Marginalization in Lazy Propagation

Alg. 3.1.1 [Madsen’s Dissertation]

Let \( \Phi = \{ \psi_1, \ldots, \psi_n \} \) be a set of potentials.

1. Set \( \Phi_x = \{ \psi \in \Phi \mid x \in \text{dom} (\psi) \} \)

2. \( \psi_x^* = \sum \prod_{x \in \phi_x} \psi \)

3. \( \Phi^* = \{ \psi^*_x \} \cup \Phi \setminus \Phi_x \)

\( \Phi^* \) is the set of potentials that results from
eliminating $X$ from $\Phi$.

This is Defn. 4.1 on p. 116 with different terminology.

elimination for marginalisation

$\Phi^{x}$ for $x^{e}$.

4.8 Stochastic Simulation in Bayesian Networks

$P(E = y) \approx \frac{\text{N}(E = y)}{N} = \frac{\text{number of cases in which } E = y}{\text{total number of cases}}$
\[ P(B=y) = \frac{N(B=y)}{N} = \frac{62}{100} = 0.62 \]

\[ P(E) = \left( \frac{N(E=y)}{N} \right) \left( \frac{N(E=x)}{N} \right) = \left( \frac{99}{100}, \frac{1}{100} \right) = 0.99, 0.01 \]

The algorithm (p. 168 [JOOZ]).

1. Let \( \langle x_1, \ldots, x_n \rangle \) be a topological ordering of the variables (e.g., \( A, B, C, D, E \)).

2. For \( j = 1 \) to \( N \):
   a) For \( i = 1 \) to \( n \):
      - Sample a state \( x_i \) of \( X_i \) using \( P(X_i | \text{pa}(X_i) = \epsilon) \), where \( \epsilon \) is the configuration already sampled for \( \text{pa}(X_i) \).

b) If \( \underline{x} = (x_1, \ldots, x_n) \) is consistent with \( \epsilon \), then
\[ N(X_k = x_k) := N(X_k = x_k) + 1, \text{ where} \]

\[ x_k \text{ is the state that was sampled for } X_k. \]

[false: the second \( x \) becomes \( \text{it is inconsistent with} \)

the evidence]  

3. Return

\[ P(X_k = x_k | e) = \frac{N(X_k = x_k)}{\sum_{x_k} P(x_k) | (X_k = x_k)} \]

\[ P(X_k = x_k | e) = \frac{P(X_k = x_k | e)}{P(e)} \]

The alg. described requires generating a lot of
irrelevant samples when the evidence has low probability.
4.8.2 Likelihood weighting was designed to overcome this problem.

Idea: to avoid flipping a biased coin for each variable on which there is evidence. **Wrong results!**

\[ P(A | B = \text{true}, E = \text{true}) \]

By following the idea, you estimate

\[ P(A), \ 	ext{not} \ P(A | E), \ 	ext{b/c when the state of} \ A \ \text{is determined, only} \ P(A) \ \text{is used} \]

The idea can be salvaged by carefully writing down

\[ P(E, \text{true}) = \prod_{x \in \text{UN} \cup E} P(x | p_{E}(x), p_{A}(x)'' = \text{true}) \times \prod_{x \in E} P(x = e | p_{E}(x), p_{A}(x)'' = e) \]
The mistaking (naive) algorithm (based on the "rule") ignores the second part!

So the fix is to weigh each sample by

\[
\prod_{x \in \mathcal{E}} P(x = e | p(x), p(e) = e)
\]

\[
(w(x, e) = \prod_{E \in \mathcal{E}} P(E = e | p(E) = e))
\]

configuration in the sample point

In practice, once a sample point (case) has been produced, \(w(x, e)\) is calculated and it is added to \(N(\_\_\_)\) instead of 1.
4.8.3 Gibbs Sampling (used in Nielsen slides, unpublished)

4.9 Loopy Belief Propagation [in 1998?]

Used in "Turbo coding" (McEliece)
\[
\lambda_C(A) = \sum_{B,C} P(C | A, B) \phi_B \phi_D \phi_C
\]
\[
\pi_E(C) = \phi_D \sum_{A,B} P(C | A, B) \phi_A \phi_C
\]

In the special case of BNs that are trees, this algorithm is the same as the junction tree algorithm.

The algorithm, in the case of trees, is due to Pearl (1982).

The algorithm also works for polytrees — BNs whose moral graph is a singly connected graph.
not a polytree, b/c its moral graph is

a polytree, b/c its moral graph is

By the observation that each clique in a jtree for a polytree is small, you may show that belief updates in polytrees can be done in linear time.