GHENT UNIVERSITY

FACULTY OF ENGINEERING AND ARCHITECTURE

E016350 - Artificial Intelligence

Lecture 10

Reasoning under Uncertainty & Bayesian ML

Inference in Bayesian networks

Aleksandra Pizurica

Ghent University

Spring 2024

Overview

- What is probabilistic inference?
- Exact inference by enumeration
- Exact inference by variable elimination
- Belief propagation
- Approximate inference by stochastic simulation

[R&N], Chapter 13 (Sec 13.3; 13.4)

This presentation is partly based on: S. Russel and P. Norvig: *Artificial Intelligence: A Modern Approach*, Fourth Ed.), denoted as [R&N] and the resource page http://aima.cs.berkeley.edu/

Inference tasks

Denote
$\mathbf{X} = \{X_1, ..., X_n\}$ – the complete set of variables
X – the query variable
$\mathbf{E} = \{E_1, ..., E_n\}$ – the set of evidence variables
$\mathbf{e} = \{e_1, ..., e_n\}$ – an observed event (assignment to evidence variables)
$\mathbf{Y} = \{Y_1, ..., Y_n\}$ – the non-evidence, non-query variables, called hidden variables,
so that $\mathbf{X} = \{X\} \cup \mathbf{E} \cup \mathbf{Y}$

A typical query asks for the posterior probability distribution $\mathbf{P}(X|\mathbf{e})$

This is an example of a simple inference task. $\mathbf{P}(X|\mathbf{e})$ is called posterior marginal (because it is posterior distribution of a subset of variables, in this particular case this subset is only one variable X).

Inference tasks, contd.

- Simple queries: compute posterior marginal $\mathbf{P}(X_i|\mathbf{E}=\mathbf{e})$
  e.g., $P(NoGas|Gauge=empty, Lights=on, Starts=false)$
- Conjunctive queries: $\mathbf{P}(X_i, X_j|\mathbf{E}=\mathbf{e}) = \mathbf{P}(X_i|\mathbf{E}=\mathbf{e})\mathbf{P}(X_j|X_i, \mathbf{E}=\mathbf{e})$
- Optimal decisions: decision networks include utility information;
  probabilistic inference required for $P(outcome|action, evidence)$
- Value of information: which evidence to seek next?
- Sensitivity analysis: which probability values are most critical?
- Explanation: why do I need a new starter motor?

Inference by enumeration: Reminder 'Dentist' example

Consider the query: $\mathbf{P}(Cavity|toothache)$

	toothache		$\neg$ *toothache*	
	catch	$\neg$ *catch*	*catch*	$\neg$ *catch*
cavity	.108	.012	.072	.008
$\neg$ *cavity*	.016	.064	.144	.576

Denominator can be viewed as a normalization constant α

$$
\begin{aligned}
\mathbf{P}(Cavity|toothache) &= \alpha\, \mathbf{P}(Cavity, toothache) \\
&= \alpha\, [\mathbf{P}(Cavity, toothache, catch) + \mathbf{P}(Cavity, toothache, \neg catch)] \\
&= \alpha\, [\langle 0.108, 0.016 \rangle + \langle 0.012, 0.064 \rangle] \\
&= \alpha\, \langle 0.12, 0.08 \rangle = \langle 0.6, 0.4 \rangle
\end{aligned}
$$

Inference by enumeration

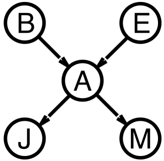
Conditional probabilities can be computed by summing terms from the joint distribution: $\mathbf{P}(X|\mathbf{e}) = \alpha \mathbf{P}(X, \mathbf{e}) = \alpha \sum_y \mathbf{P}(X, \mathbf{e}, \mathbf{y})$

Inference by enumeration

Conditional probabilities can be computed by summing terms from the joint distribution: $\mathbf{P}(X|\mathbf{e}) = \alpha\mathbf{P}(X,\mathbf{e}) = \alpha\sum_y \mathbf{P}(X,\mathbf{e},\mathbf{y})$

Example: simple query on the burglary network

$$\begin{aligned}\mathbf{P}(B|j,m) &= \mathbf{P}(B,j,m)/P(j,m)\\ &= \alpha\mathbf{P}(B,j,m)\\ &= \alpha\sum_e\sum_a \mathbf{P}(B,e,a,j,m)\end{aligned}$$

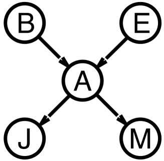


Inference by enumeration

Conditional probabilities can be computed by summing terms from the joint distribution: $\mathbf{P}(X|\mathbf{e}) = \alpha \mathbf{P}(X, \mathbf{e}) = \alpha \sum_y \mathbf{P}(X, \mathbf{e}, \mathbf{y})$

Example: simple query on the burglary network

$$\begin{aligned}\mathbf{P}(B|j,m) &= \mathbf{P}(B,j,m)/P(j,m) \\ &= \alpha \mathbf{P}(B,j,m) \\ &= \alpha \sum_e \sum_a \mathbf{P}(B,e,a,j,m)\end{aligned}$$



Rewrite using the actual network structure and its CPT entries:

$$\begin{aligned}&\mathbf{P}(B|j,m) \\ &= \alpha \sum_e \sum_a \mathbf{P}(B)P(e)\mathbf{P}(a|B,e)P(j|a)P(m|a) \\ &= \alpha \mathbf{P}(B) \sum_e P(e) \sum_a \mathbf{P}(a|B,e)P(j|a)P(m|a)\end{aligned}$$

Recursive depth-first enumeration: $O(n)$ space, $O(d^n)$ time

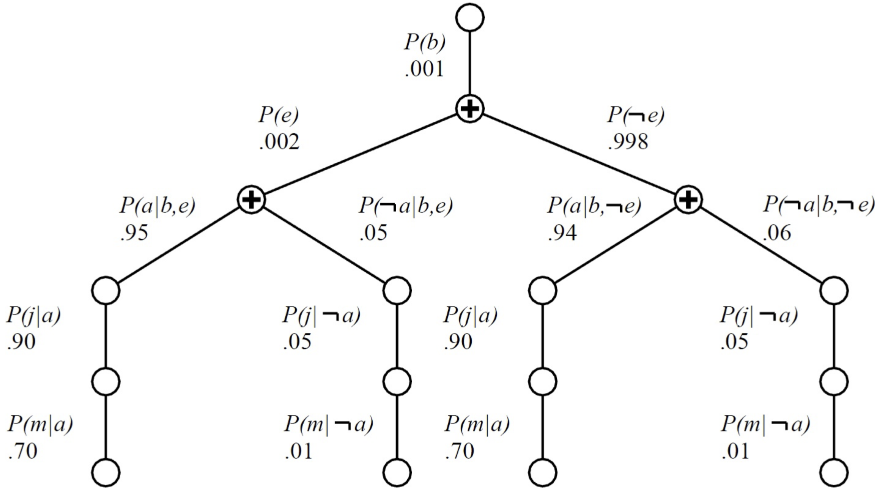
Inference by enumeration

```
function ENUMERATION-ASK(X, e, bn) returns a distribution over X
   inputs: X, the query variable
           e, observed values for variables E
           bn, a Bayesian network with variables {X} ∪ E ∪ Y

   Q(X) ← a distribution over X, initially empty
   for each value x_i of X do
      extend e with value x_i for X
      Q(x_i) ← ENUMERATE-ALL(VARS[bn], e)
   return NORMALIZE(Q(X))
```

```
function ENUMERATE-ALL(vars, e) returns a real number
   if EMPTY?(vars) then return 1.0
   Y ← FIRST(vars)
   if Y has value y in e
      then return P(y | Pa(Y)) × ENUMERATE-ALL(REST(vars), e)
      else return Σ_y P(y | Pa(Y)) × ENUMERATE-ALL(REST(vars), e_y)
           where e_y is e extended with Y = y
```

Inference by enumeration



Inefficient: repeated computations, e.g., computes $P(j|a)P(m|a)$ for each value of e.

Inference by variable elimination

Idea: eliminate repeated calculations carry out summations right-to-left (bottom-up) storing intermediate results for later use

$$\mathbf{P}(B|j,m) = \alpha \underbrace{\mathbf{P}(B)}_{\mathbf{f}_1(B)} \sum_e \underbrace{P(e)}_{\mathbf{f}_2(E)} \sum_a \underbrace{\mathbf{P}(a|B,e)}_{\mathbf{f}_3(A,B,E)} \underbrace{P(j|a)}_{\mathbf{f}_4(A)} \underbrace{P(m|a)}_{\mathbf{f}_5(A)}$$

Here the factors are vectors like

$$\mathbf{f}_1(B) = \begin{bmatrix} P(b) \\ P(\neg b) \end{bmatrix} ; \mathbf{f}_4(A) = \begin{bmatrix} P(j|a) \\ P(j|\neg a) \end{bmatrix} \text{ etc.}$$

so, we have

$$\mathbf{P}(B|j,m) = \alpha \mathbf{f}_1(B) \sum_e \mathbf{f}_2(E) \sum_a \mathbf{f}_3(A,B,E) \times \mathbf{f}_4(A) \times \mathbf{f}_5(A)$$

where $\times$ is pointwise product

Inference by variable elimination, contd.

Now, compute from right to left

$$\mathbf{P}(B|j,m) = \alpha \mathbf{f}_1(B) \sum_e \mathbf{f}_2(E) \underbrace{\sum_a \mathbf{f}_3(A,B,E) \times \ \mathbf{f}_4(A) \times \mathbf{f}_5(A)}$$

Inference by variable elimination, contd.

Now, compute from right to left

$$\mathbf{P}(B|j,m) = \alpha \mathbf{f}_1(B) \sum_e \mathbf{f}_2(E) \underbrace{\sum_a \mathbf{f}_3(A,B,E) \times \ \mathbf{f}_4(A) \times \mathbf{f}_5(A)}_{\mathbf{f}_6(B,E)}$$

$$\mathbf{f}_6(B,E) = \sum_a \mathbf{f}_3(A,B,E) \times \ \mathbf{f}_4(A) \times \mathbf{f}_5(A) =$$
$$= (\mathbf{f}_3(a,B,E) \times \mathbf{f}_4(a) \times \mathbf{f}_5(a)) + (\mathbf{f}_3(\neg a,B,E) \times \mathbf{f}_4(\neg a) \times \mathbf{f}_5(\neg a))$$

$$\mathbf{P}(B|j,m) = \alpha \mathbf{f}_1(B) \underbrace{\sum_e \mathbf{f}_2(E) \times \mathbf{f}_6(B,E)}$$

Inference by variable elimination, contd.

Now, compute from right to left

$$\mathbf{P}(B|j,m) = \alpha \mathbf{f}_1(B) \sum_e \mathbf{f}_2(E) \underbrace{\sum_a \mathbf{f}_3(A,B,E) \times \ \mathbf{f}_4(A) \times \mathbf{f}_5(A)}_{\mathbf{f}_6(B,E)}$$

$$\mathbf{f}_6(B,E) = \sum_a \mathbf{f}_3(A,B,E) \times \ \mathbf{f}_4(A) \times \mathbf{f}_5(A) =$$
$$= (\mathbf{f}_3(a,B,E) \times \mathbf{f}_4(a) \times \mathbf{f}_5(a)) + (\mathbf{f}_3(\neg a,B,E) \times \mathbf{f}_4(\neg a) \times \mathbf{f}_5(\neg a))$$

$$\mathbf{P}(B|j,m) = \alpha \mathbf{f}_1(B) \underbrace{\sum_e \mathbf{f}_2(E) \times \mathbf{f}_6(B,E)}_{\mathbf{f}_7(B)}$$

$$\mathbf{P}(B|j,m) = \alpha \mathbf{f}_1(B) \mathbf{f}_7(B)$$

Variable elimination: Basic operations

Pointwise product of factors $\mathbf{f}_1$ and $\mathbf{f}_2$:

$$\mathbf{f}_1(x_1, \ldots, x_j, y_1, \ldots, y_k) \times \mathbf{f}_2(y_1, \ldots, y_k, z_1, \ldots, z_l)$$
$$= \mathbf{f}(x_1, \ldots, x_j, y_1, \ldots, y_k, z_1, \ldots, z_l)$$

E.g., $\mathbf{f}_1(a, b) \times \mathbf{f}_2(b, c) = \mathbf{f}(a, b, c)$

Summing out a variable from a product of factors:

move any constant factors outside the summation

add up submatrices in pointwise product of remaining factors:

$$\sum_x \mathbf{f}_1 \times \cdots \times \mathbf{f}_k = \mathbf{f}_1 \times \cdots \times \mathbf{f}_i \underbrace{\sum_x \mathbf{f}_{i+1} \times \cdots \times \mathbf{f}_k}_{\mathbf{f}_{\bar{X}}} = \mathbf{f}_1 \times \cdots \times \mathbf{f}_i \times \mathbf{f}_{\bar{X}}$$

assuming $\mathbf{f}_1, \ldots, \mathbf{f}_i$ do not depend on X.

Example: pointwise product of factors

The pointwise product of two factors $\mathbf{f}_1$ and $\mathbf{f}_2$ yields a new factor whose variables are the union of the variables in $\mathbf{f}_1$ and $\mathbf{f}_2$ and whose elements are given by the product of the corresponding elements in the two factors.

A	B	$\mathbf{f}_1(A,B)$	B	C	$\mathbf{f}_2(B,C)$	A	B	C	$\mathbf{f}_3(A,B,C)$
T	T	.3	T	T	.2	T	T	T	$.3 \times .2 = .06$
T	F	.7	T	F	.8	T	T	F	$.3 \times .8 = .24$
F	T	.9	F	T	.6	T	F	T	$.7 \times .6 = .42$
F	F	.1	F	F	.4	T	F	F	$.7 \times .4 = .28$
						F	T	T	$.9 \times .2 = .18$
						F	T	F	$.9 \times .8 = .72$
						F	F	T	$.1 \times .6 = .06$
						F	F	F	$.1 \times .4 = .04$

Figure 14.10 Illustrating pointwise multiplication: $\mathbf{f}_1(A,B) \times \mathbf{f}_2(B,C) = \mathbf{f}_3(A,B,C)$.

Example: summing out a variable from a product of factors

A	B	$\mathbf{f}_1(A,B)$	B	C	$\mathbf{f}_2(B,C)$	A	B	C	$\mathbf{f}_3(A,B,C)$
T	T	.3	T	T	.2	T	T	T	$.3 \times .2 = .06$
T	F	.7	T	F	.8	T	T	F	$.3 \times .8 = .24$
F	T	.9	F	T	.6	T	F	T	$.7 \times .6 = .42$
F	F	.1	F	F	.4	T	F	F	$.7 \times .4 = .28$
						F	T	T	$.9 \times .2 = .18$
						F	T	F	$.9 \times .8 = .72$
						F	F	T	$.1 \times .6 = .06$
						F	F	F	$.1 \times .4 = .04$

Figure 14.10 Illustrating pointwise multiplication: $\mathbf{f}_1(A,B) \times \mathbf{f}_2(B,C) = \mathbf{f}_3(A,B,C)$.

$$\mathbf{f}(B,C) = \sum_a \mathbf{f}_3(A,B,C) = \mathbf{f}_3(a,B,C) + \mathbf{f}_3(\neg a,B,C)$$

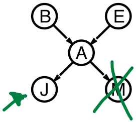
$$= \begin{bmatrix} 0.06 & 0.24 \\ 0.42 & 0.28 \end{bmatrix} + \begin{bmatrix} 0.18 & 0.72 \\ 0.06 & 0.04 \end{bmatrix} = \begin{bmatrix} 0.24 & 0.96 \\ 0.48 & 0.32 \end{bmatrix}$$

Variable elimination algorithm

```
function ELIMINATION-ASK(X, e, bn) returns a distribution over X
   inputs: X, the query variable
           e, evidence specified as an event
           bn, a belief network specifying joint distribution P(X_1, ..., X_n)
   factors ← []; vars ⟵ ORDER (VARS[bn])
   for each var in vars do
       factors ← [MAKE-FACTOR(var, e)|factors]
       if var is a hidden variable then factors ← SUM-OUT(var, factors)
   return NORMALIZE(POINTWISE-PRODUCT(factors))
```

Think of variable ordering.

Variable relevance - Example



$$\mathbf{P}(J|b) = \alpha \sum_e \sum_a \sum_m P(J,b,e,a,m)$$

$$= \alpha \sum_e \sum_a \sum_m P(b)P(e)P(a|b,e)P(J|a)P(m|a)$$

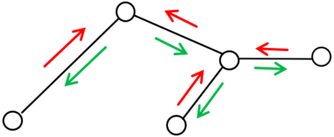
$$= \alpha P(b) \sum_e P(e) \sum_a P(a|b,e)P(J|a) \underbrace{\sum_m P(m|a)}_{1}$$

So, there was no need to include m!

Every variable that is not an ancestor of a query variable or evidence variable is irrelevant to the query!

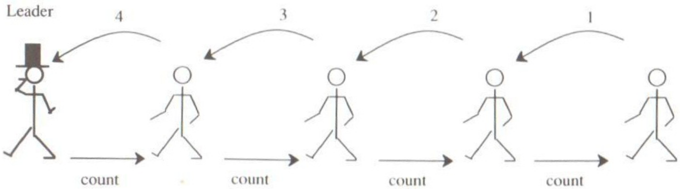
Belief propagation

- Belief propagation algorithm was introduced by Judea Pearl, 1982
- Exact inference in networks without loops; time complexity linear in the number of nodes
- Became very popular after it was shown that the same computations are in turbo codes and the same principles in the Viterbi algorithm
- Main idea: inference by local **message passing** among neighboring nodes;
  The message can loosely be interpreted as "I (node i) think that you (node j) are that much likely to be in a given state".

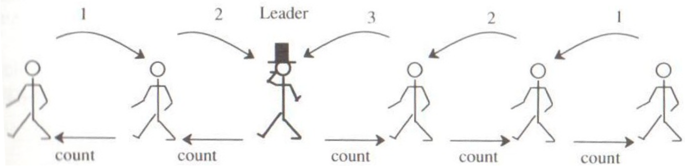


Message passing revisited

Distributed soldier counting:



Distributed soldier counting with leader in line:



Do we need the leader for this process? Think of leaderless soldier counting.

Belief propagation

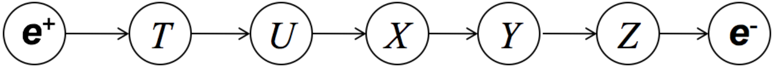
Problem: express the probability of X given the set of old evidences $\mathbf{e}_n = \{e_1 \ldots e_n\}$ and a new piece of evidence e_{n+1}

Belief propagation

Problem: express the probability of X given the set of old evidences $\mathbf{e}_n = \{e_1 \dots e_n\}$ and a new piece of evidence e_{n+1}

$$
\begin{aligned}
P(x|\mathbf{e}_n, e_{n+1}) &= \frac{P(x, \mathbf{e}_n, e_{n+1})}{P(\mathbf{e}_n, e_{n+1})} = \frac{P(e_{n+1}|x, \mathbf{e}_n)P(x, \mathbf{e}_n)}{P(\mathbf{e}_n, e_{n+1})} \\
&= \frac{P(e_{n+1}|x, \mathbf{e}_n)P(x|\mathbf{e}_n)\cancel{P(\mathbf{e}_n)}}{P(e_{n+1}|\mathbf{e}_n)\cancel{P(\mathbf{e}_n)}} \\
&= \underbrace{P(e_{n+1}|\mathbf{e}_n)^{-1}}_{\alpha} P(e_{n+1}|x, \mathbf{e}_n)P(x|\mathbf{e}_n)
\end{aligned}
$$

Belief propagation in chains

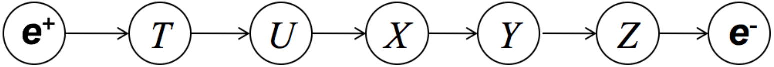


$$P(x|\mathbf{e}^+,\mathbf{e}^-) = \frac{P(x,\mathbf{e}^+,\mathbf{e}^-)}{P(\mathbf{e}^+,\mathbf{e}^-)} = \frac{\overbrace{P(\mathbf{e}^-|x,\mathbf{e}^+)}^{P(\mathbf{e}^-|x)}\,P(x,\mathbf{e}^+)}{P(\mathbf{e}^+,\mathbf{e}^-)} = \frac{P(\mathbf{e}^-|x)P(x|\mathbf{e}^+)P(\mathbf{e}^+)}{P(\mathbf{e}^+,\mathbf{e}^-)}$$

$$= \frac{P(\mathbf{e}^-|x)P(x|\mathbf{e}^+)\cancel{P(\mathbf{e}^+)}}{P(\mathbf{e}^-|\mathbf{e}^+)\cancel{P(\mathbf{e}^+)}} = \underbrace{P(\mathbf{e}^-|\mathbf{e}^+)^{-1}}_{\alpha}\,\underbrace{P(\mathbf{e}^-|x)}_{\lambda(x)}\,\underbrace{P(x|\mathbf{e}^+)}_{\pi(x)}$$

Note: we assumed here that all available evidence $\mathbf{E}$ is split into $\mathbf{E}^+$ and $\mathbf{E}^-$, i.e., $\mathbf{E} = \mathbf{E}^+ \cup \mathbf{E}^-$, and $\mathbf{e}^+$ and $\mathbf{e}^-$ are assignments to $\mathbf{E}^+$ and $\mathbf{E}^-$, respectively.

Belief propagation in chains, contd.



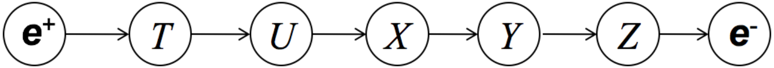
$$P(x|\mathbf{e}^+, \mathbf{e}^-) = \alpha \lambda(x) \pi(x)$$

$$\lambda(x) = P(\mathbf{e}^-|x), \; \pi(x) = P(x|\mathbf{e}^+)$$

Notice how $\pi(x)$ propagates down the chain:

$$\pi(x) = P(x|\mathbf{e}^+) = \sum_u \underbrace{P(x|u, \mathbf{e}^+)}_{P(x|u)} \underbrace{P(u|\mathbf{e}^+)}_{\pi(u)} = \sum_u P(x|u)\pi(u)$$

Belief propagation in chains, contd.



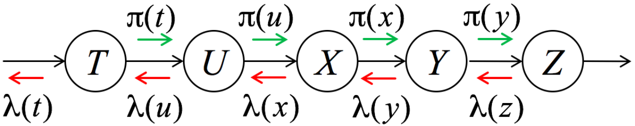
$$P(x|\mathbf{e}^+,\mathbf{e}^-) = \alpha\lambda(x)\pi(x)$$

$$\lambda(x) = P(\mathbf{e}^-|x),\ \pi(x) = P(x|\mathbf{e}^+)$$

Similarly, $\lambda(x)$ propagates in the other direction:

$$\lambda(x) = P(\mathbf{e}^-|x) = \sum_y \underbrace{P(\mathbf{e}^-|y,x)}_{P(\mathbf{e}^-|y)=\lambda(y)} P(y|x) = \sum_y \lambda(y)P(y|x)$$

Belief propagation in chains, contd.



$$BEL(x) = P(x|\mathbf{e}) = P(x|\mathbf{e}^+, \mathbf{e}^-) = \alpha\lambda(x)\pi(x)$$

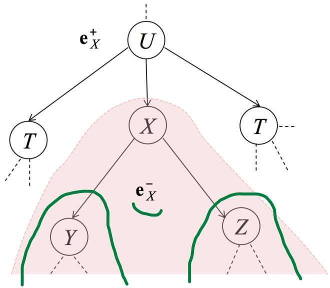
$$\lambda(x) = P(\mathbf{e}^-|x),\ \pi(x) = P(x|\mathbf{e}^+)$$

$BEL(x)$ – belief accorded to proposition $X = x$ by all evidence $\mathbf{e}$ so far received.

$\pi(x)$ – causal or predictive support attributed to the assertion $X = x$ by all non-descendants of X, mediated by X's parent.

$\lambda(x)$ – diagnostic or retrospective support that $X = x$ receives from X's descendents.

Belief propagation in trees

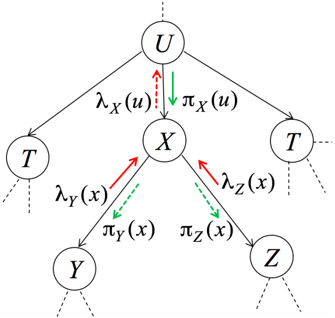


- Let the query be $BEL(x) = P(x|\mathbf{e})$
- Divide $\mathbf{e}$ into $\mathbf{e}_X^-$ and $\mathbf{e}_X^+$. Suppose $\mathbf{e}_X^-$ is in the network rooted at X and $\mathbf{e}_X^+$ is in the rest of the network.
- Like with the chain, we can show $BEL(x) = P(x|\mathbf{e}) = \alpha\lambda(x)\pi(x)$, with $\lambda(x) = P(\mathbf{e}_X^-|x)$, $\pi(x) = P(x|\mathbf{e}_X^+)$; $\alpha = P(\mathbf{e}_X^-|\mathbf{e}_X^+)^{-1}$

$$\lambda(x) = P(\mathbf{e}_X^-|x) = P(\mathbf{e}_Y^-, \mathbf{e}_Z^-|x) = P(\mathbf{e}_Y^-|x)P(\mathbf{e}_Z^-|x) = \underbrace{\lambda_Y(x)\lambda_Z(x)}_{\text{messages from children}}$$

$$\pi(x) = P(x|\mathbf{e}_X^+) = \sum_u P(x|\mathbf{e}_X^+, u)P(u|\mathbf{e}_X^+) = \underbrace{\sum_u P(x|u)\pi_X(u)}_{\text{message from the parent}}$$

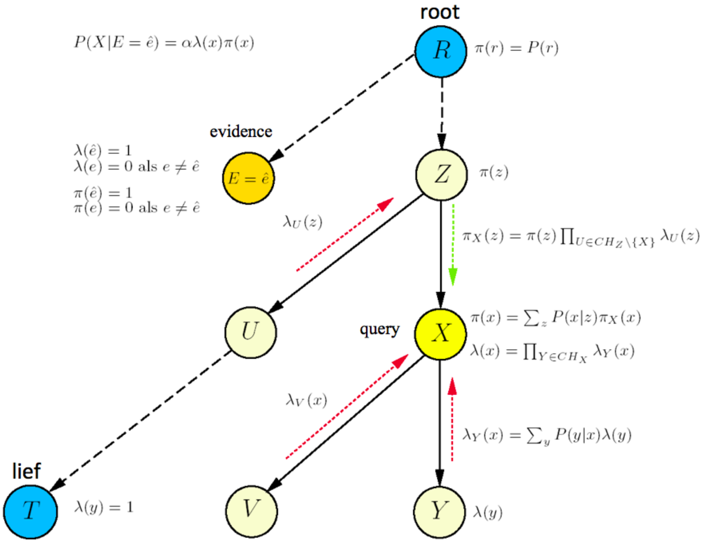
Belief propagation in trees, contd.



- Belief updating
  $BEL(x) = P(x|\mathbf{e}) = \alpha\lambda(x)\pi(x)$
  $\lambda(x) = \prod_j \lambda_{Y_j}(x)$
  $\pi(x) = \sum_u P(x|u)\pi_X(u)$
  α is const. such that $\sum_x BEL(x) = 1$
- Bottom-up propagation
  $\lambda_X(u) = \sum_x \lambda(x)P(x|u)$
- Top-down propagation
  $\pi_{Y_j}(x) = \alpha\pi(x)\prod_{k \neq j} \lambda_{Y_k}(x)$

For more details, see (optional):
Judea Pearl, *Probabilistic reasoning in intelligence Systems: Networks of Plausible Inference*, (2nd Edition, Section 4.2)

Belief propagation in trees, contd.



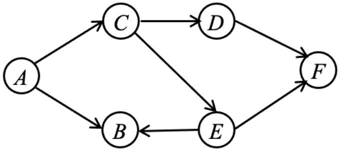
Inference by stochastic simulation

- Basic idea
  - Draw samples from a sampling distribution
  - Compute an approximate posterior probability
  - Show this converges to the true probability
- Different methods from this class:
  - Sampling from an empty network
  - Rejection sampling: reject samples disagreeing with evidence
  - Likelihood weighting: use evidence to weight samples
  - Markov chain Monte Carlo (MCMC): sample from a stochastic process whose stationary distribution is the true posterior
- Applicable to arbitrary network topologies and arbitrary combinations of discrete and continuous r.v.s
- Convergence can be very slow

Network separation

A **simple path** through a graph (or a **simple chain**) is a sequence of vertices and edges where no vertices (and hence no edges) are repeated.

In other words, a simple chain contains no loops.

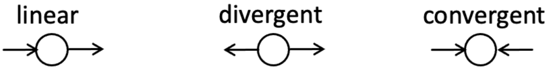


Examples:

$A \rightarrow B \leftarrow E \rightarrow F$

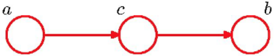
$A \rightarrow C \rightarrow E \rightarrow B$

The internal nodes of a simple path can be classified as:



Network separation, contd.

We investigate (conditional) independence in three simple networks featuring these types of nodes. Let $a \perp\!\!\!\perp b \mid c$ denote "a and b are conditionally independent given c"



$$P(a,b,c) = P(a)P(c \mid a)P(b \mid c)$$

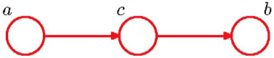
$$P(a,b) = \sum_c P(a)P(c \mid a)P(b \mid c) = P(a)P(b \mid a)$$

$$\neq P(a)P(b) \Rightarrow a \not\perp\!\!\!\perp b \mid \emptyset$$

(in this network a and b are in general **not** independent)

Network separation, contd.

We investigate (conditional) independence in three simple networks featuring these types of nodes. Let $a \perp\!\!\!\perp b \mid c$ denote "a and b are conditionally independent given c"



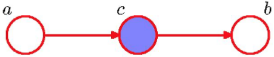
$$P(a,b,c) = P(a)P(c \mid a)P(b \mid c)$$

$$P(a,b) = \sum_c P(a)P(c \mid a)P(b \mid c) = P(a)P(b \mid a)$$

$$\neq P(a)P(b) \Rightarrow a \not\perp\!\!\!\perp b \mid \emptyset$$

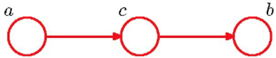
(in this network a and b are in general **not** independent)

Consider now evidence in c:



Network separation, contd.

We investigate (conditional) independence in three simple networks featuring these types of nodes. Let $a \perp\!\!\!\perp b \mid c$ denote "a and b are conditionally independent given c"



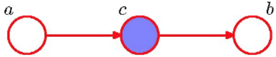
$$P(a,b,c) = P(a)P(c \mid a)P(b \mid c)$$

$$P(a,b) = \sum_c P(a)P(c \mid a)P(b \mid c) = P(a)P(b \mid a)$$

$$\neq P(a)P(b) \Rightarrow a \not\perp\!\!\!\perp b \mid \emptyset$$

(in this network a and b are in general **not** independent)

Consider now evidence in c:



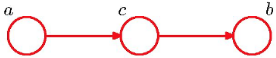
$$P(a,b \mid c) = \frac{P(a,b,c)}{P(c)} = \frac{P(a)P(c \mid a)P(b \mid c)}{P(c)} =$$

$$= P(a \mid c)P(b \mid c)$$

$$\Rightarrow a \perp\!\!\!\perp b \mid c$$

Network separation, contd.

We investigate (conditional) independence in three simple networks featuring these types of nodes. Let $a \perp\!\!\!\perp b \mid c$ denote "a and b are conditionally independent given c"



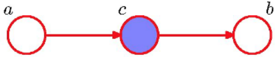
$$P(a,b,c) = P(a)P(c \mid a)P(b \mid c)$$

$$P(a,b) = \sum_c P(a)P(c \mid a)P(b \mid c) = P(a)P(b \mid a)$$

$$\neq P(a)P(b) \Rightarrow a \not\perp\!\!\!\perp b \mid \emptyset$$

(in this network a and b are in general **not** independent)

Consider now evidence in c:



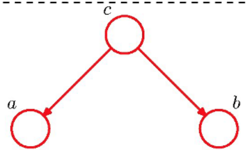
$$P(a,b \mid c) = \frac{P(a,b,c)}{P(c)} = \frac{P(a)P(c \mid a)P(b \mid c)}{P(c)} =$$

$$= P(a \mid c)P(b \mid c)$$

$$\Rightarrow a \perp\!\!\!\perp b \mid c$$

Network separation, contd.

We investigate (conditional) independence in three simple networks featuring these types of nodes. Let $a \perp\!\!\!\perp b \mid c$ denote "a and b are conditionally independent given c"



$$P(a,b,c) = P(a \mid c)P(b \mid c)P(c)$$

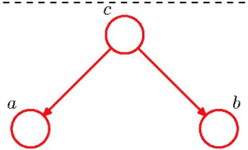
$$P(a,b) = \sum_c P(a \mid c)P(b \mid c)P(c) \neq P(a)P(b)$$

$$\Rightarrow \; a \not\perp\!\!\!\perp b \mid \emptyset$$

(in this network *a* and *b* are in general **not** independent)

Network separation, contd.

We investigate (conditional) independence in three simple networks featuring these types of nodes. Let $a \perp\!\!\!\perp b \mid c$ denote "a and b are conditionally independent given c"



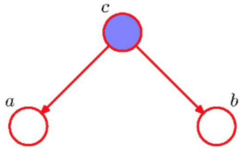
$$P(a,b,c) = P(a \mid c)P(b \mid c)P(c)$$

$$P(a,b) = \sum_c P(a \mid c)P(b \mid c)P(c) \neq P(a)P(b)$$

$$\Rightarrow \; a \not\perp\!\!\!\perp b \mid \emptyset$$

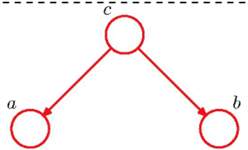
(in this network a and b are in general **not** independent)

Consider now evidence in c:



Network separation, contd.

We investigate (conditional) independence in three simple networks featuring these types of nodes. Let $a \perp\!\!\!\perp b \mid c$ denote "a and b are conditionally independent given c"



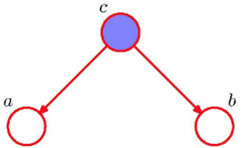
$$P(a,b,c) = P(a \mid c)P(b \mid c)P(c)$$

$$P(a,b) = \sum_c P(a \mid c)P(b \mid c)P(c) \neq P(a)P(b)$$

$$\Rightarrow a \not\perp\!\!\!\perp b \mid \emptyset$$

(in this network a and b are in general **not** independent)

Consider now evidence in c:



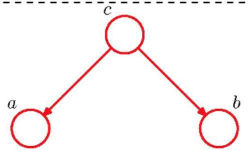
$$P(a,b \mid c) = \frac{P(a,b,c)}{P(c)} = \frac{P(a \mid c)P(b \mid c)P(c)}{P(c)} =$$

$$= P(a \mid c)P(b \mid c)$$

$$\Rightarrow a \perp\!\!\!\perp b \mid c$$

Network separation, contd.

We investigate (conditional) independence in three simple networks featuring these types of nodes. Let $a \perp\!\!\!\perp b \mid c$ denote "a and b are conditionally independent given c"



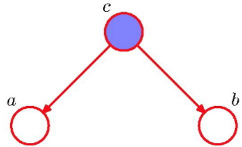
$$P(a,b,c) = P(a \mid c)P(b \mid c)P(c)$$

$$P(a,b) = \sum_c P(a \mid c)P(b \mid c)P(c) \neq P(a)P(b)$$

$$\Rightarrow\ a \not\perp\!\!\!\perp b \mid \emptyset$$

(in this network a and b are in general **not** independent)

Consider now evidence in c:



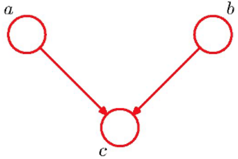
$$P(a,b \mid c) = \frac{P(a,b,c)}{P(c)} = \frac{P(a \mid c)P(b \mid c)P(c)}{P(c)} =$$

$$= P(a \mid c)P(b \mid c)$$

$$\Rightarrow\ a \perp\!\!\!\perp b \mid c$$

Network separation, contd.

We investigate (conditional) independence in three simple networks featuring these types of nodes. Let $a \perp\!\!\!\perp b \mid c$ denote "a and b are conditionally independent given c"



$$P(a,b,c) = P(a)P(b)P(c \mid a,b)$$

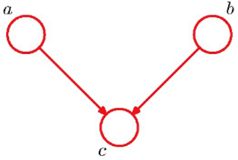
$$P(a,b) = P(a)P(b)\underbrace{\sum_c P(c \mid a,b)}_{1} = P(a)P(b)$$

$$\Rightarrow a \perp\!\!\!\perp b \mid \emptyset$$

(in this network *a* and *b* are independent)

Network separation, contd.

We investigate (conditional) independence in three simple networks featuring these types of nodes. Let $a \perp\!\!\!\perp b \mid c$ denote "a and b are conditionally independent given c"



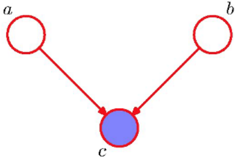
$$P(a,b,c) = P(a)P(b)P(c \mid a,b)$$

$$P(a,b) = P(a)P(b)\underbrace{\sum_c P(c \mid a,b)}_{1} = P(a)P(b)$$

$$\Rightarrow a \perp\!\!\!\perp b \mid \emptyset$$

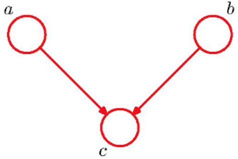
(in this network *a* and *b* are independent)

Consider now evidence in *c*:



Network separation, contd.

We investigate (conditional) independence in three simple networks featuring these types of nodes. Let $a \perp\!\!\!\perp b \mid c$ denote "a and b are conditionally independent given c"



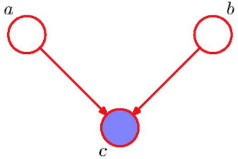
$$P(a,b,c) = P(a)P(b)P(c \mid a,b)$$

$$P(a,b) = P(a)P(b)\underbrace{\sum_c P(c \mid a,b)}_{1} = P(a)P(b)$$

$$\Rightarrow a \perp\!\!\!\perp b \mid \emptyset$$

(in this network *a* and *b* are independent)

Consider now evidence in *c*:



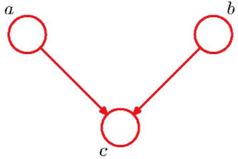
$$P(a,b \mid c) = \frac{P(a,b,c)}{P(c)} = \frac{P(a)P(b)P(c \mid a,b)}{P(c)} =$$

$$\neq P(a \mid c)P(b \mid c)$$

$$\Rightarrow a \not\perp\!\!\!\perp b \mid c$$

Network separation, contd.

We investigate (conditional) independence in three simple networks featuring these types of nodes. Let $a \perp\!\!\!\perp b \mid c$ denote "a and b are conditionally independent given c"



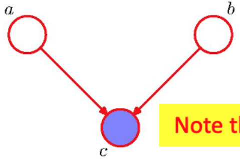
$$P(a,b,c) = P(a)P(b)P(c \mid a,b)$$

$$P(a,b) = P(a)P(b)\underbrace{\sum_c P(c \mid a,b)}_{1} = P(a)P(b)$$

$$\Rightarrow a \perp\!\!\!\perp b \mid \emptyset$$

(in this network a and b are independent)

Consider now evidence in c:



$$P(a,b \mid c) = \frac{P(a,b,c)}{P(c)} = \frac{P(a)P(b)P(c \mid a,b)}{P(c)} =$$

$$\neq P(a \mid c)P(b \mid c)$$

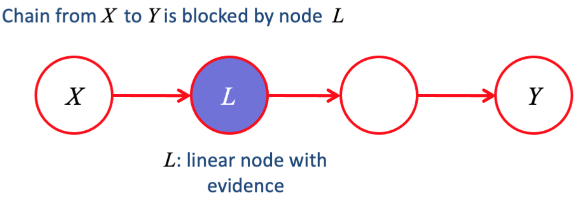
$$\Rightarrow a \not\!\perp\!\!\!\perp b \mid c$$

Note the opposite behavior w.r.t linear and divergent cases!

Using network separation

- Using these properties of the three types of internal nodes in a chain, we can see which parts of the network can be "separated" from the rest.
- We also say that the chain is blocked by the corresponding node.

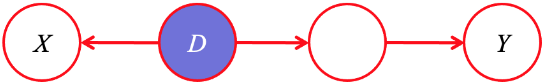
Examples:



Using network separation

- Using these properties of the three types of internal nodes in a chain, we can see which parts of the network can be "separated" from the rest.
- We also say that the chain is blocked by the corresponding node.

Examples:



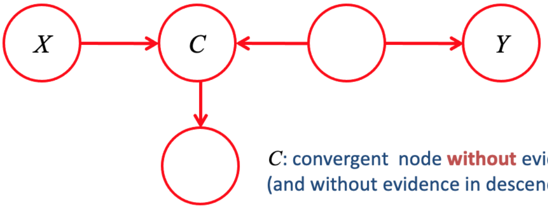
D: divergent node
with evidence

Using network separation

- Using these properties of the three types of internal nodes in a chain, we can see which parts of the network can be "separated" from the rest.
- We also say that the chain is blocked by the corresponding node.

Examples:

Chain from X to Y is blocked by node C



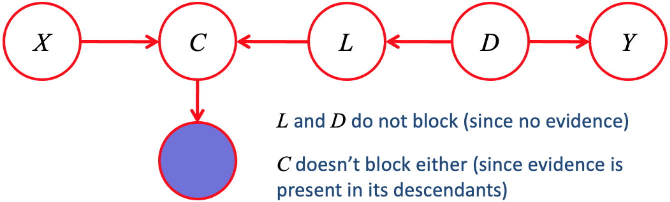
C: convergent node **without** evidence
(and without evidence in descendants)

Using network separation

- Using these properties of the three types of internal nodes in a chain, we can see which parts of the network can be "separated" from the rest.
- We also say that the chain is blocked by the corresponding node.

Examples:

Chain from X to Y is **NOT** blocked.

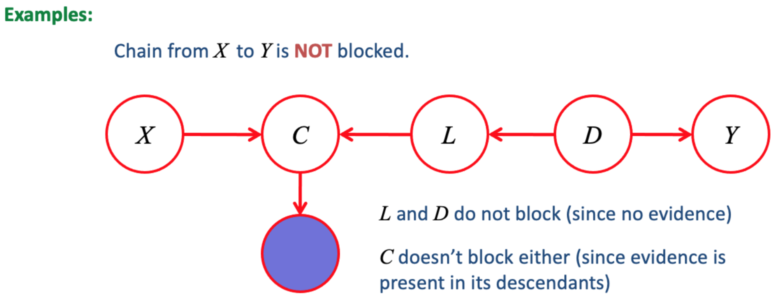


L and D do not block (since no evidence)

C doesn't block either (since evidence is present in its descendants)

Using network separation

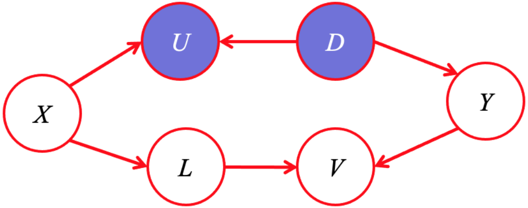
- Using these properties of the three types of internal nodes in a chain, we can see which parts of the network can be "separated" from the rest.
- We also say that the chain is blocked by the corresponding node.



Using network separation

Let's see the cases where nodes are connected with multiple chains

The nodes X and Y are **d-separated** by evidence $A = \{U, D\}$.

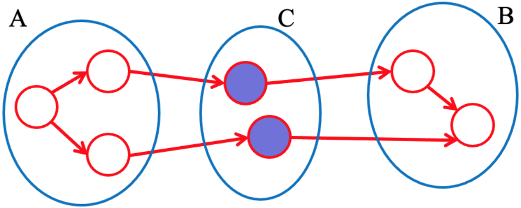


All the chains between *X* and *Y* are blocked
- Chain [*X*, *U*, *D*, *Y*] is blocked by the divergent node *D*
- Chain [*X*, *L*, *V*, *Y*] is blocked by the convergent node *V*

Using network separation

We use the same principle in larger networks

A, B and C are non-overlapping sets

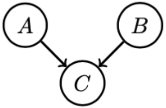


The sets A and B are d-separated by C if
each node in A is d-separated from each node in B by C

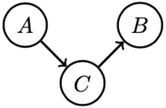
We denote this by: $A \perp\!\!\!\perp B \mid C$

Collider

Here is an easy way to remember the (conditional) independence structure.
A **collider** contains two or more incoming arrows along a chosen path.



If C has more than one incoming link, then $A \perp\!\!\!\perp B$ and $A \not\perp\!\!\!\perp B \,|\, C$. In this case C is called collider.



If C has at most one incoming link, then $A \perp\!\!\!\perp B \,|\, C$ and $A \not\perp\!\!\!\perp B$. In this case C is called non-collider.

Summary

- Probabilistic inference computes the posterior probability distribution for a set of query variables, given some observed event (i.e., some assignment of values to a set of evidence variables)
- Exact inference
  - Inference by enumeration is conceptually simple, but inefficient
  - Variable elimination – smarter approach - avoids recalculations
  - Belief propagation (exact on networks without loops)
- Approximate inference
  - Stochastic simulation
  - Can handle arbitrary combinations of discrete and continuous r.v.s
  - Convergence can be very slow
  - We will learn about this later in the context of general graphical models
- Use the network separation where possible!