Markov chain Monte Carlo methods

Goal

Sample from a distribution:

$$p(x) = \frac{\gamma(x)}{Z},$$

where $Z = \int \gamma(x)dx$.

Goal

Sample from a distribution:

$$p(x) = \frac{\gamma(x)}{Z},$$

where $Z = \int \gamma(x)dx$.

We want to approximate complex integral: let $x_1, ..., x_N \sim p(x)$, then for any test function $h(x)$:

$$\mathbb{E}_{X\sim p}[h(X)] = \int h(x)p(x)dx \approx \frac{1}{N}\sum_{n=1}^{N} h(x_n).$$

Bayesian statistics

Sample from a distribution:

$$p(x) = \frac{\gamma(x)}{Z}$$

Bayesian statistics

Sample from a distribution:

$$p(x) = \frac{\gamma(x)}{Z}$$

- $p(x) = p(x|y)$: posterior

Bayesian statistics

Sample from a distribution:

$$p(x) = \frac{\gamma(x)}{Z}$$

- $p(x) = p(x|y)$: posterior
- $\gamma(x) = p(x, y)$: joint likelihood

Bayesian statistics

Sample from a distribution:

$$p(x) = \frac{\gamma(x)}{Z}$$

- $p(x) = p(x|y)$: posterior
- $\gamma(x) = p(x, y)$: joint likelihood
- $Z = p(y) = \int p(x, y)dx$: marginal likelihood, which is usually intractable.

Bayesian statistics

Sample from a distribution:

$$p(x) = \frac{\gamma(x)}{Z}$$

- $p(x) = p(x|y)$: posterior
- $\gamma(x) = p(x, y)$: joint likelihood
- $Z = p(y) = \int p(x, y)dx$: marginal likelihood, which is usually intractable.

Here, x can be parameters or any latent variables of interest.

Importance sampling

Instead of sampling from p (hard to do because Z is unknown), sample $x \sim q$ and adjust for the difference between γ and q:

$$\int h(x)p(x)dx \approx \sum \bar{w}(x)h(x),$$

where $w(x) = \gamma(x)/q(x)$ and $\bar{w}(x) = w(x)/\sum_n w(x)$.

Sequential Monte Carlo methods

If $x = (x_1, ..., x_D)$ is high-dimensional, we can sample each component sequentially:

Sequential Monte Carlo methods

If $x = (x_1, ..., x_D)$ is high-dimensional, we can sample each component sequentially:

- $x_d^n \sim q_d(x_d | x_{1:d-1})$.

Sequential Monte Carlo methods

If $x = (x_1, ..., x_D)$ is high-dimensional, we can sample each component sequentially:

- $x_d^n \sim q_d(x_d | x_{1:d-1})$.
- Interleave resampling step to maintain particle diversity and prune unpromising particles.

Sequential Monte Carlo methods

SMC methods work well when there is a temporal structure in x, where it is natural to sample one dimension at a time.

Sequential Monte Carlo methods

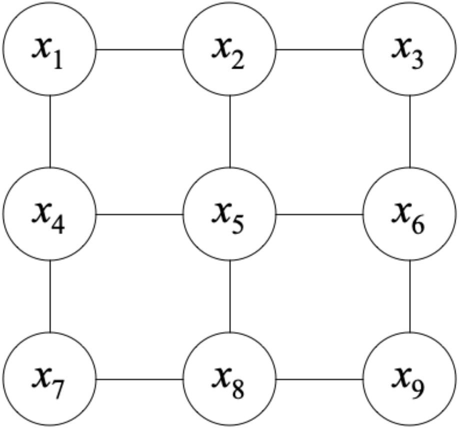
SMC methods work well when there is a temporal structure in x, where it is natural to sample one dimension at a time.

So why do we need another algorithm/method?

- Large variance associated with choice of proposal distribution.
- Curse of dimensionality may still manifest and approximation can be poor.

Example: Lattice

Consider a $K \times K$ 2-dimensional lattice $G = (V, E)$.



Example: Lattice

- Ising model: Each node represents a RV X_v which takes a value in $\{-1, +1\}$, denoting the "spin" of an atom/molecule.

Example: Lattice

- Ising model: Each node represents a RV X_v which takes a value in $\{-1, +1\}$, denoting the "spin" of an atom/molecule.
- Spatial analysis: Each node represents a spatial coordinate (spatial statistics) and $X_v \in \{\text{no gold}, \text{gold!}\}$ or $X_v \in \mathbb{R}^+$ some measurement of the amount of gold at location v.

Example: Lattice

- Ising model: Each node represents a RV X_v which takes a value in $\{-1, +1\}$, denoting the "spin" of an atom/molecule.
- Spatial analysis: Each node represents a spatial coordinate (spatial statistics) and $X_v \in \{\text{no gold}, \text{gold!}\}$ or $X_v \in \mathbb{R}^+$ some measurement of the amount of gold at location v.
- Image processing: Each node represents a pixel of an image. $X_v \in \{\text{black}, \text{white}\}$ or gray scale $X_v \in [0, 1]$ or RGB color.

Example: Ising model

Let $X = (X_v)$. The "energy" function for the Ising model is defined as:

$$H(x) = \sum_{v} \phi(x_v) + \sum_{(u,v)\in E} \psi(x_u, x_v),$$

- $(u, v) \in E$ denote neighbors (adjacent nodes),
- ϕ: unary potential,
- ψ: pairwise potential (measuring interaction strength).

Example: $\phi(x_v) = \beta x_v$ and $\psi(x_u, x_v) = \kappa x_u x_v$ for $\beta, \kappa \in \mathbb{R}$.

Example: Ising model

The probability distribution on X is defined as,

$$p(x) = \frac{1}{Z} \exp(-H(x))$$

where

$$Z = \sum_{x_v : v \in V} \exp(-H(x)).$$

Z in statistical physics is referred to as “partition function”. Essentially a normalization constant.

SMC for Ising model?

- Not obvious what order to sample the variables.
- It could lead to very few unique values for x_v sampled earlier in the SMC iteration.
- Leads to poor approximation involving those sampled earlier on.

SMC for Ising model?

- Not obvious what order to sample the variables.
- It could lead to very few unique values for x_v sampled earlier in the SMC iteration.
- Leads to poor approximation involving those sampled earlier on.

- For the Ising model, maybe it makese more sense to continually sample new values for x_v given x_{-v} until we are satisfied.

Metropolis-Hastings algorithm

Initialize x_0.

Metropolis-Hastings algorithm

Initialize x_0.

For $t = 1, ..., T$:

Metropolis-Hastings algorithm

Initialize x_0.

For $t = 1, ..., T$:

- Propose a value $x' \sim q(\cdot|x_{t-1})$.

Metropolis-Hastings algorithm

Initialize x_0.

For $t = 1, ..., T$:

- Propose a value $x' \sim q(\cdot|x_{t-1})$.
- Compute acceptance probability:

$$A(x'|x) = \min\left(1, \frac{\gamma(x')}{\gamma(x_{t-1})}\frac{q(x_{t-1}|x')}{q(x'|x_{t-1})}\right).$$

Metropolis-Hastings algorithm

Initialize x_0.

For $t = 1, ..., T$:

- Propose a value $x' \sim q(\cdot|x_{t-1})$.
- Compute acceptance probability:

$$A(x'|x) = \min\left(1, \frac{\gamma(x')}{\gamma(x_{t-1})}\frac{q(x_{t-1}|x')}{q(x'|x_{t-1})}\right).$$

- Sample $u \sim U(0,1)$

Metropolis-Hastings algorithm

Initialize x_0.

For $t = 1, ..., T$:

- Propose a value $x' \sim q(\cdot|x_{t-1})$.
- Compute acceptance probability:

$$A(x'|x) = \min\left(1, \frac{\gamma(x')}{\gamma(x_{t-1})}\frac{q(x_{t-1}|x')}{q(x'|x_{t-1})}\right).$$

- Sample $u \sim U(0, 1)$
- Set,

$$x_t = \begin{cases} x' & \text{if } u < A \\ x_{t-1} & \text{otherwise} \end{cases}$$

Why does MH work?

- If we take samples $x_1, ..., x_N$ using MH algorithm, why is this equivalent to taking samples from the target distribution $p(x)$?

Markov chain

Markov chain $\{X_t\}$ is a stochastic process modeling a sequence of events where the probability of each event depends only on the previous event.

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

Markov chain

Given measurable space $(\mathcal{X}, \mathcal{F})$,

$$K : \mathcal{X} \times \mathcal{F} \to [0, 1]$$

is referred to as the Markov kernel (a probability measure).

- Each random variable $X_t \in \mathcal{X}$
- $K(x, A)$ specifies the probability of moving to a set $F \in \mathcal{F}$ given that the chain is in state $x \in \mathcal{X}$.

Markov chain: continuous state space

For continuous state space, $\mathcal{X} = \mathbb{R}$, the transition probability can be described using a density function $K(x_{t-1}, x_t) = k(x_t | x_{t-1})$.

Markov chain: discrete state space

For discrete state space, the Markov chain is described using a transition matrix P, where P_{ij}represents the probability of transitioning from state $P(x_t = j|x_{t-1} = i)$.

Markov chain: stationary distribution

The Markov chain $\{X_t\}$ converges to unique **stationary** distribution as $t \to \infty$ if some conditions are satisfied.

A probability distribution π defined on $\mathcal{X}$ is invariant (stationary) under a Markov kernel K if for all $F \in \mathcal{F}$

$$\pi(A) = \int \pi(x)K(x, F)dx.$$

For discrete case: $\pi = \pi P$.

Markov chain: detailed balance (reversibility)

A Markov chain with kernel $K : \mathcal{X} \times \mathcal{F}$ satisfies the detailed balance condition with respect to a probability distribution π if,

$$\pi(x)k(x'|x) = \pi(x')k(x|x').$$

Reversibility: probability of being in state x and moving to x' from x is the same as being in state x' and moving to x.

- Note: detailed balance is a stronger condition than stationary condition: if detailed balance is satisfied, π is a stationary distribution of the Markov chain with kernel K.

Markov chain: Ergodicity

1. Aperiodic: Markov chain does not return to the same state at some fixed interval.
2. Positive recurrent: the expected number of steps for returning to the same state is finite.

Metropolis-Hastings is a Markov chain

Claim: MH algorithm constructs a Markov chain on $\mathcal{X}$ whose stationary distribution is $p(x)$.

Metropolis-Hastings is a Markov chain

Claim: MH algorithm constructs a Markov chain on $\mathcal{X}$ whose stationary distribution is $p(x)$.

Markov transition kernel:

- given current state x, we move to a new state x' with probability

$$q(x'|x)A(x'|x)$$

Metropolis-Hastings is a Markov chain

Claim: MH algorithm constructs a Markov chain on $\mathcal{X}$ whose stationary distribution is $p(x)$.

Markov transition kernel:

- given current state x, we move to a new state x' with probability

$$q(x'|x)A(x'|x)$$

- stay at current state x with probability

$$q(x|x) + q(x'|x)(1 - A(x)).$$

MH satisfies detailed balance

To prove: $p(x)k(x'|x) = p(x')k(x|x')$.

MH satisfies detailed balance

To prove: $p(x)k(x'|x) = p(x')k(x|x')$.

To move from state x to x', we must first propose x' and accept x'.

MH satisfies detailed balance

To prove: $p(x)k(x'|x) = p(x')k(x|x')$.

To move from state x to x', we must first propose x' and accept x'.

Case 1: $A(x'|x) = \frac{p(x')q(x|x')}{p(x)q(x'|x)} < 1$.

$$
\begin{aligned}
p(x)q(x'|x)A(x'|x) &= p(x)q(x'|x)\frac{p(x')q(x|x')}{p(x)q(x'|x)} && (1) \\
&= p(x')q(x|x'). && (2)
\end{aligned}
$$

MH satisfies detailed balance

Case 2: $A(x'|x) \geq 1$.

$$p(x')q(x|x')A(x|x') = p(x')q(x|x')\frac{p(x)q(x'|x)}{p(x')q(x|x')} \quad (3)$$

$$= p(x)q(x'|x). \quad (4)$$

Is MH an ergodic Markov chain?

Yes, as long as we choose our proposal carefully.

- Randomness in k is needed to prevent apriodicity. This is built-in the MH kernel where we reject proposed values with some chance.

Is MH an ergodic Markov chain?

Yes, as long as we choose our proposal carefully.

- Randomness in k is needed to prevent apriodicity. This is built-in the MH kernel where we reject proposed values with some chance.
- Recurrent: ensure that we choose a proposal that allows to visit every state $x \in \mathcal{X}$. For example, Gaussian random walk $q(x'|x) = N(x'|x, \sigma^2 I)$ satisfies this.

Is MH an ergodic Markov chain?

Yes, as long as we choose our proposal carefully.

- Randomness in k is needed to prevent apriodicity. This is built-in the MH kernel where we reject proposed values with some chance.
- Recurrent: ensure that we choose a proposal that allows to visit every state $x \in \mathcal{X}$. For example, Gaussian random walk $q(x'|x) = N(x'|x, \sigma^2 I)$ satisfies this.
- For discrete case, ensure $q(x'|x) > 0$ for all $x', x \in \mathcal{X}$.

Metropolis-Hastings algorithm

- Choosing a suitable Metropolis-Hastings proposal distribution $q(x'|x)$ is crucial.

Metropolis-Hastings algorithm

- Choosing a suitable Metropolis-Hastings proposal distribution $q(x'|x)$ is crucial.
- A **local proposal** (e.g., small Guassian perturbation) allows gradual exploration and prevents the chain from getting stuck.

Metropolis-Hastings algorithm

- Choosing a suitable Metropolis-Hastings proposal distribution $q(x'|x)$ is crucial.
- A **local proposal** (e.g., small Guassian perturbation) allows gradual exploration and prevents the chain from getting stuck.
- Independent Metropolis refers to the case where a global proposal is used $q(x'|x) = q(x')$, which can lead to high rejection rates (why?).

Metropolis-Hastings algorithm

- Choosing a suitable Metropolis-Hastings proposal distribution $q(x'|x)$ is crucial.
- A **local proposal** (e.g., small Guassian perturbation) allows gradual exploration and prevents the chain from getting stuck.
- Independent Metropolis refers to the case where a global proposal is used $q(x'|x) = q(x')$, which can lead to high rejection rates (why?).
- Large global proposals tend to be rejected, causing the chain to get stuck at a point for long periods.

Gibbs sampling

Gibbs sampling is an MCMC algorithm, which is well suited for high-dimensional distributions where sampling directly from the joint distribution is difficult.

Gibbs sampling

1. Initialize x^0.

Gibbs sampling

1. Initialize x^0.
2. For $t = 1, ..., T$:

- Iterate over each variable x_i:
  - Sample $x_i^t \sim p(x_i | x_{-i}^t)$, where x_{-i} refers to all other variables except x_i.

Gibbs sampling

1. Initialize x^0.
2. For $t = 1, ..., T$:
   - Iterate over each variable x_i:
     - Sample $x_i^t \sim p(x_i | x_{-i}^t)$, where x_{-i} refers to all other variables except x_i.

This means that each variable is sampled from its conditional distribution given the current values of all other variables.

Gibbs sampling

1. Initialize x^0.
2. For $t = 1, ..., T$:
   - Iterate over each variable x_i:
     - Sample $x_i^t \sim p(x_i | x_{-i}^t)$, where x_{-i} refers to all other variables except x_i.

This means that each variable is sampled from its conditional distribution given the current values of all other variables.

Gibbs sampling is particularly effective when the conditional distributions $p(x_i | x_{-i})$ are easy to sample from.

Why doe Gibbs sampling work?

Claim: Gibbs sampling can be viewed as a special case of the Metropolis-Hastings algorithm where the proposal distribution is always accepted.

Why doe Gibbs sampling work?

Claim: Gibbs sampling can be viewed as a special case of the Metropolis-Hastings algorithm where the proposal distribution is always accepted.

Proof: Suppose we are proposing a new value for x_i. Let $x' = (x_1, ..., x_{i-1}, x_i', x_{i+1}, ..., x_N)$.

Why doe Gibbs sampling work?

Claim: Gibbs sampling can be viewed as a special case of the Metropolis-Hastings algorithm where the proposal distribution is always accepted.

Proof: Suppose we are proposing a new value for x_i. Let $x' = (x_1, ..., x_{i-1}, x'_i, x_{i+1}, ..., x_N)$.

$$\begin{aligned} A(x'|x) &= \frac{p(x')q(x|x')}{p(x)q(x'|x)} \\ &= \frac{p(x'_i|x_{-i})p(x_{-i})q(x|x')}{p(x_i|x_{-i})p(x_{-i})q(x'|x)}. \end{aligned}$$

Why doe Gibbs sampling work?

Claim: Gibbs sampling can be viewed as a special case of the Metropolis-Hastings algorithm where the proposal distribution is always accepted.

Proof: Suppose we are proposing a new value for x_i. Let $x' = (x_1, ..., x_{i-1}, x'_i, x_{i+1}, ..., x_N)$.

$$\begin{aligned} A(x'|x) &= \frac{p(x')q(x|x')}{p(x)q(x'|x)} \\ &= \frac{p(x'_i|x_{-i})p(x_{-i})q(x|x')}{p(x_i|x_{-i})p(x_{-i})q(x'|x)}. \end{aligned}$$

Since $q(x'|x) = p(x'_i|x_{-i})$ and $q(x|x') = p(x_i|x_{-i})$, the acceptance probability simplifies to 1.

Gibbs for Ising model

For $t = 1, ..., T$:

- Sample $x_v \sim p(x_v | x_{-v})$ for each $v \in V$.

Sample each variable in turn, conditioned on the values of all of the other variables.

Gibbs for Ising model

For $t = 1, ..., T$:

- Sample $x_v \sim p(x_v | x_{-v})$ for each $v \in V$.

Sample each variable in turn, conditioned on the values of all of the other variables.

What does $p(x_v | x_{-v})$ look like?

Gibbs sampling for Ising model

$$p(x_v|x_{-v}) = \frac{p(x_v, x_{-v})}{\sum_{x'_v} p(x'_v, x_{-v})} \tag{5}$$

$$\propto \exp(-\phi(x_v) - \sum_{(u,v)\in E} \psi(x_u, x_v)). \tag{6}$$

Example: image denoising

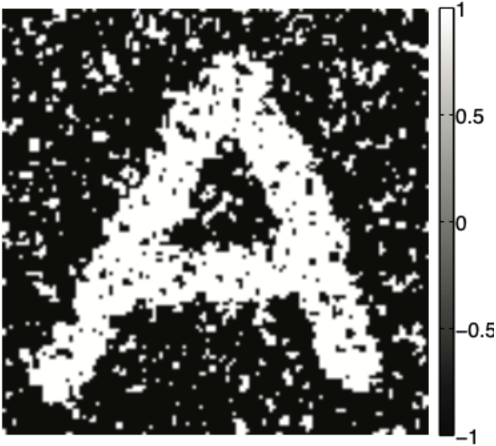


Figure 1: Fig 12.3 (a), PML 2

If all of the neighbors of x_v is white/black, x_v is likely to be white/black.

Example: image denoising

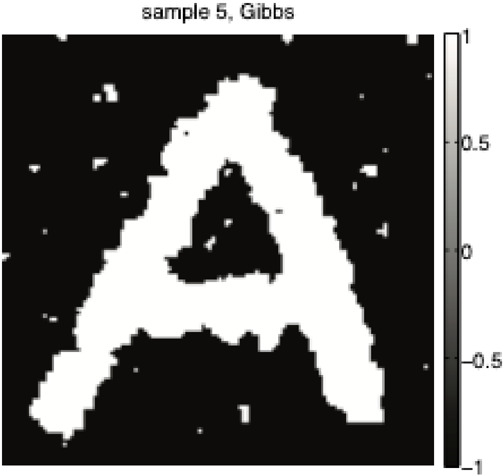


Figure 2: Fig 12.3 (b), PML 2

Example: image denoising

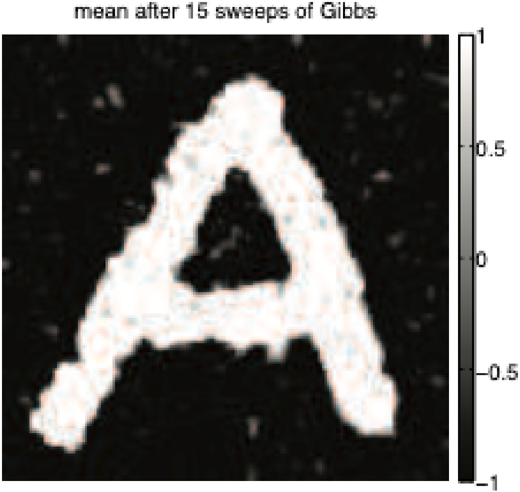


Figure 3: Fig 12.3 (c), PML 2

Undirected graphical models

Undirected graphical models (UGM):

- Each node $v \in V$ represents a random variable.

Undirected graphical models

Undirected graphical models (UGM):

- Each node $v \in V$ represents a random variable.
- Edge between $u, v \in V$ is denoted (u, v). Presence of an edge indicates that there is a symmetric relationship between u and v but we cannot easily pinpoint directionality.

Undirected graphical models

- UGMs are commonly referred to as Markov Random Field (MRF).

Undirected graphical models

- UGMs are commonly referred to as Markov Random Field (MRF).
- Commonly used for modeling dependence structure where directionality is unclear.

Undirected graphical models

- UGMs are commonly referred to as Markov Random Field (MRF).
- Commonly used for modeling dependence structure where directionality is unclear.
- Example: The value taken at each pixel (random variable X_v) is related to the value taken by its neighbors but it is not causal.

Undirected graphical models

- Pairwise Markov property: For any two non-adjacent nodes u, v, $X_u \perp X_v | X_{rest}$.

Undirected graphical models

- Pairwise Markov property: For any two non-adjacent nodes u, v, $X_u \perp X_v | X_{rest}$.
- Local Markov property: $X_u \perp X_{rest} | X_{nbr(u)}$, where $nbr(u) = \{v : (u, v) \in E\}$.

Undirected graphical models

- Pairwise Markov property: For any two non-adjacent nodes u, v, $X_u \perp X_v | X_{rest}$.
- Local Markov property: $X_u \perp X_{rest} | X_{nbr(u)}$, where $nbr(u) = \{v : (u, v) \in E\}$.
- Global Markov property:

Any two sets $A, B \subset V$, are conditionally independent given a separating set S, i.e., $X_A \perp X_B | X_S$, if S separates A and B in G.

Undirected graphical models

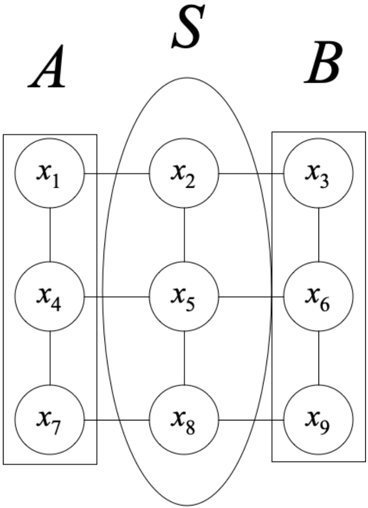


Figure 4: Global Markov property

Undirected graphical models

Markov blanket of v is defined as a minimal set of nodes that separates v from the rest of the nodes. It is given by, $MB(v) = nbr(v)$.

Undirected graphical models

Markov blanket of v is defined as a minimal set of nodes that separates v from the rest of the nodes. It is given by, $MB(v) = nbr(v)$.

MB plays a central role in determining efficient inference algorithm. Example, Gibbs sampling update of a variable X_v is conditioned on its MB and nothing else.

Undirected graphical models

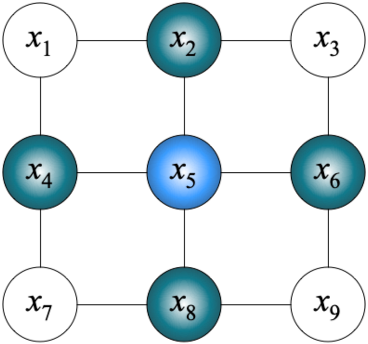


Figure 5: Markov blanket

Undirected graphical models: Hammersley-Clifford Theorem

A strictly positive probability distribution $p(x_V)$ satisfies the global Markov property with respect to G if and only if it can be factorized as,

$$p(x_V) = \frac{1}{Z} \prod_{C \in \mathcal{C}} \psi_C(x_C),$$

- $\mathcal{C}$ denotes the set of (maximal) **cliques**,
- ψ_C denotes potential function for clicque C,
- Z is normalization constant also referred to as partition function.

Undirected graphial models

Clique $C \subseteq V$ of $G = (V, E)$ is a fully connected subgraph of G such that every pair of nodes $u, v \in C$ are adjacent i.e., $\{u, v\} \in E$.

A clique C is maximal if adding a node to C does not preserve full connectivity.

Undirected graphical models

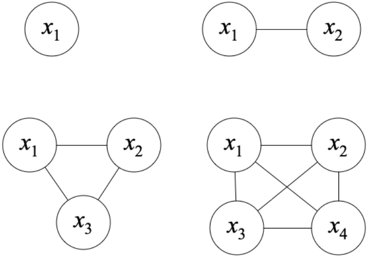


Figure 6: Example: Markov blanket

An edge $\{u, v\}$ is a clique. A fully connected set of nodes is a clique.

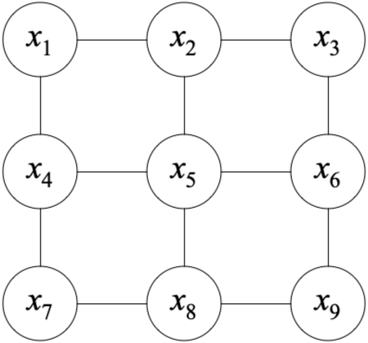
Undirected graphial models

Computing partition function is a source of great computational challenge:

$$Z = \int_{\mathcal{X}_V} \prod_{C \in \mathcal{C}} \psi_C(x_C).$$

In most cases, the inference involving UGM requires approximate methods.

Undirected graphical models



What are the maximal cliques in this graph?

Back to Gibbs sampling

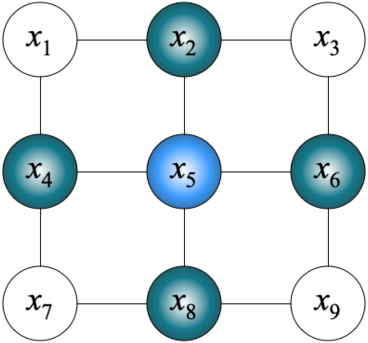
Given an UGM, determine the Markov blanket for each node v.

Back to Gibbs sampling

Given an UGM, determine the Markov blanket for each node v.

Determine the conditional $p(x_v|x_{mb(v)})$.

Back to Gibbs sampling



Blocked Gibbs sampling

- Partition the nodes into disjoint sets $A, B \subset V$ such that

$$x_u \perp x_v | B, \quad u, v \in A$$

and

$$x_u \perp x_v | A, \quad u, v \in B.$$

Blocked Gibbs sampling

- Partition the nodes into disjoint sets $A, B \subset V$ such that

$$x_u \perp x_v | B, \quad u, v \in A$$

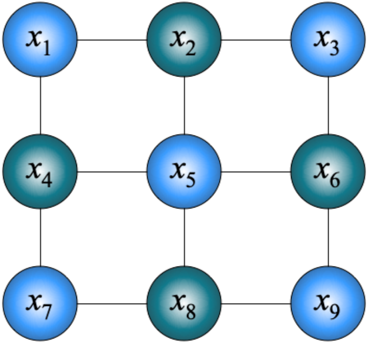
and

$$x_u \perp x_v | A, \quad u, v \in B.$$

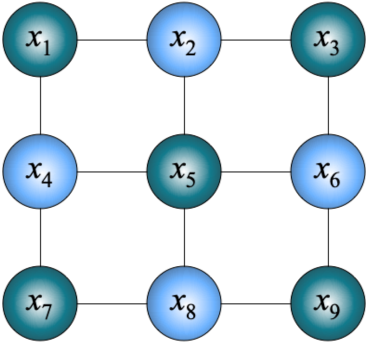
At each iteration $t = 1, ..., T$:

- Sample $p(x_A | x_{-A})$,
- Sample $p(x_B | x_{-B})$.

Blocked Gibbs sampling



Blocked Gibbs sampling



Summary

- Selecting the right inference algorithm depends on the problem's structure and computational constraints.

Summary

- Selecting the right inference algorithm depends on the problem's structure and computational constraints.
- MCMC methods can be utilized to sample from intractable distributions.

Summary

- Selecting the right inference algorithm depends on the problem's structure and computational constraints.
- MCMC methods can be utilized to sample from intractable distributions.
- Metropolis-Hastings provides general sampling but requires careful proposal design for efficiency.

Summary

- Selecting the right inference algorithm depends on the problem's structure and computational constraints.
- MCMC methods can be utilized to sample from intractable distributions.
- Metropolis-Hastings provides general sampling but requires careful proposal design for efficiency.
- Gibbs sampling is efficient when conditional distributions are easy to sample from, leveraging local dependencies.

Summary

- Selecting the right inference algorithm depends on the problem's structure and computational constraints.
- MCMC methods can be utilized to sample from intractable distributions.
- Metropolis-Hastings provides general sampling but requires careful proposal design for efficiency.
- Gibbs sampling is efficient when conditional distributions are easy to sample from, leveraging local dependencies.
- MRFs serve as a foundation for probabilistic inference, particularly in structured probabilistic models.

Applications of UGMs

- Neuroscience and associative memory: Hopfield networks (1982, 1984).
  - Energy-based models for pattern recognition and memory retrieval.

Applications of UGMs

- Neuroscience and associative memory: Hopfield networks (1982, 1984).
  - Energy-based models for pattern recognition and memory retrieval.
- Deep learning: Restricted Boltzmann Machines (1986,2006).
  - Probabilistic generative models used in unsupervised pre-training of deep networks. Inspired contrastive divergence and other energy-based models in deep learning: the first "deep" neural network.

Applications of UGMs

- Neuroscience and associative memory: Hopfield networks (1982, 1984).
  - Energy-based models for pattern recognition and memory retrieval.
- Deep learning: Restricted Boltzmann Machines (1986,2006).
  - Probabilistic generative models used in unsupervised pre-training of deep networks. Inspired contrastive divergence and other energy-based models in deep learning: the first "deep" neural network.
- Natural language processing and large language models (2017)
  - GPT-based large language models capture token dependencies.
  - The architecture is not a UGM (transformers use self-attention) but GPT learns long-range dependencies between tokens (subword) in non-sequential manner (not directional).