Week 1: Quantifying Uncertainty && Bayes Network

1. Quantifying Uncertainty

1.1. Introduction

1.1.1. Definition of Agent

1. sensors → Perceiving Enviroment
2. Actuators → actions

1.1.2. Types of Uncertainty

1. stochasticity

   a. outcome uncertainty
   b. environment fluctuation
2. state uncertainty
   a. sensor limited
   b. noise

1.1.3. Why cares about uncertainty

an agent may never know **its state/sequence of actions** exactly

1.1.4. Traditional way

Belief state → set of all possible states (**no probability**)

1. cons: **qualification problems**: needs to specify all exceptions
2. solution: **rational decision**: goals+likelihood+achieved degree (why we need **probability** tools)

1.1.5. Utility Theory & Decision Theory

1. **Utility Theory** → preference
2. Decision Theory = probability theory + utility theory
3. **rational** agent ⇔it chooses the highest expected utility (**MEU**)
4. Expected Utility:
   a. outcomes uncertainty:

$$U(a) = \sum_o u(o) * b(o|a)$$

   b. states uncertainty: $U(a) = \sum_s u(s,a) * b(s)$

1.1.5. Why probability:

But de Finetti Argument: “If Agent 1 expresses a set of degrees of belief that violate the axioms of probability theory then there is a combination of bets by Agent 2 that guarantees that Agent 1 will lose money every time”

1.1.6. Basic Probability:

1.1.6.1. Full Joint Distributions:

cons: Too large

1.1.6.2. Absolute independence:

pros: helpful when abstract distributions $2n \rightarrow n$

cons: hard to meet

1.1.6.3. Conditional Independence：

$$P(X, Y|Z) \ = \ P(X|Z)P(Y|Z)$$
$$\text{or } \mathbf{P}(X \mid Y, Z) = \mathbf{P}(X \mid Z)$$
$$\text{or } \mathbf{P}(Y \mid X, Z) = \mathbf{P}(Y \mid Z)$$

1.1.6.4. Bayes Rule：

$$P(b|a) = \frac{P(a|b)P(b)}{P(a)} = \frac{P(a|b)P(b)}{\sum P(a|b_i)p(b_i)}$$

$$P(Y|X,e) = P(X|Y,e)P(Y|e)/P(X|e)$$

$$P(Y|X) = \alpha P(X|Y)P(Y)$$

1. enable update probability
2. “diagnostic knowledge is often more fragile than causal knowledge”

2. Probabilistic Reasoning

2.1. Bayesian network

Bayesian network (compact representaion) ⇔ any joint probability distribution

2.1.1. Rules:

1. node → random variable
2. If there is an arrow from node X to node Y, X is said to be a **parent** of Y, intuitively meaning X has a **directly influence** on Y
3. Each node has a conditional probability distribution $P(X_i|Parents(X_i))$
4. Acyclic

2.1.2. Semantics

2.1.2.1. Representing full joint distribution

the conditional probability distribution ensured that the Bayesian Network can representing full joint distribution comprehensively

$$P(x_1, x_2, ..., x_n) = \prod_{i=1}^{n} P(x_i|Parents(X_i))$$

2.1.2.2. conditional independence relations

1. given its **parents**, each variables is conditionally independent of its **non-descendants**
2. given its **Markov blanket**(parents, children, children’s parents) → all other nodes

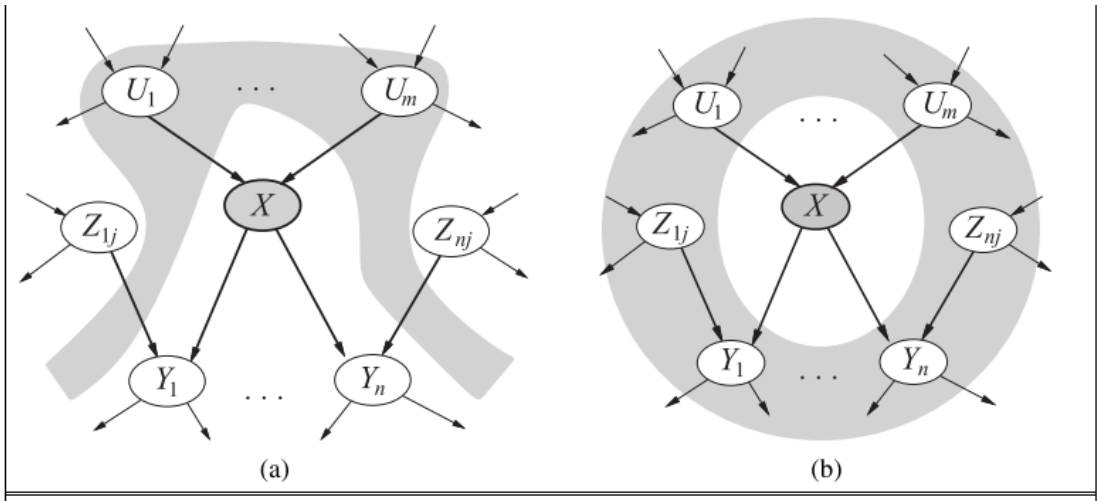


Figure 14.4 (a) A node X is conditionally independent of its non-descendants (e.g., the Z_{ij}s) given its parents (the U_is shown in the gray area). (b) A node X is conditionally independent of all other nodes in the network given its Markov blanket (the gray area).

2.1.3. Ways of constructing

Based on chain rule:

$$\mathbf{P}\left(X_i \mid X_{i-1}, \ldots, X_1\right) = \mathbf{P}\left(X_i \mid \text{ Parents } \left(X_i\right)\right) (14.3)$$

1. *Nodes:* First determine the set of variables that are required to model the domain. Now order them, $\{X_1, \ldots, X_n\}$. Any order will work, but the resulting network will be more compact if the variables are ordered such that causes precede effects.
2. *Links:* For $i = 1$ to n do:
   - Choose, from $X_1, \ldots, X_{i-1}$, a minimal set of parents for X_i, such that Equation (14.3) is satisfied.
   - For each parent insert a link from the parent to X_i.
   - CPTs: Write down the conditional probability table, $\mathbf{P}(X_i|Parents(X_i))$.

2.2. Exact Inference in Bayesian Networks

2.2.1. Variables

$$P(X|e)$$

1. Query Variables: X ;
2. Evidence Variables: Always the condition variables, e;
3. Hidden Variables: non-evidence non query variables;

2.2.2. Inference by enumeration

1. Using Law of total cumulance:

$$P(X|e) \;=\; \alpha P(X, e) \;=\; \alpha \sum y P(X, e, y)$$

2. Using Bayes Network:

$$P(B|j,m) = \alpha \sum_{e} \sum_{a} P(B, j, m, e, a)$$

$$\Rightarrow \text{using } P(x_1, x_2, ..., x_n) = \prod_{i=1}^{n} P(x_i | Parents(X_i))$$

2.2.3. Variable Elimination Algorithm

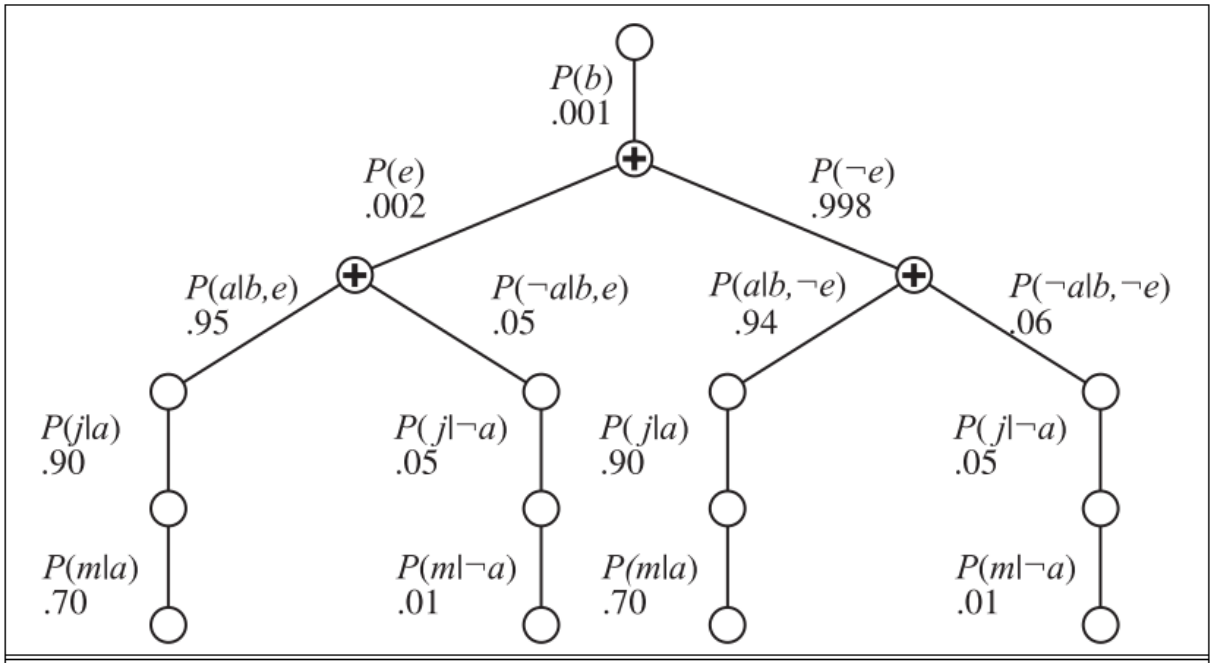


Figure 14.8 The structure of the expression shown in Equation (14.4). The evaluation proceeds top down, multiplying values along each path and summing at the "+" nodes. Notice the repetition of the paths for j and m.

By using point pointwise product of a pair of factors, and summing out a variable from a product of factors:

- First push in the summations as far as possible
- Then carry out summations **right-to-left**, caching intermediate results in new **factors**

$$
\begin{aligned}
&\mathbf{P}(B|j,m) \\
&\quad = \alpha \underbrace{\mathbf{P}(B)}_{B} \Sigma_e \underbrace{P(e)}_{E} \Sigma_a \underbrace{\mathbf{P}(a|B,e)}_{A} \underbrace{P(j|a)}_{J} \underbrace{P(m|a)}_{M} \\
&\quad = \alpha \mathbf{P}(B) \Sigma_e P(e) \Sigma_a \mathbf{P}(a|B,e) P(j|a) f_M(a) \\
&\quad = \alpha \mathbf{P}(B) \Sigma_e P(e) \Sigma_a \mathbf{P}(a|B,e) f_J(a) f_M(a) \\
&\quad = \alpha \mathbf{P}(B) \Sigma_e P(e) \Sigma_a f_A(a,b,e) f_J(a) f_M(a) \\
&\quad = \alpha \mathbf{P}(B) \Sigma_e P(e) f_{\bar{A}JM}(b,e) \text{ (sum out } A\text{)} \\
&\quad = \alpha \mathbf{P}(B) f_{\bar{E}\bar{A}JM}(b) \text{ (sum out } E\text{)} \\
&\quad = \alpha f_B(b) \times f_{\bar{E}\bar{A}JM}(b)
\end{aligned}
$$

The time and space requirements of variable elimination are dominated by the size of the largest factor constructed during the operation of the algorithm

One fairly effective method is a **greedy one**: eliminate whichever variable minimizes the size of the next factor to be constructed

2.3. Approximate Inference

Typically two ways:

1. Based on sampling
2. Based on optimization

We only covered method based on sampling

2.3.1. Ancestral Sampling

The idea is to sample each variable in turn, in topological order

function PRIOR-SAMPLE(bn) **returns** an event sampled from bn
  inputs: bn, a belief network specifying joint distribution $\mathbf{P}(X_1, \ldots, X_n)$

  $\mathbf{x} \leftarrow$ an event with n elements
  for $i = 1$ **to** n **do**
    $x_i \leftarrow$ a random sample from $\mathbf{P}(X_i \mid parents(X_i))$
      given the values of $Parents(X_i)$ in $\mathbf{x}$
  return $\mathbf{x}$

The consistency can be proved from law of large numbers:

Probability that PRIORSAMPLE generates a particular event
$S_{PS}(x_1 \ldots x_n) = \prod_{i=1}^{n} P(x_i | parents(X_i)) = P(x_1 \ldots x_n)$
i.e., the true prior probability

E.g., $S_{PS}(t, f, t, t) = 0.5 \times 0.9 \times 0.8 \times 0.9 = 0.324 = P(t, f, t, t)$

Let $N_{PS}(x_1 \ldots x_n)$ be the number of samples generated for event $x_1, \ldots, x_n$

Then we have

$$\begin{aligned} \lim_{N\to\infty} \hat{P}(x_1, \ldots, x_n) &= \lim_{N\to\infty} N_{PS}(x_1, \ldots, x_n)/N \\ &= S_{PS}(x_1, \ldots, x_n) \\ &= P(x_1 \ldots x_n) \end{aligned}$$

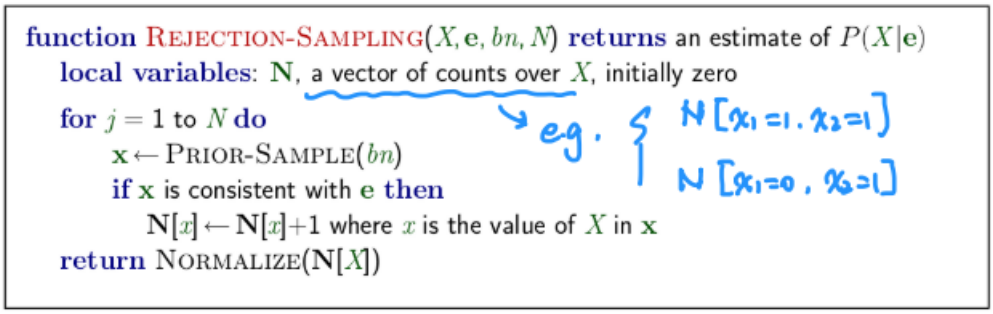
{law of large numbers}

That is, estimates derived from PRIORSAMPLE are consistent

Shorthand: $\hat{P}(x_1, \ldots, x_n) \approx P(x_1 \ldots x_n)$

2.3.2. Rejection Sampling

Ancestral Sampling do not care about observations variables. Rejection Sampling Method is to estimate something more complicated… like P(X|e).



E.g., estimate $\mathbf{P}(Rain|Sprinkler = true)$ using 100 samples
27 samples have $Sprinkler = true$
Of these, 8 have $Rain = true$ and 19 have $Rain = false$.

$\hat{\mathbf{P}}(Rain|Sprinkler = true) = \text{NORMALIZE}(\langle 8, 19\rangle) = \langle 0.296, 0.704\rangle$

The consistency can be proved from law of large numbers:

$$\begin{aligned}
\hat{\mathbf{P}}(X|\mathbf{e}) &= \alpha \mathbf{N}_{PS}(X, \mathbf{e}) && \text{(algorithm defn.)}\\
&= \mathbf{N}_{PS}(X, \mathbf{e})/N_{PS}(\mathbf{e}) && \text{(normalized by } N_{PS}(\mathbf{e})\text{)}\\
&\approx \mathbf{P}(X, \mathbf{e})/P(\mathbf{e}) && \text{(property of PRIORSAMPLE)}\\
&= \mathbf{P}(X|\mathbf{e}) && \text{(defn. of conditional probability)}
\end{aligned}$$

Hence rejection sampling returns consistent posterior estimates

Cons: Hopeless Expensive if P(e) is small, and P(e) does drop off exponentially with number of evidence variables.

2.3.3. Likelihood Weighting

Likelihood Weighting Method only sample points consistent with evidence e by using weight.

function LIKELIHOOD-WEIGHTING($X, \mathbf{e}, bn, N$) **returns** an estimate of $P(X|\mathbf{e})$
  local variables: $\mathbf{W}$, a vector of weighted counts over X, initially zero

  for $j = 1$ to N **do**
    $\mathbf{x}, w \leftarrow$ WEIGHTED-SAMPLE(bn)
    $\mathbf{W}[x] \leftarrow \mathbf{W}[x] + w$ where x is the value of X in $\mathbf{x}$
  return NORMALIZE($\mathbf{W}[X]$)

---

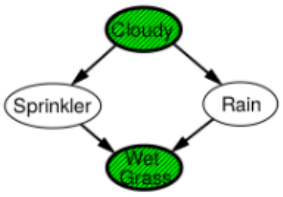
function WEIGHTED-SAMPLE($bn, \mathbf{e}$) **returns** an event and a weight

  $\mathbf{x} \leftarrow$ an event with n elements; $w \leftarrow 1$
  for $i = 1$ **to** n **do**
    if X_i has a value x_i in $\mathbf{e}$
      then $w \leftarrow w \times P(X_i = x_i \mid parents(X_i))$
      else $x_i \leftarrow$ a random sample from $\mathbf{P}(X_i \mid parents(X_i))$
  return $\mathbf{x}$, w

The consistency can be proved from law of large numbers:

- Can again show this is consistent:

  Sampling probability for WEIGHTEDSAMPLE is
  $S_{WS}(\mathbf{z}, \mathbf{e}) = \prod_{i=1}^{l} P(z_i|parents(Z_i))$
  Note: pays attention to evidence in **ancestors** only
  $\Rightarrow$ somewhere "in between" prior and posterior distribution



  Weight for a given sample $\mathbf{z}, \mathbf{e}$ is
  $w(\mathbf{z}, \mathbf{e}) = \prod_{i=1}^{m} P(e_i|parents(E_i))$

  Weighted sampling probability is
  $S_{WS}(\mathbf{z}, \mathbf{e}) w(\mathbf{z}, \mathbf{e})$
  $= \prod_{i=1}^{l} P(z_i|parents(Z_i)) \prod_{i=1}^{m} P(e_i|parents(E_i))$
  $= P(\mathbf{z}, \mathbf{e})$ (by standard global semantics of network)

cons:

1. So, this algorithms may suffer degradation in performance as the number of evidence variables increases. This is because **most samples will have very low weights** and hence the weighted estimate will **be dominated by the tiny fraction of samples** that accord more than an infinitesimal likelihood to the evidence

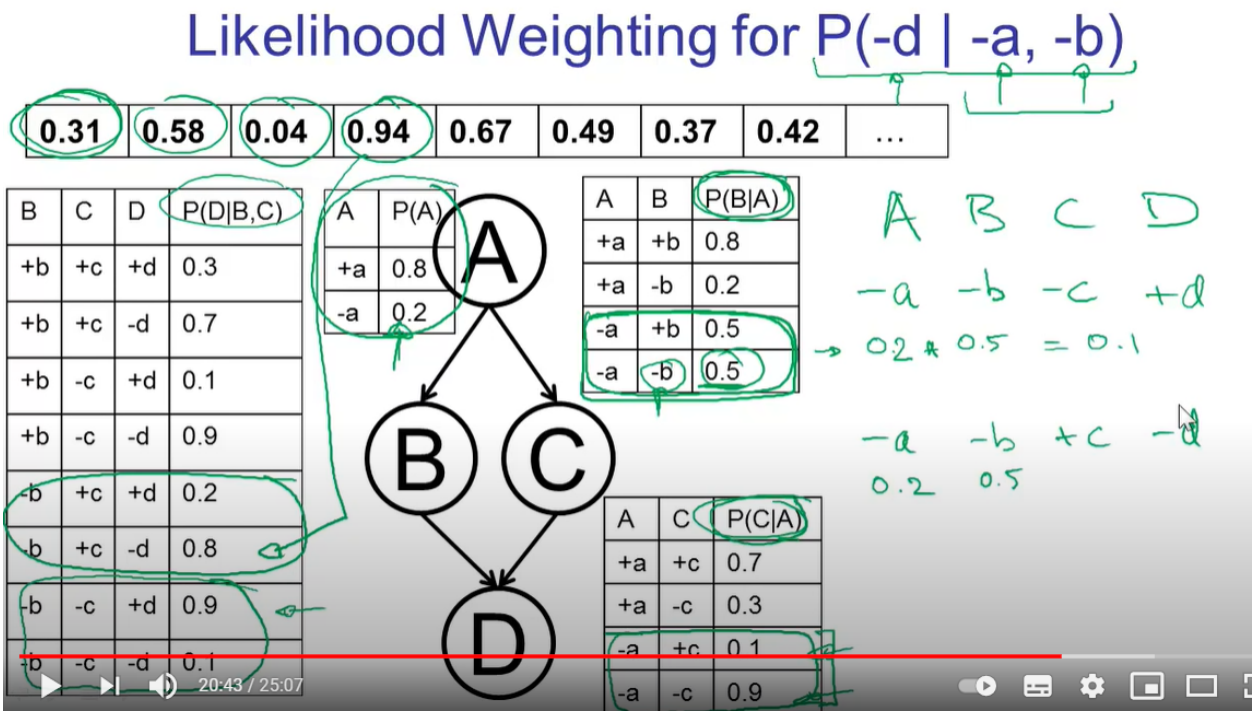
2. The evidence based on which we sample **influences the downstream random variables** that we sample and not upstream. This could bias the samples that are generated.
3. If the evidence variables occur late in the variable ordering, the nonevidence variables will have no evidence in their parents and ancestors to guide the generation of samples.

Intuition:

Based on the conditionally independent in the Bayesian Network and limitedly taken known information (observation) into consideration

Example

https://www.youtube.com/watch?v=ol0l6aTfb_g



2.4. Markov chain Monte Carlo (MCMC)

Waiting for updating