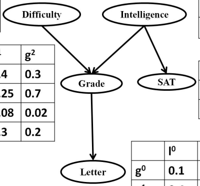
Sampling Algorithms for Probabilistic Graphical models

Vibhav Gogate

THE UNIVERSITY OF TEXAS AT DALLAS
Erik Jonsson School of Engineering and Computer Science

Bayesian Networks



d^0	d^1
0.6	0.4

i^0	i^1
0.7	0.3

	g^0	g^1	g^2
i^0,d^0	0.3	0.4	0.3
i^0,d^1	0.05	0.25	0.7
i^1,d^0	0.9	0.08	0.02
i^1,d^1	0.5	0.3	0.2

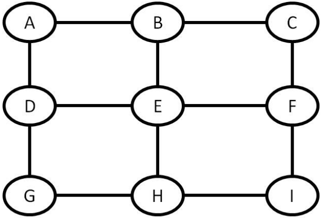
	s^0	s^1
i^0	0.95	0.05
i^1	0.2	0.8

	l^0	l^1
g^0	0.1	0.9
g^1	0.4	0.6
g^2	0.99	0.01

- CPTS: $P(X_i|pa(X_i))$
- Joint Distribution: $P(X) = \prod_{i=1}^{N} P(X_i|pa(X_i))$
- $P(D, I, G, S, L) = P(D)P(I)P(G|D, I)P(S|I)P(L|G)$

- Common Inference Tasks
  - Probability of Evidence: $P(L = l^0, S = s^1) = ?$
  - Posterior Marginal Estimation: $P(D = d^1|L = l^0, S = s^1) = ?$

Markov networks



Graphical model

- Edge Potentials: $\phi_{i,j}$
- Node Potentials: ϕ_i
- Joint Distribution:

$$P(x) = \frac{1}{Z} \prod_{i,j \in E} \phi_{i,j}(x) \prod_{i \in V} \phi_i(x)$$

- Common Inference Tasks
  - Compute the partition function: $Z = ?$.
  - Posterior Marginal Estimation: $P(D = d^1 | I = i^1) = ?$.

Inference tasks: Definitions

- Probability of Evidence (or the partition function)

$$P(E = e) = \sum_{X \setminus E} \prod_{i=1}^{n} P(X_i | pa(X_i))|_{E=e}$$

$$Z = \sum_{x \in X} \prod_{i} \phi_i(x)$$

- Posterior marginals (belief updating)

$$\begin{aligned} P(X_i = x_i | E = e) &= \frac{P(X_i = x_i, E = e)}{P(E = e)} \\ &= \frac{\sum_{X \setminus E \cup X_i} \prod_{i=1}^{n} P(X_i | pa(X_i))|_{E=e, X_i = x_i}}{\sum_{X \setminus E} \prod_{i=1}^{n} P(X_i | pa(X_i))|_{E=e}} \end{aligned}$$

Why Approximate Inference?

- Both problems are #P-complete.
  - Computationally intractable. No hope!
- A tractable class: When the treewidth of the graphical model is small (< 25).
  - Most real world problems have high treewidth.
- In many applications, approximations are sufficient.
  - $P(X_i = x_i | E = e) = 0.29292$
  - Approximate inference yields $\hat{P}(X_i = x_i | E = e) = 0.3$
  - Buy the stock X_i if $P(X_i = x_i | E = e) < 0.4$.

What we will cover today

- Sampling fundamentals
- Monte Carlo techniques
  - Rejection Sampling
  - Likelihood Weighting
  - Importance sampling
- Markov Chain Monte Carlo techniques
  - Metropolis-Hastings
  - Gibbs sampling
- Advanced Schemes
  - Advanced Importance sampling schemes
  - Rao-Blackwellisation

What is a sample and how to generate one?

- Given a set of variables $X = \{X_1, \ldots, X_n\}$, a sample is an instantiation or an assignment to all variables.

$$x^t = (x_1^t, \ldots, x_n^t)$$

- Algorithm to draw a sample from a univariate distribution $P(X_i)$. Domain of $X_i = \{x_i^0, \ldots, x_i^{k-1}\}$
  1. Divide a real line $[0, 1]$ into k intervals such that the width of the j-th interval is proportional to $P(X_i = x_i^j)$
  2. Draw a random number $r \in [0, 1]$
  3. Determine the region j in which r lies. Output x_i^j
- Example
  1. Domain of $X_i = \{x_i^0, x_i^1, x_i^2, x_i^3\}$; $P(X_i) = (0.3, 0.25, 0.27, 0.18)$
  2. Random numbers:
     (a) r=0.2929. $X_i = ?$,
     (b) r=0.5209. $X_i = ?$.

What is a sample and how to generate one?

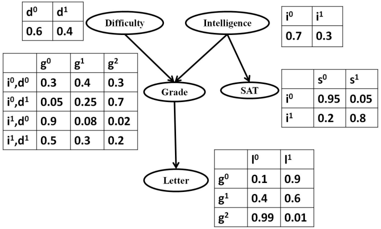
- Given a set of variables $X = \{X_1, \ldots, X_n\}$, a sample is an instantiation or an assignment to all variables.

$$x^t = (x_1^t, \ldots, x_n^t)$$

- Algorithm to draw a sample from a univariate distribution $P(X_i)$. Domain of $X_i = \{x_i^0, \ldots, x_i^{k-1}\}$
  1. Divide a real line $[0, 1]$ into k intervals such that the width of the j-th interval is proportional to $P(X_i = x_i^j)$
  2. Draw a random number $r \in [0, 1]$
  3. Determine the region j in which r lies. Output x_i^j
- Example
  1. Domain of $X_i = \{x_i^0, x_i^1, x_i^2, x_i^3\}$; $P(X_i) = (0.3, 0.25, 0.27, 0.18)$
  2. Random numbers:
     (a) r=0.2929. $X_i =?$,
     (b) r=0.5209. $X_i =?$.

Sampling from a Bayesian network (Logic Sampling)

- Sample variables one by one in a topological order (parents of a node before the node)



d^0	d^1
0.6	0.4

i^0	i^1
0.7	0.3

	g^0	g^1	g^2
i^0,d^0	0.3	0.4	0.3
i^0,d^1	0.05	0.25	0.7
i^1,d^0	0.9	0.08	0.02
i^1,d^1	0.5	0.3	0.2

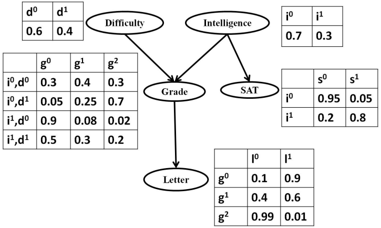
	s^0	s^1
i^0	0.95	0.05
i^1	0.2	0.8

	l^0	l^1
g^0	0.1	0.9
g^1	0.4	0.6
g^2	0.99	0.01

- Sample Difficulty from $P(D)$.
  $r = 0.723$. $D =?$
- Sample Intelligence from $P(I)$.
  $r = 0.34$. $I =?$.

Sampling from a Bayesian network (Logic Sampling)

- Sample variables one by one in a topological order (parents of a node before the node)



d^0	d^1
0.6	0.4

i^0	i^1
0.7	0.3

	g^1	g^2	g^3
i^0,d^0	0.3	0.4	0.3
i^0,d^1	0.05	0.25	0.7
i^1,d^0	0.9	0.08	0.02
i^1,d^1	0.5	0.3	0.2

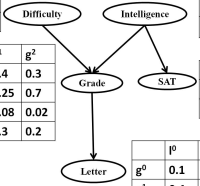
	s^0	s^1
i^0	0.95	0.05
i^1	0.2	0.8

	l^0	l^1
g^1	0.1	0.9
g^2	0.4	0.6
g^3	0.99	0.01

- Sample Difficulty from $P(D)$.
  $r = 0.723.\ D = d^1$
- Sample Intelligence from $P(I)$.
  $r = 0.349.\ I = i^0$.
- Sample Grade from $P(G|i^0, d^1)$.
  $r = 0.281,\ G = ?$.
- Sample SAT from $P(S|i^0)$.
  $r = 0.992,\ S = ?$.

Sampling from a Bayesian network (Logic Sampling)

- Sample variables one by one in a topological order (parents of a node before the node)



d^0	d^1
0.6	0.4

i^0	i^1
0.7	0.3

	g^0	g^1	g^2
i^0,d^0	0.3	0.4	0.3
i^0,d^1	0.05	0.25	0.7
i^1,d^0	0.9	0.08	0.02
i^1,d^1	0.5	0.3	0.2

	s^0	s^1
i^0	0.95	0.05
i^1	0.2	0.8

	l^0	l^1
g^0	0.1	0.9
g^1	0.4	0.6
g^2	0.99	0.01

Sample $= (d^1, i^0, g^1, s^1, l^0)$

- Sample Difficulty from $P(D)$. $r = 0.723$. $D = d^1$
- Sample Intelligence from $P(I)$. $r = 0.349$. $I = i^0$.
- Sample Grade from $P(G|i^0, d^1)$. $r = 0.281$, $G = g^1$.
- Sample SAT from $P(S|i^0)$. $r = 0.992$, $S = s^1$.
- Sample Letter from $P(L|g^1)$. $r = 0.034$, $L = l^0$.

Main idea in Monte Carlo Estimation

- Express the given task as an expected value of a random variable.

$$E_P[g(x)] = \sum_x g(x)P(x)$$

- Generate samples from the distribution P with respect to which the expectation was taken.
- Estimate the expected value from the samples using:

$$\hat{g} = \frac{1}{T}\sum_{i=1}^{T} g(x^t)$$

where $x^1, \ldots, x^T$ are independent samples from P.

Properties of the Monte Carlo Estimate

- **Convergence:** By law of large numbers

$$\hat{g} = \frac{1}{T}\sum_{i=1}^{T} g(x^t) \rightarrow E_P[g(x)] \text{ for } T \rightarrow \infty$$

- **Unbiased:**

$$E_P[\hat{g}] = E_P[g(x)]$$

- **Variance:**

$$V_P[\hat{g}] = V_P\left[\frac{1}{T}\sum_{t=1}^{T} g(x^t)\right] = \frac{V_P[g(x)]}{T}$$

  Thus, variance of the estimator can be reduced by increasing the number of samples. We have no control over the numerator when P is given.

What we will cover today

- Sampling fundamentals
- Monte Carlo techniques
  - Rejection Sampling
  - Likelihood Weighting
  - Importance sampling
- Markov Chain Monte Carlo techniques
  - Metropolis-Hastings
  - Gibbs sampling
- Advanced Schemes
  - Advanced Importance sampling schemes
  - Rao-Blackwellisation

Rejection Sampling

- Express $P(E = e)$ as an expectation problem.

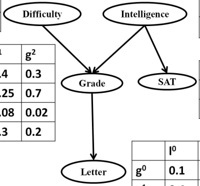
$$\begin{aligned} P(E = e) &= \sum_{x} \delta_e(x) P(x) \\ &= E_P[\delta_e(x)] \end{aligned}$$

  where $\delta_e(x)$ is a dirac-delta function which is 1 if x contains $E = e$ and 0 otherwise.
- Generate samples from the Bayesian network.
- Monte Carlo estimate:

$$\hat{P}(E = e) = \frac{\text{Number of samples that have } E = e}{\text{Total number of samples}}$$

- Issues: If $P(E = e)$ is very small (e.g., 10^{-55}), all samples will be rejected.

Rejection Sampling: Example



d^0	d^1
0.6	0.4

i^0	i^1
0.7	0.3

	g^0	g^1	g^2
i^0,d^0	0.3	0.4	0.3
i^0,d^1	0.05	0.25	0.7
i^1,d^0	0.9	0.08	0.02
i^1,d^1	0.5	0.3	0.2

	s^0	s^1
i^0	0.95	0.05
i^1	0.2	0.8

	l^0	l^1
g^0	0.1	0.9
g^1	0.4	0.6
g^2	0.99	0.01

- Let the evidence be $e = (i^0, g^1, s^1, l^0)$
- Probability of evidence $= 0.00475$
- On an average, you will need approximately $1/0.00475 \approx 210$ samples to get a non-zero estimate for $P(E = e)$.
- Imagine how many samples will be needed if $P(E = e)$ is small!

Importance Sampling

- Use a proposal distribution $Q(Z = X \setminus E)$ satisfying $P(Z = z, E = e) > 0 \Rightarrow Q(Z = z) > 0$. Express $P(E = e)$ as follows:

$$\begin{aligned} P(E=e) &= \sum_z P(Z=z, E=e) \\ &= \sum_z P(Z=z, E=e) \frac{Q(Z=z)}{Q(Z=z)} \\ &= E_Q\left[\frac{P(Z=z, E=e)}{Q(Z=z)}\right] = E_Q[w(z)] \end{aligned}$$

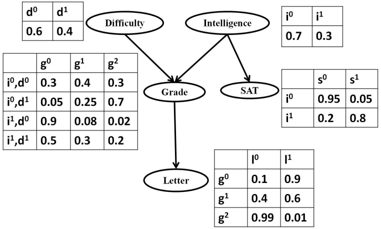
- Generate samples from Q and estimate using $P(E = e)$ using the following Monte Carlo estimate:

$$\hat{P}(E=e) = \frac{1}{T}\sum_{t=1}^{T} \frac{P(Z=z^t, E=e)}{Q(Z=z^t)} = \frac{1}{T}\sum_{t=1}^{T} w(z^t)$$

where $(z^1, \ldots, z^T)$ are sampled from Q.

Importance Sampling: Example

- Let l^1, s^0 be the evidence
- Imagine a uniform Q defined over (D, I, G) and the following samples are generated.
- $\hat{P}(E = e) =$ Average of $\{P(sample, evidence)/Q(sample)\}$



d^0	d^1
0.6	0.4

i^0	i^1
0.7	0.3

	g^0	g^1	g^2
i^0,d^0	0.3	0.4	0.3
i^0,d^1	0.05	0.25	0.7
i^1,d^0	0.9	0.08	0.02
i^1,d^1	0.5	0.3	0.2

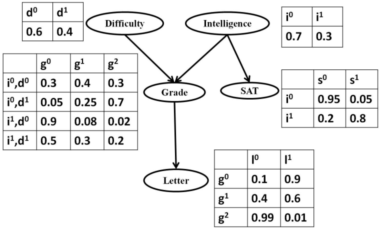
	s^0	s^1
i^0	0.95	0.05
i^1	0.2	0.8

	l^0	l^1
g^0	0.1	0.9
g^1	0.4	0.6
g^2	0.99	0.01

- $sample = (d^0, i^0, g^0)$, $P(sample, evidence) = 0.6 \times 0.7 \times 0.3 \times 0.9 \times 0.95$, $Q(sample) = 0.5 \times 0.5 \times 0.333$
- $sample = (d^1, i^0, g^0)$, $P(sample, evidence) = 0.4 \times 0.7 \times 0.05 \times 0.9 \times 0.95$, $Q(sample) = 0.5 \times 0.5 \times 0.333$
- and so on

Likelihood weighting

- A special kind of Importance sampling in which Q equals the network obtained by clamping evidence.
- Evidence $= (g^0, s^0)$



d^0	d^1
0.6	0.4

i^0	i^1
0.7	0.3

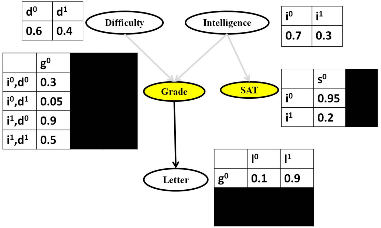
	g^0	g^1	g^2
i^0,d^0	0.3	0.4	0.3
i^0,d^1	0.05	0.25	0.7
i^1,d^0	0.9	0.08	0.02
i^1,d^1	0.5	0.3	0.2

	s^0	s^1
i^0	0.95	0.05
i^1	0.2	0.8

	l^0	l^1
g^0	0.1	0.9
g^1	0.4	0.6
g^2	0.99	0.01

Likelihood weighting

- A special kind of Importance sampling in which Q equals the network obtained by clamping evidence.
- Evidence $= (g^0, s^0)$



- P(sample,evidence)/Q(sample) can be efficiently computed.
- The ratio equals the product of the corresponding CPT values at the evidence nodes. The remaining values cancel out.
- Let the sample $= (d^0, i^0, l^1)$.

$$\frac{P(sample, evidence)}{Q(sample)} = 0.3 \times 0.95$$

Normalized Importance sampling

- (Un-normalized) IS is not suitable for estimating $P(X_i = x_i | E = e)$.
- One option: Estimate the numerator and denominator by IS.

$$\hat{P}(X_i = x_i | E = e) = \frac{\hat{P}(X_i = x_i, E = e)}{\hat{P}(E = e)}$$

- This ratio estimate is often very bad because the numerator and denominator errors may be cumulative and may have a different source.
  - For example, if the numerator is an under-estimate and the denominator is an over-estimate.
- How to fix this? Use: Normalized importance sampling.

Normalized Importance sampling: Theory

- Given a dirac delta function $\delta_{x_i}(z)$ (which is 1 if z contains $X_i = x_i$ and 0 otherwise), we can write $P(X_i = x_i|E = e)$ as:

$$P(X_i = x_i|E = e) \quad = \quad \frac{\sum_z \delta_{x_i}(z) P(Z = z, E = e)}{\sum_z P(Z = z, E = e)}$$

- Now we can use the same Q and samples from it to estimate both the numerator and the denominator.

$$\hat{P}(X_i = x_i|E = e) = \frac{\sum_{t=1}^{T} \delta_{x_i}(z^t) w(z^t)}{\sum_{t=1}^{T} w(z^t)}$$

- This reduces variance because of common random numbers. (Read about it on Wikipedia. Not covered in standard machine learning texts.)

Normalized Importance sampling: Properties

- Asymptotically Unbiased:

$$\lim_{T\to\infty} E_Q[\hat{P}(X_i = x_i|E = e)] = P(X_i = x_i|E = e)$$

- Variance: Harder to analyze
- Liu (2003) suggests a performance measure called effective sample size
  - Definition:

$$ESS(Ideal, Q) = \frac{1}{1 + V_Q[w(z)]}$$

  - It means that T samples from Q are worth only $T/(1 + V_Q[w(z)])$ samples from the ideal proposal distribution.

Importance sampling: Issues

- For optimum performance, the proposal distribution Q should be as close as possible to $P(X|E = e)$.
  - When $Q = P(X|E = e)$, the weight of every sample is $P(E = e)$! However, achieving this is NP-hard.
- Likelihood weighting performs poorly when evidence is at the leaves and is unlikely.
- In particular, designing a good proposal distribution is an art rather than a science!

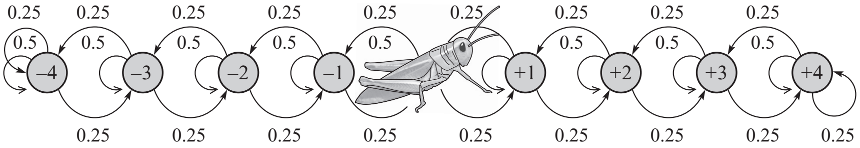
What we will cover today

- Sampling fundamentals
- Monte Carlo techniques
  - Rejection Sampling
  - Likelihood Weighting
  - Importance sampling
- Markov Chain Monte Carlo techniques
  - Metropolis-Hastings
  - Gibbs sampling
- Advanced Schemes
  - Advanced Importance sampling schemes
  - Rao-Blackwellisation

Markov Chains

A Markov chain is composed of:

- A set of states $Val(\mathbf{X}) = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_r\}$
- A process that moves from a state $\mathbf{x}$ to another state $\mathbf{x}'$ with probability $\mathcal{T}(\mathbf{x} \to \mathbf{x}')$.
  - $\mathcal{T}$ is a square matrix in which each row and column sums to 1



- Chain Dynamics

$$P^{(t+1)}(\mathbf{X}^{(t+1)} = \mathbf{x}') = \sum_{\mathbf{x} \in Val(\mathbf{X})} P^{(t)}(\mathbf{X}^{(t)} = \mathbf{x})\mathcal{T}(\mathbf{x} \to \mathbf{x}')$$

- Markov chain Monte Carlo sampling is a process that mirrors the dynamics of a Markov chain.

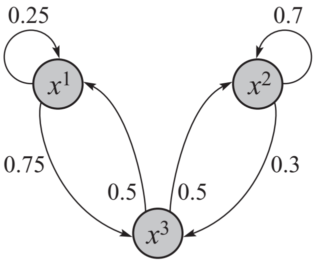
Markov Chains: Stationary Distribution

We are interested in the long-term behavior of a Markov chain, which is defined by the stationary distribution.

- A distribution $\pi(\mathbf{X})$ is a stationary distribution if it satisfies:

$$\pi(\mathbf{X} = \mathbf{x}') = \sum_{\mathbf{x} \in Val(\mathbf{X})} \pi(\mathbf{X} = \mathbf{x}) \mathcal{T}(\mathbf{x} \to \mathbf{x}')$$

- A Markov chain may or may not converge to a stationary distribution.



Constraints:

- $\pi(x^1) = 0.25\pi(x^1) + 0.5\pi(x^3)$
- $\pi(x^2) = 0.7\pi(x^2) + 0.5\pi(x^3)$
- $\pi(x^3) = 0.75\pi(x^1) + 0.3\pi(x^2)$
- $\pi(x^1) + \pi(x^2) + \pi(x^3) = 1$

Unique Solution: $\pi(x^1) = 0.2$, $\pi(x^2) = 0.5$, $\pi(x^3) = 0.3$.

Sufficient Conditions for ensuring convergence

- Regular Markov chain: A Markov chain is said to be regular if there exists some number k such that for every $\mathbf{x}, \mathbf{x}' \in Val(\mathbf{X})$, the probability of getting from $\mathbf{x}$ to $\mathbf{x}'$ in exactly k steps is greater than zero.

Theorem

If a finite Markov chain is regular and is defined over a finite space, then it has a unique stationary distribution.

- Sufficient conditions for ensuring Regularity:
  - Construct the state graph such that there is a positive probability to get from any state to any state.
  - For each state $\mathbf{x}$, there is a positive probability self-loop.

MCMC for computing $P(X_i = x_i | E = e)$

- Main idea: Construct a Markov chain such that its stationary distribution equals $P(X|E = e)$.
- Generate samples using the Markov chain
- Estimate $P(X_i = x_i | E = e)$ using the standard Monte Carlo estimate:

$$\hat{P}(X_i = x_i | E = e) = \frac{1}{T} \sum_{t=1}^{T} \delta_{x_i}(z^t)$$

Gibbs sampling

- Start at a random assignment to all non-evidence variables.
- Select a variable X_i and compute the distribution $P(X_i | E = e, \mathbf{x}_{-i})$ where $\mathbf{x}_{-i}$ is the current sampled assignment to $X \setminus E \cup X_i$.
- Sample a value for X_i from $P(X_i | E = e, \mathbf{x}_{-i})$. Repeat.

Question: Can we compute $P(X_i | E = e, \mathbf{x}_{-i})$ efficiently?

Gibbs sampling

- Start at a random assignment to all non-evidence variables.
- Select a variable X_i and compute the distribution $P(X_i|E = e, \mathbf{x}_{-i})$ where $\mathbf{x}_{-i}$ is the current sampled assignment to $X \setminus E \cup X_i$.
- Sample a value for X_i from $P(X_i|E = e, \mathbf{x}_{-i})$. Repeat.
- Computing $P(X_i|E = e, \mathbf{x}_{-i})$
  - Exact inference is possible because only one variable is not assigned a value!
- The stationary distribution of the Markov chain equals $P(X|E = e)$ (easy to prove).

Gibbs sampling: Properties

- When the Bayesian network has no zeros, Gibbs sampling is guaranteed to converge to $P(X|E=e)$
- When the Bayesian network has zeros or the Evidence is complex (e.g., a SAT formula), Gibbs sampling may not converge.
  - Open problem!
- Mixing time: Let t_ϵ be the minimum t such that for any starting distribution $P^{(0)}$, the distance between $P(X|E=e)$ and $P^{(t)}$ is less than ϵ.
  - It is common to ignore some number of samples at the beginning, the so-called burn-in period, and then consider only every nth sample.

Metropolis-Hastings: Theory

Detailed Balance: Given a transition function $\mathcal{T}(\mathbf{x} \to \mathbf{x}')$ and an acceptance probability $A(\mathbf{x} \to \mathbf{x}')$, a Markov chain satisfies the detailed balance condition if there exists a distribution π such that:

$$\pi(\mathbf{x})\mathcal{T}(\mathbf{x} \to \mathbf{x}')A(\mathbf{x} \to \mathbf{x}') = \pi(\mathbf{x}')\mathcal{T}(\mathbf{x}' \to \mathbf{x})A(\mathbf{x}' \to \mathbf{x})$$

Theorem

If a Markov chain is regular and satisfies the detailed balance condition relative to π, then it has a unique stationary distribution π.

Metropolis-Hastings: Algorithm

Input: Current state $\mathbf{x}^t$
Output: Next state $\mathbf{x}^{t+1}$

- Draw $\mathbf{x}'$ from $\mathcal{T}(\mathbf{x}^t \to \mathbf{x}')$
- Draw a random number $r \in [0, 1]$ and update

$$\mathbf{x}^{t+1} = \begin{cases} \mathbf{x}' & \text{if } r \leq A(\mathbf{x}^t \to \mathbf{x}') \\ \mathbf{x}^t & \text{otherwise} \end{cases}$$

In Metropolis-Hastings A is defined as follows:

$$A(\mathbf{x} \to \mathbf{x}') = min\left\{1, \frac{\pi(\mathbf{x}')\mathcal{T}(\mathbf{x} \to \mathbf{x}')}{\pi(\mathbf{x})\mathcal{T}(\mathbf{x}' \to \mathbf{x})}\right\}$$

Theorem

The Metropolis Hastings algorithm satisfies the detailed balance condition.

Metropolis-Hastings: What "T" to use?

- Use an importance distribution Q to make transitions. This is called independent sampling because the transition function $\mathcal{T}$ does not depend on what state you are currently in.
- Use a random walk approach.

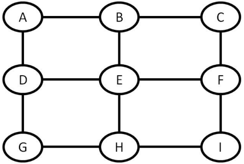
What we will cover today

- Sampling fundamentals
- Monte Carlo techniques
  - Rejection Sampling
  - Likelihood Weighting
  - Importance sampling
- Markov Chain Monte Carlo techniques
  - Metropolis-Hastings
  - Gibbs sampling
- Advanced Schemes
  - Advanced Importance sampling schemes
  - Rao-Blackwellisation

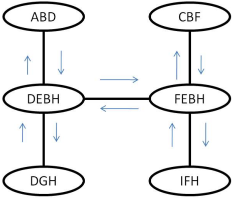
Selecting a Proposal Distribution

- For good performance Q should be as close as possible to $P(X|E = e)$.
- Use a method that yields a good approximation of $P(X|E = e)$ to construct Q
  - Variational Inference
  - Generalized Belief Propagation
- Update the proposal distribution from the samples (the machine learning approach)
- Combinations!

Using Approximations of $P(X|E = e)$ to construct Q



Graphical model

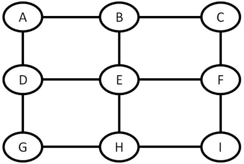


Junction tree

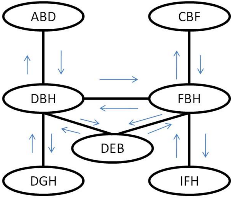
Algorithm JT-sampling (Perfect sampling)

- Let $o = (X_1, \ldots, X_n)$ be an ordering of variables
- $q = 1$
- For $i = 1$ to n do
  - Propagate evidence in the junction tree
  - Construct a distribution $Q_i(X_i)$ by referring to any cluster mentioning X_i and marginalizing out all other variables.
  - Sample $X_i = x_i$ from Q_i, $q = q \times Q_i(X_i = x_i)$
  - Set $X_i = x_i$ as evidence in the junction tree.
- Return (x, q)

Using Approximations of $P(X|E=e)$ to construct Q



Graphical model



Join graph

Algorithm IJGP-sampling (Gogate&Dechter, UAI, 2005)

- Let $o = (X_1, \ldots, X_n)$ be an ordering of variables
- $q = 1$
- For $i = 1$ to n do
  - Propagate evidence in the join graph.
  - Construct a distribution $Q_i(X_i)$ by referring to any cluster mentioning X_i and marginalizing out all other variables.
  - Sample $X_i = x_i$ from Q_i, $q = q \times Q_i(X_i = x_i)$
  - Set $X_i = x_i$ as evidence in the join graph.
- Return (x, q)

Adaptive Importance sampling

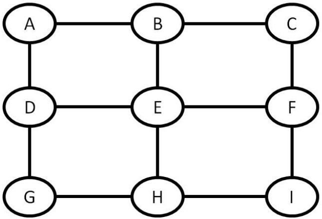
- Machine learning view of sampling: Learn from experience!
- Learn a proposal distribution Q' from the samples.
- At regular intervals, update the proposal distribution Q^t at the current interval t using:
$$Q^{t+1} = Q^t + \alpha(Q^t - Q')$$
where α is the learning rate.
- As the number of samples increases, the proposal will get closer and closer to $P(X|E = e)$.

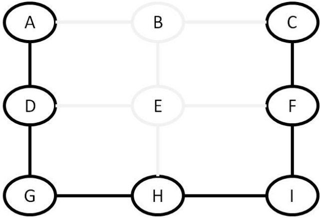
Rao-Blackwellisation of sampling schemes

- Combine exact inference with sampling.
  - Sample a few variables and analytically marginalize out other variables.
- Inference on trees (or low treewidth) graphical models is always tractable. Sample variables until the graphical model is a tree.
- Rao-Blackwell theorem: Let the non-evidence variables Z be partitioned into two sets Z_1 and Z_2, where Z_1 are sampled and Z_2 are inferred exactly. Then,

$$V_Q\left[\frac{P(z_1, z_2, e)}{Q(z_1, z_2)}\right] \geq V_Q\left[\frac{P(z_1, e)}{Q(z_1)}\right]$$

Rao-Blackwellisation of sampling schemes: Example





Traditional importance sampling

$$\hat{\Gamma} = \frac{1}{T}\sum_{t=1}^{T} \frac{F(a^t,\ldots,i^t)}{Q(a^t,\ldots,i^t)}$$

Proposal distribution and samples defined over all variables.

Rao-Blackwellised importance sampling

Let $Z = Vars \setminus \{B, E\}$

$$\hat{\Gamma} = \frac{1}{T}\sum_{t=1}^{T} \frac{\sum_z F(z,b^t,e^t)}{Q(b^t,e^t)}$$

$\sum_z F(z,b^t,e^t)$ is computed efficiently using Belief Propagation or Variable Elimination.

Summary

- Importance sampling
  - Generate samples from a proposal distribution
  - Performance depends on how close the proposal is to the posterior distribution
- Markov chain Monte Carlo (MCMC) sampling
  - Attempts to generate samples from the posterior distribution by creating a Markov chain whose stationary distribution equals the posterior distribution
  - Metropolis-Hastings and Gibbs sampling
- Advanced schemes
  - How to construct and learn a good proposal distribution.
  - How to use graph decompositions to improve the quality of estimation.