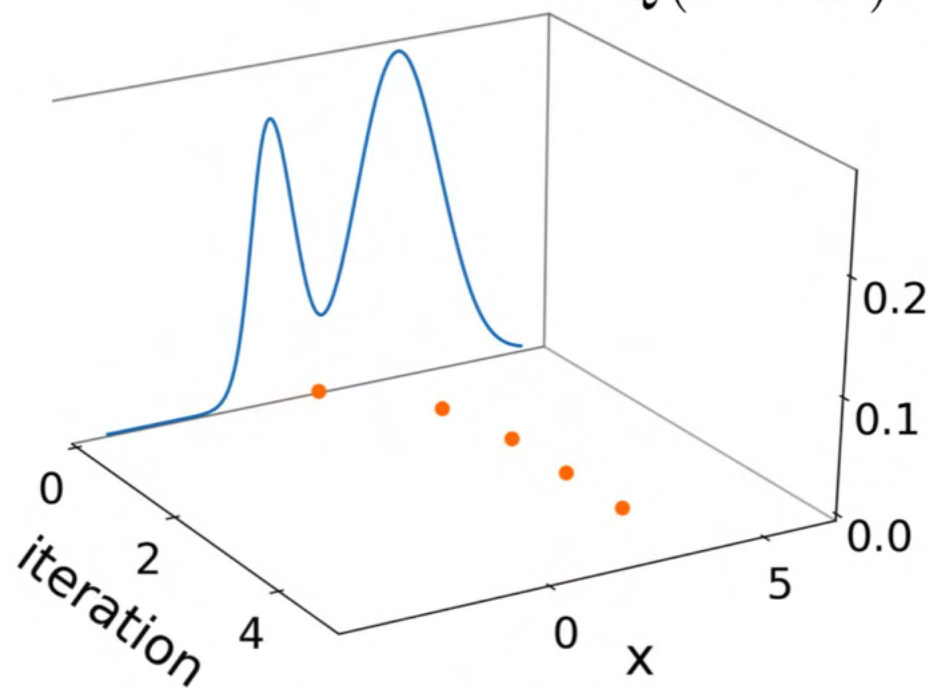
$$Q(x \to x') = \mathcal{N}(x, 1)$$



$$A(x \to x') = \min\left(1, \frac{0.20}{0.28}\right) = \min(1, 0.73)$$

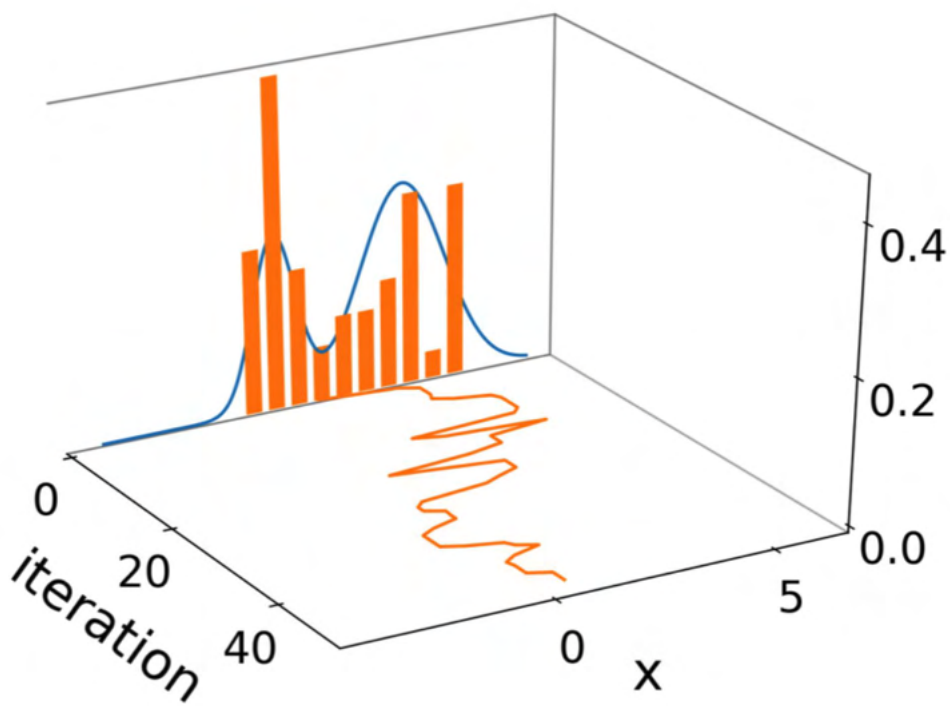
Metropolis Hastings - Demo

$$Q(x \to x') = \mathcal{N}(x, 1)$$



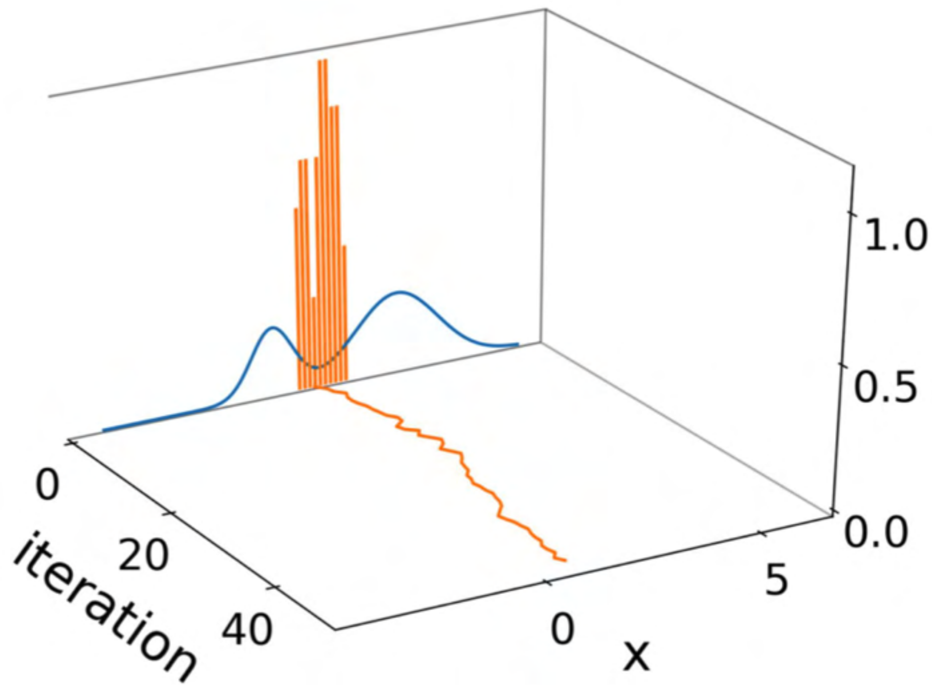
$$A(x \to x') = \min\left(1, \frac{0.20}{0.28}\right) = \min(1, 0.73)$$

Metropolis Hastings - Demo



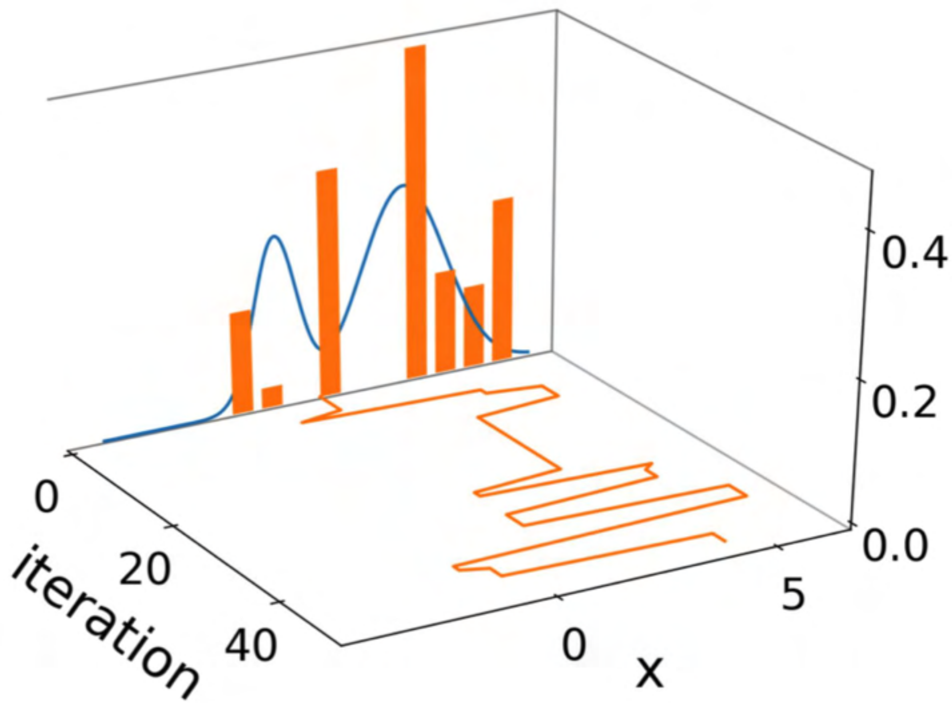
$$Q(x \to x') = \mathcal{N}(x, 1)$$

Metropolis Hastings - Demo



$$Q(x \to x') = \mathcal{N}(x, 0.1^2)$$

Metropolis Hastings - Demo



$$Q(x \to x') = \mathcal{N}(x, 10^2)$$

Metropolis Hastings as correction scheme

- Recall Gibbs sampling

$$x_1^{k+1} \sim p(x_1 \mid x_2 = x_2^k, x_3 = x_3^k)$$

$$x_2^{k+1} \sim p(x_2 \mid x_1 = x_1^{k+1}, x_3 = x_3^k)$$

$$x_3^{k+1} \sim p(x_3 \mid x_1 = x_1^{k+1}, x_2 = x_2^{k+1})$$

Metropolis Hastings as correction scheme

- Recall Gibbs sampling

- Let's make it parallel

$$x_1^{k+1} \sim p(x_1 \mid x_2 = x_2^k, x_3 = x_3^k)$$

$$x_2^{k+1} \sim p(x_2 \mid x_1 = x_1^k, x_3 = x_3^k)$$

$$x_3^{k+1} \sim p(x_3 \mid x_1 = x_1^k, x_2 = x_2^k)$$

Metropolis Hastings as correction scheme

- Recall Gibbs sampling
- Let's make it parallel
- It's wrong now, but can correct with Metropolis Hastings!

$$x_1^{k+1} \sim p(x_1 \mid x_2 = x_2^k, x_3 = x_3^k)$$

$$x_2^{k+1} \sim p(x_2 \mid x_1 = x_1^{k}, x_3 = x_3^k)$$

$$x_3^{k+1} \sim p(x_3 \mid x_1 = x_1^{k}, x_2 = x_2^{k})$$

Metropolis Hastings - Summary

- Rejection sampling applied to Markov Chains

Pros:

- You can choose among family of Markov Chains • Works for unnormalized densities

- Easy to implement

Cons:

- Samples are still correlated

- Have to choose among family of Markov Chains

Next - MCMC examples with PyMC3