Approximate marginal inference

- Given the joint \( p(x_1, \ldots, x_n) \) represented as a graphical model, how do we perform **marginal inference**, e.g. to compute \( p(x_1 | e) \)?
- We showed in Lecture 4 that doing this exactly is NP-hard
- Nearly all **approximate inference** algorithms are either:
  1. **Monte-carlo methods** (e.g., Gibbs sampling, likelihood reweighting, MCMC)
  2. Variational algorithms (e.g., mean-field, loopy belief propagation)
Generating samples from a Bayesian network

Algorithm 12.1 Forward Sampling in a Bayesian network

Procedure Forward-Sample (B // Bayesian network over $X$
)

1. Let $X_1, \ldots, X_n$ be a topological ordering of $X$

2. for $i = 1, \ldots, n$

3. $u_i \leftarrow x\langle Pa_{X_i}\rangle$ // Assignment to $Pa_{X_i}$ in $x_1, \ldots, x_{i-1}$

4. Sample $x_i$ from $P(X_i | u_i)$

5. return $(x_1, \ldots, x_n)$

Monte-Carlo algorithms

- Given a joint distribution \( p(x_1, \ldots, x_n) \), how do we compute marginals?

\[
p[X_1 = x_1] = E_{x \sim p}[f(x)], \text{ where } f(x) = 1[X_1 = x_1]
\]
\[
= \sum_x p(x)f(x).
\]

- Rather than explicitly enumerating all assignments, consider the following Monte-Carlo estimate of the expectation:

\[
x^1 \sim p(x) \\
x^2 \sim p(x) \\
\vdots \\
x^M \sim p(x)
\]

Then, our estimate is \( \hat{E}_p[f(x)] = \frac{1}{M} \sum_{m=1}^{M} f(x^m) \). How good is it?
Monte-Carlo algorithms

- Let $\mathcal{D} = \{x^1, \ldots, x^M\}$. Since $\mathcal{D}$ was drawn randomly from $p(x)$, the estimate is itself a random variable.
- The estimate is \textit{unbiased} because

\[
E_{x^1, \ldots, x^M \sim p(x)} \left[ \hat{E}[f(x)] \right] = E_{x^1, \ldots, x^M \sim p(x)} \left[ \frac{1}{M} \sum_{m=1}^{M} f(x^m) \right] 
\]

\[
= \frac{1}{M} \sum_{m=1}^{M} E_{x^m \sim p(x)}[f(x^m)] 
\]

\[
= E_{x \sim p(x)}[f(x^m)]. 
\]

- How quickly does the estimate converge to the true expectation?
There are two general results we can use, depending on whether we care about additive or multiplicative error.

**Hoeffding bound** says that:

\[
\Pr_{D \sim p(x)} \left[ E_p[f(x)] - \epsilon \leq \hat{E}_D[f(x)] \leq E_p[f(x)] + \epsilon \right] \geq 1 - 2e^{-2M\epsilon^2}
\]

**Chernoff bound** says that (assuming \( f(x) \in [0, 1] \)):

\[
\Pr_{D \sim p(x)} \left[ E_p[f(x)](1 - \epsilon) \leq \hat{E}_D[f(x)] \leq E_p[f(x)](1 + \epsilon) \right] \geq 1 - 2e^{-\frac{M\epsilon^2}{3} E_p[f(x)]}
\]

Estimating *single-variable* marginals for a BN is easy: just forward sample!

What about computing *conditional* queries such as \( p(X = x \mid E = e) \)?

Computing denominator of \( p(X = x, E = e) / p(E = e) \) needs \( \Omega(1/p(E = e)) \) samples, by Chernoff bound. In this setting, no point in even using a BN, could simply estimate directly from data!
If we could instead directly sample from $p(X \mid E = e)$, we would be in business – but this is hard!

For the same reason, sampling from an undirected graphical model $p(x) = \frac{1}{Z} \prod_{c \in C} \phi_c(x_c)$ – even without evidence – is hard, because we don’t know $Z$

Suppose we instead had a simpler-to-sample-from distribution $q(x)$, called the “proposal distribution”

Let $\tilde{p}(x)$ be an unnormalized version of the distribution, e.g.

$$\tilde{p}(x) = p(x, E = e) \quad (\text{BN with evidence})$$
$$\tilde{p}(x) = \prod_{c \in C} \phi_c(x_c) \quad (\text{MRF})$$

Note that we can efficiently evaluate $\tilde{p}(x)$ for any $x$
Consider the following estimate (now using $x^1, \ldots, x^M \sim q(x)$):

$$
\hat{E}_D[f(x)] = \frac{1}{M} \sum_{m=1}^{M} f(x^m) \tilde{w}(x^m) \frac{1}{M} \sum_{m=1}^{M} \tilde{w}(x^m),
$$

where \( \tilde{w}(x) = \frac{\tilde{p}(x)}{q(x)} \)

This is not an unbiased estimate! E.g., for $M = 1$, we have

$$
E_{x^1 \sim q(x)}\left[ \hat{E}_D[f(x)] \right] = E_{x^1 \sim q(x)} \left[ \frac{f(x^1) \tilde{w}(x^1)}{\tilde{w}(x^1)} \right] = E_{x \sim q(x)}[f(x)] \\
\neq E_{x \sim p(x)}[f(x)].
$$

However, the estimate is asymptotically correct (i.e., as $M \to \infty$)
"Normalized" Importance Sampling

- Consider the following estimate (now using $x^1, \ldots, x^M \sim q(x)$):

$$\hat{E}_D[f(x)] = \frac{1}{M} \sum_{m=1}^{M} \frac{f(x^m)\tilde{w}(x^m)}{\frac{1}{M} \sum_{m=1}^{M} \tilde{w}(x^m)}, \quad \text{where} \quad \tilde{w}(x) = \frac{\tilde{p}(x)}{q(x)}$$

- Letting \( \tilde{p}(x) = p(x)Z \), the expectation of the numerator is:

$$E_{D \sim q(x)} \left[ \frac{1}{M} \sum_{m=1}^{M} f(x^m)\tilde{w}(x^m) \right] = \frac{1}{M} \sum_{m=1}^{M} E_{x^m \sim q(x)}[f(x^m)\tilde{w}(x^m)]$$

$$= \frac{1}{M} \sum_{m=1}^{M} \sum_{x} q(x) \left[ f(x) \frac{\tilde{p}(x)}{q(x)} \right]$$

$$= \frac{1}{M} \sum_{m=1}^{M} \sum_{x} \tilde{p}(x)f(x) = ZE_p[f(x)].$$
“Normalized” Importance Sampling

- Consider the following estimate (now using $x^1, \ldots, x^M \sim q(x)$):

$$
\hat{E}_D[f(x)] = \frac{1}{M} \sum_{m=1}^{M} f(x^m) \tilde{w}(x^m), \quad \text{where} \quad \tilde{w}(x) = \frac{\tilde{p}(x)}{q(x)}
$$

- Letting $\tilde{p}(x) = p(x)Z$, the expectation of the numerator is $ZE_p[f(x)]$.

- The expectation of the denominator is $Z$!

$$
E_{D\sim q(x)} \left[ \frac{1}{M} \sum_{m=1}^{M} \tilde{w}(x^m) \right] = \frac{1}{M} \sum_{m=1}^{M} E_{x^m \sim q(x)}[\tilde{w}(x^m)]
$$

$$
= \frac{1}{M} \sum_{m=1}^{M} \sum_x q(x) \left[ \frac{\tilde{p}(x)}{q(x)} \right]
$$

$$
= \frac{1}{M} \sum_{m=1}^{M} \sum_x \tilde{p}(x) = Z.
$$
What should we use for \( q(x) \)? For a Bayesian network, we can sample from the latent variables, keeping the evidence fixed.

**Algorithm 12.2 Likelihood-weighted particle generation**

**Procedure** LW-Sample ( 

\[ B, \quad \text{// Bayesian network over } \mathcal{X} \]
\[ Z = z \quad \text{// Event in the network} \]

1. Let \( X_1, \ldots, X_n \) be a topological ordering of \( \mathcal{X} \)
2. \( w \leftarrow 1 \)
3. for \( i = 1, \ldots, n \)
4. \( u_i \leftarrow x^{\text{Pa}X_i} \quad \text{// Assignment to } \text{Pa}X_i \text{ in } x_1, \ldots, x_{i-1} \)
5. if \( X_i \notin Z \) then
6. \quad Sample \( x_i \) from \( P(X_i | u_i) \)
7. else
8. \quad \( x_i \leftarrow z(X_i) \quad \text{// Assignment to } X_i \text{ in } z \)
9. \quad \( w \leftarrow w \cdot P(x_i | u_i) \quad \text{// Multiply weight by probability of desired value} \)
10. return \((x_1, \ldots, x_n), w\)


Corresponds to importance sampling using:

\[
q(x) = \prod_{t \notin E} p(x_t | x_{pa(t)}) \prod_{t \in E} 1[x_t = e_t], \text{ so } \tilde{w}(x) = \frac{\tilde{p}(x)}{q(x)} = \prod_{t \in E} p(x_t | x_{pa(t)}).
\]