Monte Carlo - Recap

- Monte Carlo methods we have discussed so far
  - Inverse CDF works for 1D
  - Analytic transformations for some distributions
  - Grid evaluation and sampling works in low dimensions
  - Rejection sampling
  - Importance sampling

Monte Carlo - Recap

- Monte Carlo methods we have discussed so far
  - Inverse CDF works for 1D
  - Analytic transformations for some distributions
  - Grid evaluation and sampling works in low dimensions
  - Rejection sampling
  - Importance sampling
- What to do in high dimensions?
  - Markov chain Monte Carlo (Ch 11-12 in BDA 3rd)

Markov Chain

- The probability of each event depends only on the state attained in the previous event
  - Sequence $x_1, x_2, \dots, x_T$ where distribution of x_t depends only on x_{t-1}
- Defined by transition distribution $A(x^{new}|x^{old})$, together with initial state x_1

Markov Chain

- The probability of each event depends only on the state attained in the previous event
  - Sequence $x_1, x_2, \ldots, x_T$ where distribution of x_t depends only on x_{t-1}
- Defined by transition distribution $A(x^{new}|x^{old})$, together with initial state x_1
- Example of a simple Markov chain on black board
  - Random walk
  - Repeatedly shuffling a deck of cards

Markov Chain

- The Markov property is such that given the present, the future is independent of the past. Let the state space be Ω, a Markov chain on Ω is determined by the transition probability

  $$K(x,y) = P(X_{t+1} = y | X_t = x, X_{t-1}, ..., X_0) = P(X_{t+1} = y | X_t = x).$$

  Let $p^{(t)}(x)$ be the marginal distribution of X_t. Then

  $$\begin{aligned} p^{(t+1)}(y) &= P(X_{t+1} = y) \\ &= \sum_x P(X_{t+1} = y, X_t = x) \\ &= \sum_x P(X_{t+1} = y | X_t = x) P(X_t = x) \\ &= \sum_x p^{(t)}(x) K(x,y). \end{aligned}$$

- Let K be the matrix $(K(x,y))$. Let $p^{(t)}$ be the row vector $(p^{(t)}(x))$. Then $p^{(t+1)} = p^{(t)}K$. By induction, $p^{(t)} = p^{(0)}K^t$.

Markov Chain: Two-step transition

$$
\begin{aligned}
K^{(2)}(x,y) &= P(X_{t+2}=y|X_t=x) \\
&= \sum_z P(X_{t+2}=y, X_{t+1}=z|X_t=x) \\
&= \sum_z P(X_{t+2}=y|X_{t+1}=z, X_t=x)P(X_{t+1}=z|X_t=x) \\
&= \sum_z K(x,z)K(z,y) = K^2(x,y).
\end{aligned}
$$

In general, $K^{(t)} = K^t$.

Markov Chain: Stationary Distribution

- All "nice enough" Markov chains have the property that if T is large enough, the distribution over x_T is almost independent of x_1 and converges to some distribution $\pi(x)$ as $T \to \infty$.

Markov Chain: Stationary Distribution

- All ”nice enough” Markov chains have the property that if T is large enough, the distribution over x_T is almost independent of x_1 and converges to some distribution $\pi(x)$ as $T \to \infty$.
- $\pi(x)$ is called the stationary distribution. The technical condition for ”nice enough” is that the Markov chain is ergodic.
  $p^{(t)} \to \pi$, is the stationary distribution, so that $\pi = \pi K$, i.e., $\pi(y) = \sum_x \pi(x) K(x, y)$.

Markov Chain: Stationary Distribution

- All ”nice enough” Markov chains have the property that if T is large enough, the distribution over x_T is almost independent of x_1 and converges to some distribution $\pi(x)$ as $T \to \infty$.
- $\pi(x)$ is called the stationary distribution. The technical condition for ”nice enough” is that the Markov chain is ergodic.
  $p^{(t)} \to \pi$, is the stationary distribution, so that $\pi = \pi K$, i.e., $\pi(y) = \sum_x \pi(x) K(x, y)$.
- The distribution $\pi(x)$ is also what we get if we count how many times x_t visits each state, as $T \to \infty$.

Markov Chain: Stationary Distribution

- All "nice enough" Markov chains have the property that if T is large enough, the distribution over x_T is almost independent of x_1 and converges to some distribution $\pi(x)$ as $T \to \infty$.
- $\pi(x)$ is called the stationary distribution. The technical condition for "nice enough" is that the Markov chain is ergodic.
  $p^{(t)} \to \pi$, is the stationary distribution, so that $\pi = \pi K$, i.e., $\pi(y) = \sum_x \pi(x) K(x, y)$.
- The distribution $\pi(x)$ is also what we get if we count how many times x_t visits each state, as $T \to \infty$.
- The mixing time is how long it takes for x_T to be close to the stationary distribution (we won't define this formally).
  - Mixing time is how many shuffles we need for deck to be "almost random"

Markov Chain: Reversibility

- A special case is the reversible Markov chain where

$$\pi(x)K(x,y) = \pi(y)K(y,x)$$

It is also called the detailed balance condition.

- If a chain is reversible with respect to π, then π is the stationary distribution, because

$$\sum_x \pi(x)K(x,y) = \sum_x \pi(y)K(y,x) = \pi(y)\sum_x K(y,x) = \pi(y).$$

Markov Chain: Transition Matrix

The transition matrix K contains the transition probabilities. We can interpret it as follows:

(1) Forward meaning as mixing. In $p^{(t+1)} = p^{(t)}K$, K acts on a row vector, and its verb meaning is mixing $p^{(t)}$ into $p^{(t+1)}$.

(2) Backward meaning as smoothing. We can also let K act on a column vector so that $g = Kh$. Both g and h can be considered functions defined on the state space Ω.

$$
\begin{aligned}
g(x) &= \sum_y K(x,y)h(y) \\
&= \sum_y h(X_{t+1}=y)P(X_{t+1}=y|X_t=x) \\
&= \mathrm{E}(h(X_{t+1})|X_t=x),
\end{aligned}
$$

which is local average of h around x. So K is smoothing h into g.

Gibbs Sampling: Motivation

- A target distribution $p(\theta_1, \ldots, \theta_d)$ that we want to sample from

Gibbs Sampling: Motivation

- A target distribution $p(\theta_1, \ldots, \theta_d)$ that we want to sample from
- Current tool: rejection sampling
  - Proposal distribution $q(\theta_1, \ldots, \theta_d)$ for all θ_i at once
  - Issue: too slow (typically exponentially small acceptance rate in d)

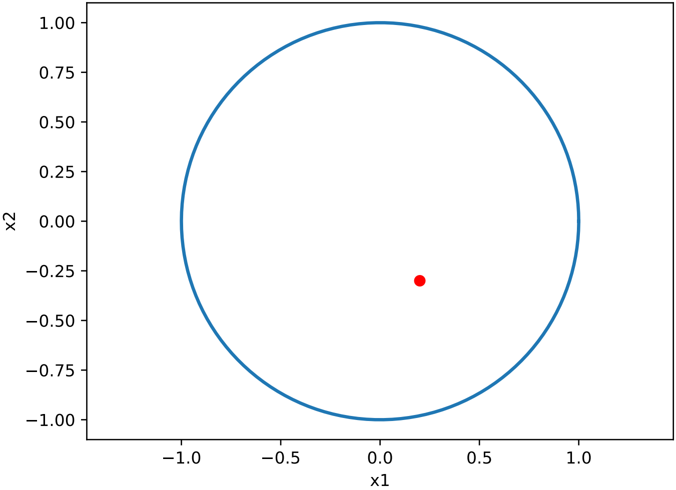
Gibbs Sampling: Motivation

- A target distribution $p(\theta_1, \ldots, \theta_d)$ that we want to sample from
- Current tool: rejection sampling
  - Proposal distribution $q(\theta_1, \ldots, \theta_d)$ for all θ_i at once
  - Issue: too slow (typically exponentially small acceptance rate in d)
- Idea behind Gibbs sampling: change one variable at a time (Markov chain)

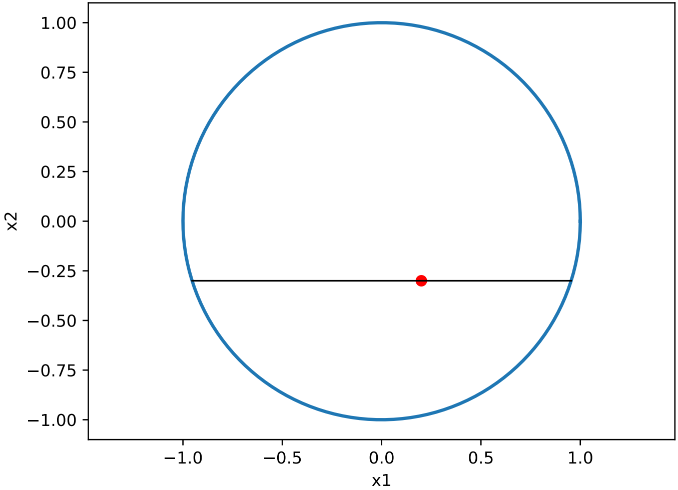
Gibbs Sampling: Motivation

- A target distribution $p(\theta_1, \ldots, \theta_d)$ that we want to sample from
- Current tool: rejection sampling
  - Proposal distribution $q(\theta_1, \ldots, \theta_d)$ for all θ_i at once
  - Issue: too slow (typically exponentially small acceptance rate in d)
- Idea behind Gibbs sampling: change one variable at a time (Markov chain)
- Algorithm
  - Initialize $(\theta_1, \ldots, \theta_n)$ arbitrarily
  - Repeat:
    Pick j (randomly or sequentially)
    Re-sample θ_j from $p(\theta_j | \theta_1^t, \ldots, \theta_{j-1}^t, \theta_{j+1}^{t-1}, \ldots, \theta_d^{t-1})$ (often denote as $p(\theta_j | \theta_{-j})$)

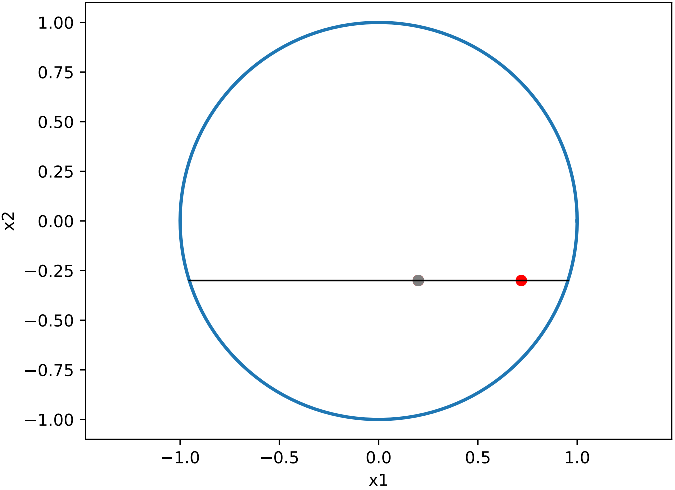
Gibbs Sampling: Unit Circle Example



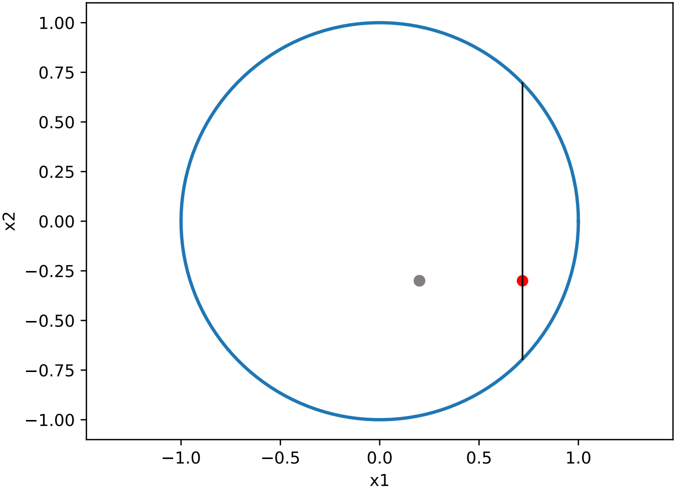
Gibbs Sampling: Unit Circle Example



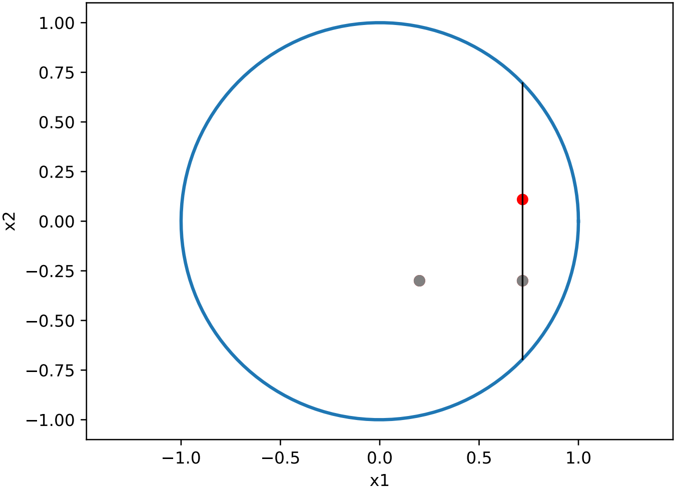
Gibbs Sampling: Unit Circle Example



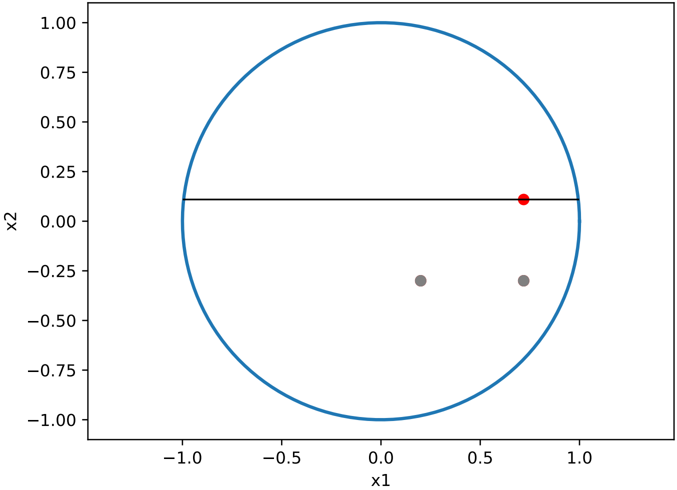
Gibbs Sampling: Unit Circle Example



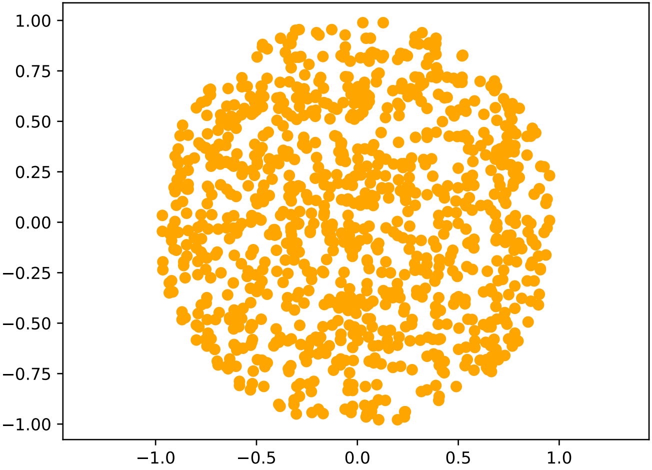
Gibbs Sampling: Unit Circle Example



Gibbs Sampling: Unit Circle Example



Gibbs Sampling: Unit Circle Example



Gibbs Sampling

- With *conditionally* conjugate priors, the sampling from the conditional distributions is easy for wide range of models

Gibbs Sampling

- With *conditionally* conjugate priors, the sampling from the conditional distributions is easy for wide range of models
- No parameters to tune

Gibbs Sampling

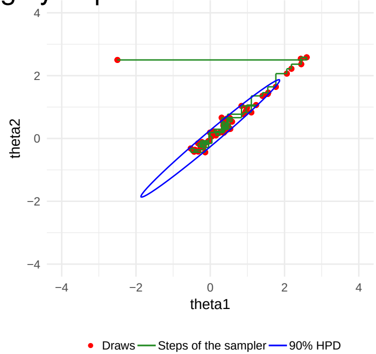
- With *conditionally* conjugate priors, the sampling from the conditional distributions is easy for wide range of models
- No parameters to tune
- If conditional distribution is not straight forward, use e.g. inverse-CDF

Gibbs Sampling

- With *conditionally* conjugate priors, the sampling from the conditional distributions is easy for wide range of models
- No parameters to tune
- If conditional distribution is not straight forward, use e.g. inverse-CDF
- Several parameters can be updated in blocks (*blocking*)

Gibbs Sampling

- With *conditionally* conjugate priors, the sampling from the conditional distributions is easy for wide range of models
- No parameters to tune
- If conditional distribution is not straight forward, use e.g. inverse-CDF
- Several parameters can be updated in blocks (*blocking*)
- Slow if parameters are highly dependent



•

Metropolis algorithm

- Algorithm
  1. starting point θ^0
  2. $t = 1, 2, \ldots$
     (a) pick a proposal θ^* from the proposal distribution $J_t(\theta^*|\theta^{t-1})$.
     Proposal distribution has to be symmetric, i.e.
     $J_t(\theta_a|\theta_b) = J_t(\theta_b|\theta_a)$, for all θ_a, θ_b

Metropolis algorithm

- Algorithm
  1. starting point θ^0
  2. $t = 1, 2, \ldots$
     (a) pick a proposal θ^* from the proposal distribution $J_t(\theta^*|\theta^{t-1})$. Proposal distribution has to be symmetric, i.e. $J_t(\theta_a|\theta_b) = J_t(\theta_b|\theta_a)$, for all θ_a, θ_b
     (b) calculate acceptance ratio

$$r = \frac{p(\theta^*|y)}{p(\theta^{t-1}|y)}$$

Metropolis algorithm

- Algorithm
  1. starting point θ^0
  2. $t = 1, 2, \ldots$
     (a) pick a proposal θ^* from the proposal distribution $J_t(\theta^*|\theta^{t-1})$. Proposal distribution has to be symmetric, i.e. $J_t(\theta_a|\theta_b) = J_t(\theta_b|\theta_a)$, for all θ_a, θ_b
     (b) calculate acceptance ratio
     $$r = \frac{p(\theta^*|y)}{p(\theta^{t-1}|y)}$$
     (c) set
     $$\theta^t = \begin{cases} \theta^* & \text{with probability } \min(r, 1) \\ \theta^{t-1} & \text{otherwise} \end{cases}$$

Metropolis algorithm

- Algorithm
  1. starting point θ^0
  2. $t = 1, 2, \ldots$
     (a) pick a proposal θ^* from the proposal distribution $J_t(\theta^*|\theta^{t-1})$. Proposal distribution has to be symmetric, i.e. $J_t(\theta_a|\theta_b) = J_t(\theta_b|\theta_a)$, for all θ_a, θ_b
     (b) calculate acceptance ratio
     $$r = \frac{p(\theta^*|y)}{p(\theta^{t-1}|y)}$$
     (c) set
     $$\theta^t = \begin{cases} \theta^* & \text{with probability } \min(r, 1) \\ \theta^{t-1} & \text{otherwise} \end{cases}$$
     ie, if $p(\theta^*|y) > p(\theta^{t-1}|y)$, always accept the proposal and otherwise accept the proposal with probability r

Metropolis algorithm

- Algorithm
  1. starting point θ^0
  2. $t = 1, 2, \ldots$
     (a) pick a proposal θ^* from the proposal distribution $J_t(\theta^*|\theta^{t-1})$. Proposal distribution has to be symmetric, i.e. $J_t(\theta_a|\theta_b) = J_t(\theta_b|\theta_a)$, for all θ_a, θ_b
     (b) calculate acceptance ratio
     $$r = \frac{p(\theta^*|y)}{p(\theta^{t-1}|y)}$$
     (c) set
     $$\theta^t = \begin{cases} \theta^* & \text{with probability } \min(r, 1) \\ \theta^{t-1} & \text{otherwise} \end{cases}$$

- step c is executed by generating a random number from U(0, 1)

Metropolis algorithm

- Algorithm
  1. starting point θ^0
  2. $t = 1, 2, \ldots$
     (a) pick a proposal θ^* from the proposal distribution $J_t(\theta^*|\theta^{t-1})$. Proposal distribution has to be symmetric, i.e. $J_t(\theta_a|\theta_b) = J_t(\theta_b|\theta_a)$, for all θ_a, θ_b
     (b) calculate acceptance ratio
     $$r = \frac{p(\theta^*|y)}{p(\theta^{t-1}|y)}$$
     (c) set
     $$\theta^t = \begin{cases} \theta^* & \text{with probability } \min(r, 1) \\ \theta^{t-1} & \text{otherwise} \end{cases}$$

  - step c is executed by generating a random number from U(0, 1)
  - $p(\theta^*|y)$ and $p(\theta^{t-1}|y)$ have the same normalization terms, and thus instead of $p(\cdot|y)$, unnormalized $q(\cdot|y)$ can be used, as the normalization terms cancel out!

Why Metropolis algorithm works

- Intuitively more draws from the higher density areas as jumps to higher density are always accepted and only some of the jumps to the lower density are accepted

Why Metropolis algorithm works

- Intuitively more draws from the higher density areas as jumps to higher density are always accepted and only some of the jumps to the lower density are accepted

- Theoretically
  1. Prove that simulated series is a Markov chain which has unique stationary distribution
  2. Prove that this stationary distribution is the desired target distribution

Metropolis-Hastings algorithm

- Generalization of Metropolis algorithm for non-symmetric proposal distributions
  - acceptance ratio includes ratio of proposal distributions

$$r = \frac{p(\theta^*|y)/J_t(\theta^*|\theta^{t-1})}{p(\theta^{t-1}|y)/J_t(\theta^{t-1}|\theta^*)}$$

Metropolis-Hastings algorithm

- Generalization of Metropolis algorithm for non-symmetric proposal distributions
  - acceptance ratio includes ratio of proposal distributions

$$r = \frac{p(\theta^*|y)/J_t(\theta^*|\theta^{t-1})}{p(\theta^{t-1}|y)/J_t(\theta^{t-1}|\theta^*)} = \frac{p(\theta^*|y)J_t(\theta^{t-1}|\theta^*)}{p(\theta^{t-1}|y)J_t(\theta^*|\theta^{t-1})}$$

Metropolis-Hastings algorithm

- Ideal proposal distribution is the distribution itself
  - $J(\theta^*|\theta) \equiv p(\theta^*|y)$ for all θ
  - acceptance probability is 1
  - independent draws
  - not usually feasible

Metropolis-Hastings algorithm

- Ideal proposal distribution is the distribution itself
  - $J(\theta^*|\theta) \equiv p(\theta^*|y)$ for all θ
  - acceptance probability is 1
  - independent draws
  - not usually feasible
- Good proposal distribution resembles the target distribution
  - if the shape of the target distribution is unknown, usually normal or t distribution is used

Metropolis-Hastings algorithm

- Ideal proposal distribution is the distribution itself
  - $J(\theta^*|\theta) \equiv p(\theta^*|y)$ for all θ
  - acceptance probability is 1
  - independent draws
  - not usually feasible
- Good proposal distribution resembles the target distribution
  - if the shape of the target distribution is unknown, usually normal or *t* distribution is used
- After the shape has been selected, it is important to select the scale
  - small scale
    $\rightarrow$ many steps accepted, but the chain moves slowly due to small steps
  - big scale
    $\rightarrow$ long steps proposed, but many of those rejected and again chain moves slowly

Metropolis-Hastings algorithm

- Ideal proposal distribution is the distribution itself
  - $J(\theta^*|\theta) \equiv p(\theta^*|y)$ for all θ
  - acceptance probability is 1
  - independent draws
  - not usually feasible
- Good proposal distribution resembles the target distribution
  - if the shape of the target distribution is unknown, usually normal or *t* distribution is used
- After the shape has been selected, it is important to select the scale
  - small scale
    $\rightarrow$ many steps accepted, but the chain moves slowly due to small steps
  - big scale
    $\rightarrow$ long steps proposed, but many of those rejected and again chain moves slowly
- Generic rule for rejection rate is 60-90% (but depends on dimensionality and a specific algorithm variation)

Gibbs sampling

- Specific case of Metropolis-Hastings algorithm
  - single updated (or blocked)
  - proposal distribution is the conditional distribution
    $\rightarrow$ proposal and target distributions are same
    $\rightarrow$ acceptance probability is 1