Artificial Intelligence

INFERENCE IN BAYESIAN NETWORKS

Nguyễn Ngọc Thảo – Nguyễn Hải Minh
{nnthao, nhminh}@fit.hcmus.edu.vn

Acknowledgements

- This slide is mainly based on the textbook AIMA (3rd edition)
- Some parts of the slide are adapted from
  - Cristina Conati, *Lecture 20: Bayesian Networks: Construction,* Computer Science CPSC 322, University of British Columbia*.*

Outline

- Representing knowledge in an uncertain domain
- Exact inference in Bayesian networks
- Constructing a Bayesian network

Knowledge in uncertain domain

Full joint probability distribution

- The probabilities of all possible worlds can be described by using full joint probability distribution (FJPD).
  - The elements are indexed by values of random variables.
  - We can calculate probabilities of values of any random variable.

	toothache		*¬ toothache*	
	catch	*¬ catch*	*catch*	*¬ catch*
cavity	.108	.012	.072	.008
¬ cavity	.016	.064	.144	.576

$$\mathbf{P}(Toothache) = \langle 0.2, 0.8 \rangle$$

$$P(\neg cavity \mid toothache) = \frac{P(\neg cavity \wedge toothache)}{P(toothache)} = \frac{0.016 + 0.064}{0.108 + 0.012 + 0.016 + 0.064}$$

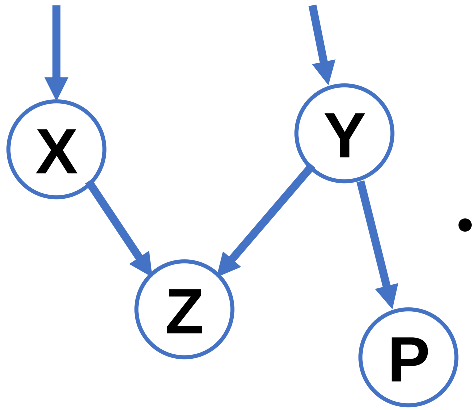
$$P(cavity \vee toothache) = 0.108 + 0.012 + 0.072 + 0.008 + 0.016 + 0.064$$

Full joint probability distribution

- FJPD answers any question about the domain.
- Plain FJPD becomes intractably large as the number of variables grows.
  - Defining probabilities for possible worlds is unnatural and tedious.
- (Conditional) independence relationships among variables can greatly reduce the number of probabilities required.

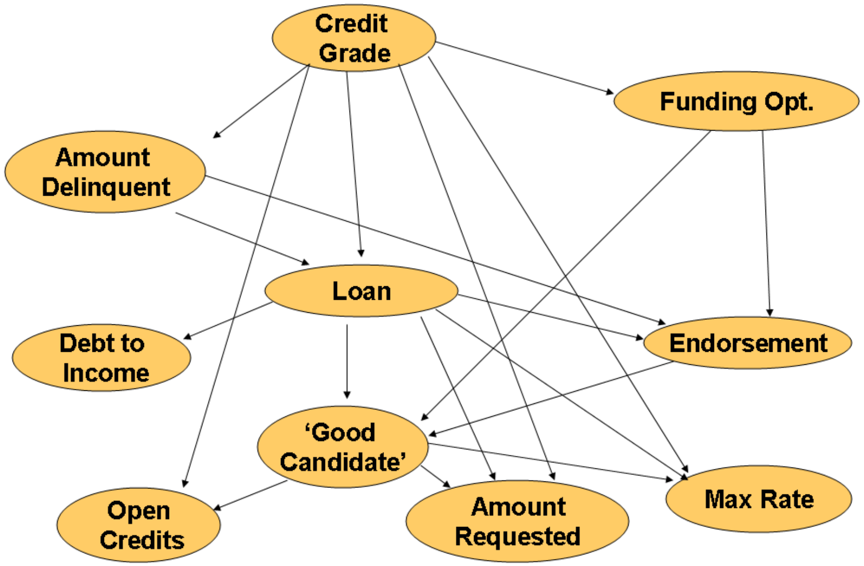
Bayesian networks

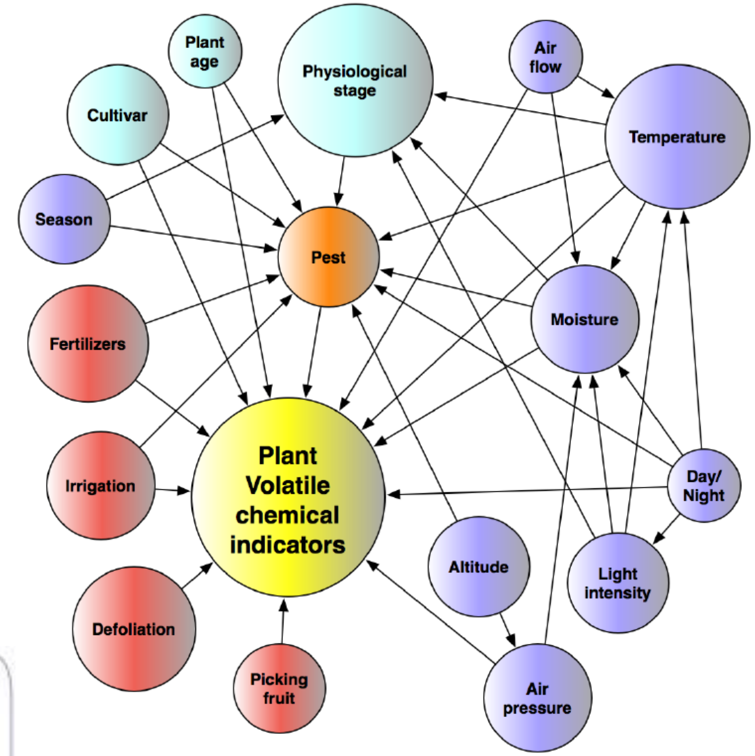
- Bayesian networks (BN) can represent essentially, and in many cases very concisely, any FJPD.
  - Belief network, probabilistic network, causal network, knowledge map



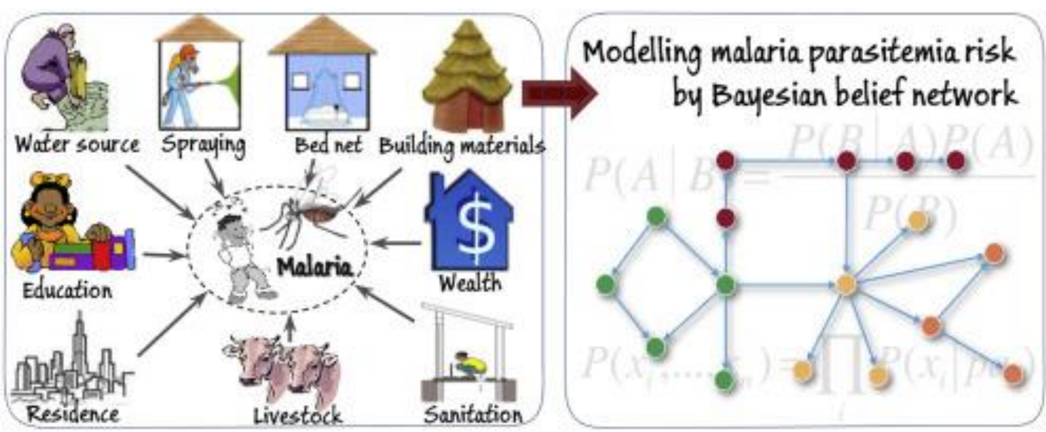
- Each node X_i presents a random variable.
  - It associates with $\mathbf{P}(X_i \mid Parent(X_i))$, which quantifies the effect of the parents on the node.
- A set of directed links connects pairs of nodes.
  - If there is a link from node X to node Y, X is a parent of Y.
  - The graph has no directed cycles $\rightarrow$ DAG.

Bayesian networks: Applications





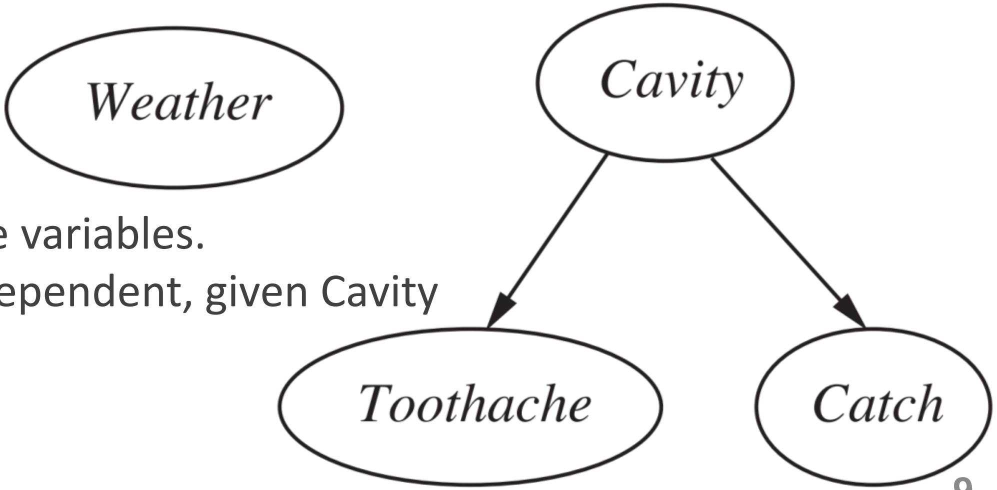
Integrating plant chemical ecology, sensors and AI for accurate pest monitoring



A model of household factors influencing the risk of malaria

Bayesian network topology

- The network topology defines the conditional independence relationships that hold in the domain.
  - $X \rightarrow Y$: X has a direct influence on Y, suggesting that causes should be parents of effects.
  - A domain expert decides what direct influences exist in the domain.
- A conditional probability distribution is specified for each variable, given its parents.



Weather is independent of the other three variables.
Toothache and Catch are conditionally independent, given Cavity

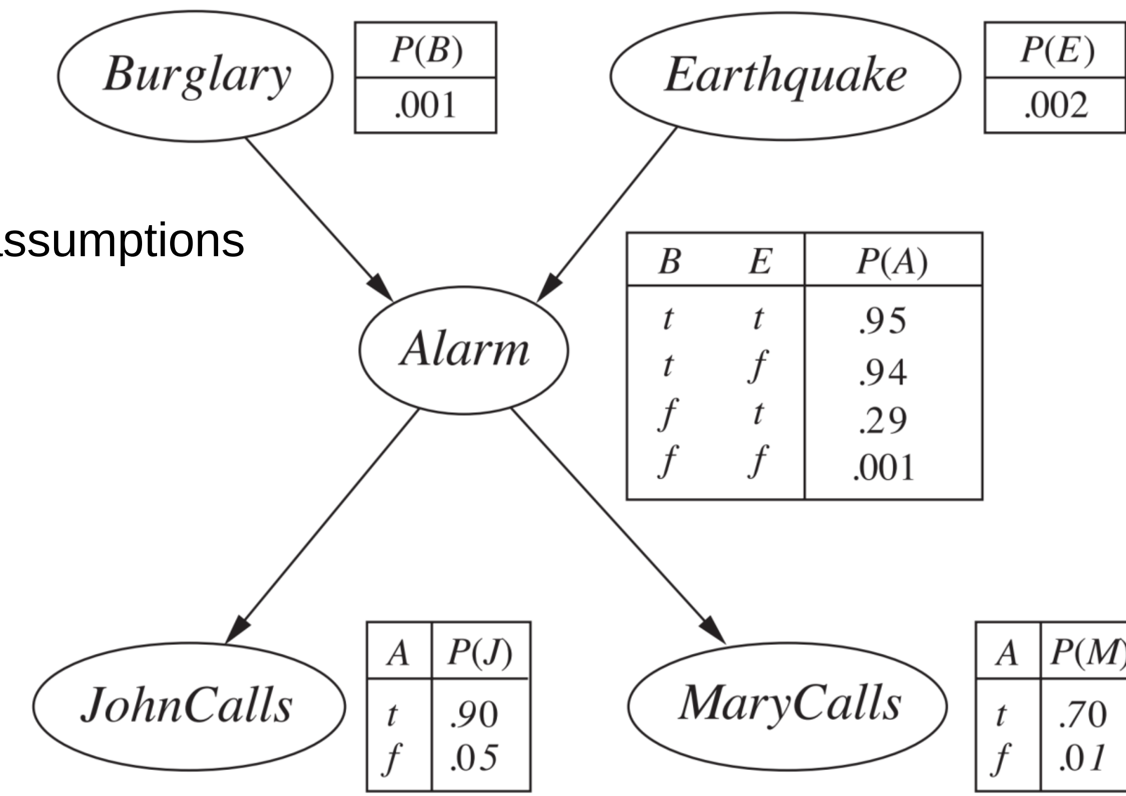
Bayesian networks: An example

The network topology shows that

- Burglary and earthquakes directly affect the probability of the alarm's going off
- Whether John and Mary call depends only on the alarm.

The network thus expresses assumptions that John and Mary do not

- Perceive burglaries directly
- Notice minor earthquakes
- Confer before calling

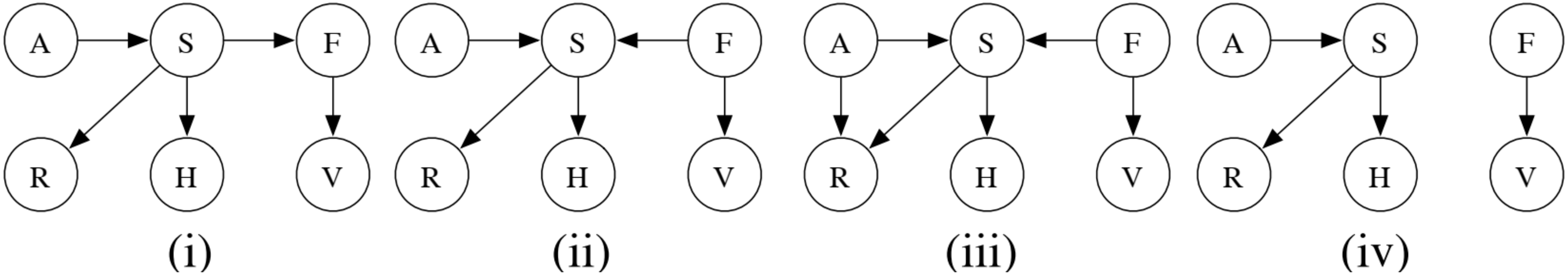


Conditional probability table (CPT)

- Each row contains the conditional probability of each node value for a conditioning case.
  - Conditioning case = a possible combination of values for the parent nodes, or a miniature possible world.
- Each row must sum to 1.
  - The entries represent an exhaustive set of cases for the variable.
  - A Boolean variable requires only the probability of true value p.
- The probabilities summarizes a potentially infinite set of circumstances in which an event does (not) happen.
  - E.g., the alarm might fail to go off (high humidity, power failure, dead battery, a dead mouse stuck inside, etc.) or John or Mary might fail to call and report it (on vacation, negligent, passing helicopter, etc.).

Quiz 01: Bayesian nets: Snuffles

- Assume there are two types of conditions: (S)inus congestion and (F)lu. Sinus congestion is caused by (A)llergy or the flu.
- There are three observed symptoms for these conditions: (H)eadache, (R)unny nose, and fe(V)er. Runny nose and headaches are directly caused by sinus congestion (only), while fever comes from having the flu (only). For example, allergies only cause runny noses indirectly.
- Assume each variable is Boolean. Consider the four Bayesian networks shown. Choose the one which models the domain best. Explain why the others do not.



Quiz 01: Bayesian nets: Snuffles

- Assume we wanted to remove the Sinus congestion (S) node. Draw the minimal Bayes network over the remaining variables which can encode the original model's marginal distribution over the remaining variables.

Inference in Bayesian networks

Full joint distribution with BN

- An entry in the joint distribution is the probability of a variable assignment, such as $\mathbf{P}(X_1 = x_1 \wedge \cdots \wedge X_n = x_n)$.

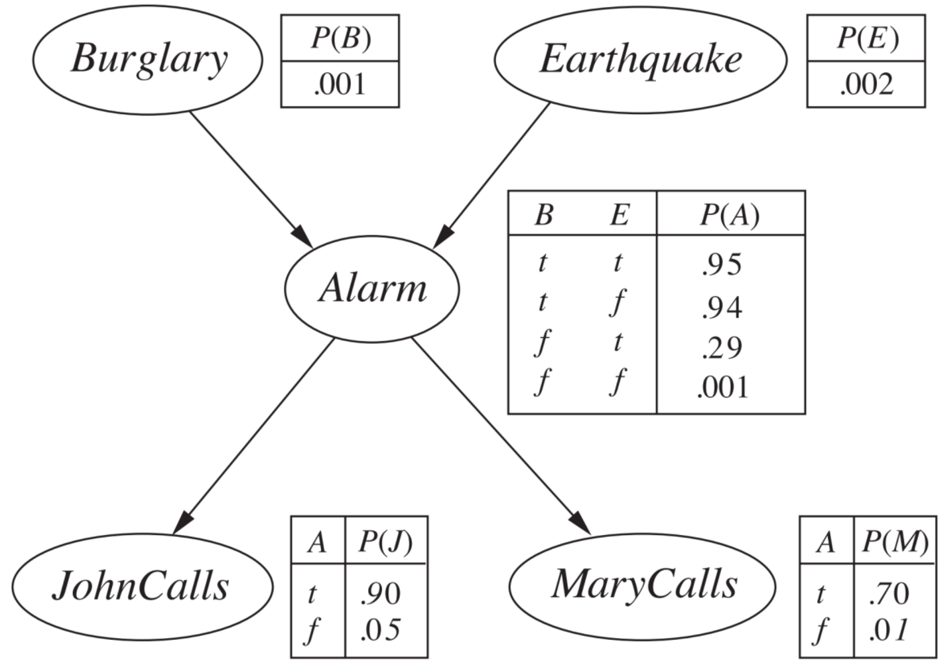
$$\mathbf{P}(x_1, \ldots, x_n) = \prod_{i=1}^{n} \mathbf{P}(X_i \mid parent(X_i))$$

  - where $parent(X_i)$ denotes the values of $Parent(X_i)$ that appear in $x_1, \ldots, x_n$.
- Thus, it is the product of the appropriate elements of the CPTs in the Bayesian network.

- A Bayesian network can be used to answer any query, by summing all the relevant joint entries.

Full joint distribution with BN

- For example, the probability that the alarm has sounded, but neither a burglary nor an earthquake has occurred, and both John and Mary call

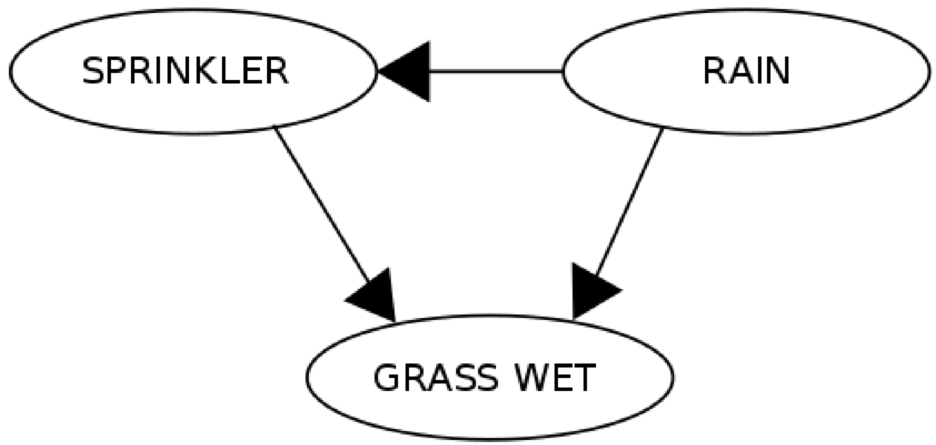


$$P(j, m, a, \neg b, \neg e) = P(j \mid a)\, P(m \mid a)\, P(a \mid \neg b \wedge \neg e)\, P(\neg b)\, P(\neg e)$$

$$= 0.90 \times 0.70 \times 0.001 \times 0.999 \times 0.998 = 0.000628$$

The wet grass example

	SPRINKLER	
RAIN	T	F
F	0.4	0.6
T	0.01	0.99



RAIN	
T	F
0.2	0.8

		GRASS WET	
SPRINKLER	RAIN	T	F
F	F	0.0	1.0
F	T	0.8	0.2
T	F	0.9	0.1
T	T	0.99	0.01

$G \ = \ Grass\ wet\ (True/False)$

$S \ = \ Sprinkler\ turned\ on\ (True/False)$

$R \ = \ Raining\ (True/False)$

The wet grass example

- What is the probability that it is raining, given the grass is wet?

$$P(R=T|G=T)=\frac{P(G=T,R=T)}{P(G=T)}=\frac{\sum_{S\in\{T,F\}}P(G=T,S,R=T)}{\sum_{S,R\in\{T,F\}}P(G=T,S,R)}$$

- Using the expansion for the joint probability function $P(G,S,R)$ and the conditional probabilities from the CPTs stated in the diagram

$$P(G=T,S=T,R=T)=P(G=T\mid S=T,R=T)\,P(S=T\mid R=T)P(R=T)$$

$$=0.99\times 0.01\times 0.2=0.00198$$

- The numerical results (subscripted by the associated variable values) are

$$P(R=T\mid G=T)=\frac{0.00198_{TTT}+0.1584_{TFT}}{0.00198_{TTT}+0.288_{TTF}+0.1584_{TFT}+0.0_{TFF}}$$

$$=\frac{891}{2491}\approx 35.77\%$$

Inference in BN: Notations

- X: query variable
- $\mathbf{E}$: the set of evidence variables $E_1, \ldots, E_m$, $\boldsymbol{e}$ is a particular observed event
- $\mathbf{Y}$: nonevidence variables, $Y_1, \ldots, Y_l$, (hidden variables)
- Thus, the complete set of variables is $\boldsymbol{X} = \{X\} \cup \mathbf{E} \cup \mathbf{Y}$

- A typical query asks for the posterior probability $\mathbf{P}(X \mid \boldsymbol{e})$
  - E.g., $\mathbf{P}(Burglary \mid JohnCalls = true, MaryCalls = true)$

$$= \langle 0.284, 0.716 \rangle$$

Inference by enumeration

- A query can be answered by computing sums of products of conditional probabilities from the Bayesian network.

$$\mathbf{P}(X \mid \boldsymbol{e}) = \alpha \mathbf{P}(X, \boldsymbol{e}) = \alpha \sum_{y} \mathbf{P}(X, \boldsymbol{e}, \boldsymbol{y})$$

  - where α stands for the constant denominator term, which is usually simplified during calculation.

Inference by enumeration

- Consider the following query

$$\mathbf{P}(Burglary \mid JohnCalls = true, MaryCalls = true)$$

- The hidden variables are $Earthquake$ and $Alarm$.
- Using initial letters for the variables, we have

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

- For simplicity, we do this for $Burglary = true$.

$$P(b \mid j, m) = \alpha \sum_{e} \sum_{a} P(b)\, P(e)\, P(a \mid b, e) P(j \mid a)\, P(m \mid a)$$

- **Complexity:** $\boldsymbol{O(n2^n)}$ for a network of n Boolean variables

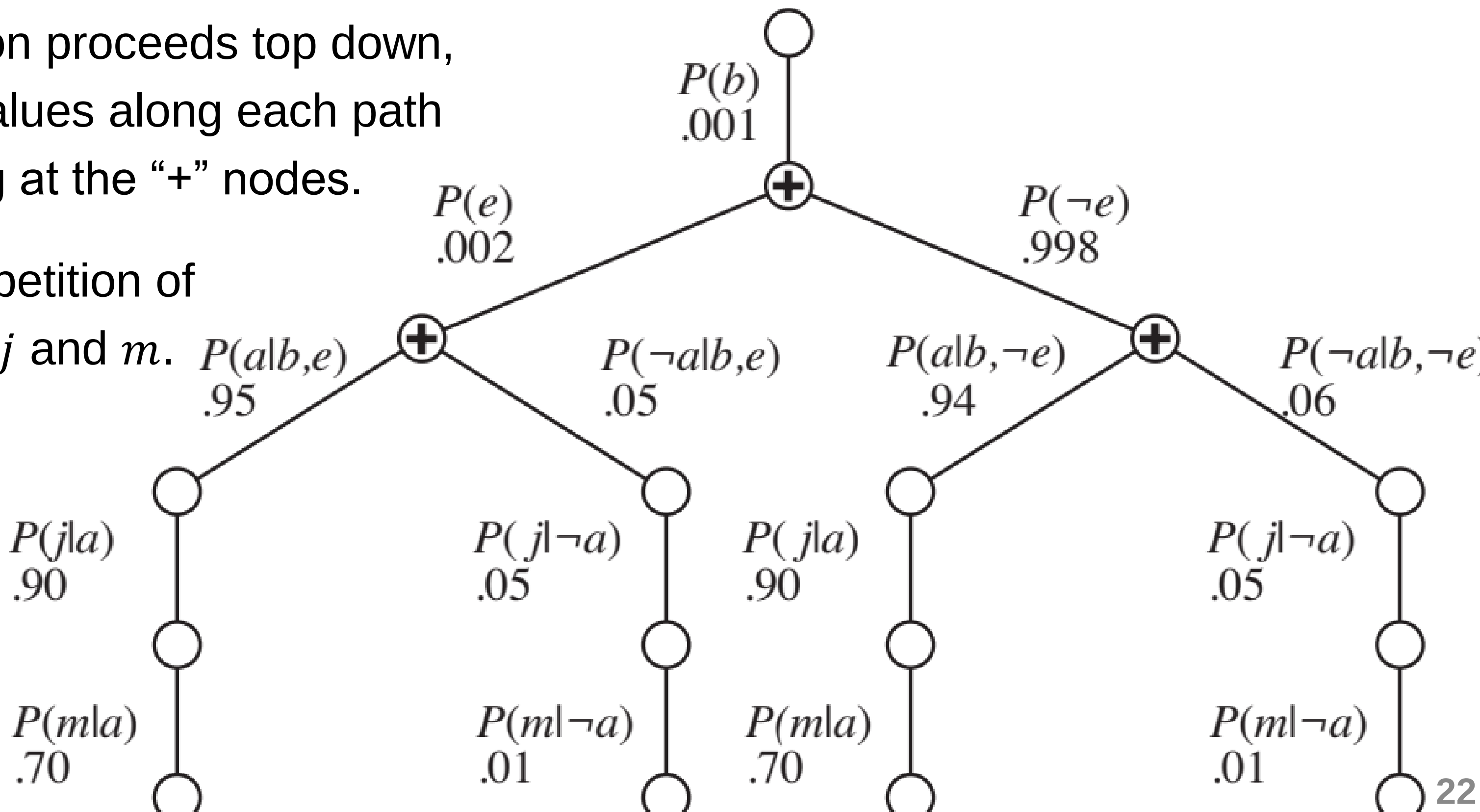
Inference by enumeration

- An improvement can be obtained as follows.

$$P(b \mid j, m) = \alpha\, P(b) \sum_{e} P(e) \sum_{a} P(a \mid b, e) P(j \mid a)\, P(m \mid a)$$

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.



Inference by enumeration: Pseudocode

function ENUMERATION-ASK(X, **e**, bn) **returns** a distribution over X
    inputs: X, the query variable
            e, observed values for variables **E**
            bn, a Bayes net with variables $\{X\} \cup \mathbf{E} \cup \mathbf{Y}$ /* **Y** = *hidden variables* */
    Q(X) ← a distribution over X, initially empty
    for each value x_i of X **do**
        Q(x_i) ← ENUMERATE-ALL(bn.VARS, $\mathbf{e}_{xi}$)
            where $\mathbf{e}_{xi}$ is **e** extended with $X = x_i$
    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 $P(y \mid parents(Y))$ × ENUMERATE-ALL(REST($vars$), **e**)
        else return $\Sigma_y P(y \mid parents(Y))$ × ENUMERATE-ALL(REST($vars$), $\mathbf{e}_y$)
            where $\mathbf{e}_y$ is **e** extended with Y = y

Inference by enumeration: Complexity

- The space complexity of ENUMERATION-ASK is only linear in the number of variables.
  - The algorithm sums over the full JPD without constructing it explicitly.
- The time complexity for a network with n Boolean variables is always $O(2^n)$
  - Better than the $O(n2^n)$ for the simple approach, but still rather grim.
- There are still repeated subexpressions to be evaluated.
  - E.g., $P(j \mid a)P(m \mid a)$ and $P(j \mid \neg a)P(m \mid \neg a)$ are computed twice, once for each value of e.

Variable elimination algorithm

- Dynamic programming: do the calculation once and save the results for later use
  - Evaluate expressions in *right-to-left* order (i.e., bottom up in the tree)
  - Store intermediate results and do summations over each variable only for portions of the expression that depend on the variable.

function ELIMINATION-ASK(X, **e**, *bn*) **returns** a distribution over X
  inputs: X, the query variable **e**, observed values for variables **E**
    bn, a Bayesian network specifying joint distribution $\mathbf{P}(X_1, \ldots, X_n)$
    factors ← [ ]
  for each *var* **in** ORDER(*bn*.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*))

Variable elimination: Factorization

- Consider the burglary network. We evaluate the following

$$\boldsymbol{P}(B \mid j,m) = \alpha \underbrace{P(B)}_{\mathbf{f_1}(B)} \sum_e \underbrace{P(e)}_{\mathbf{f_2}(E)} \sum_a \underbrace{P(a \mid B,e)}_{\mathbf{f_3}(A,B,E)} \underbrace{P(j \mid a)}_{\mathbf{f_4}(A)} \underbrace{P(m \mid a)}_{\mathbf{f_5}(A)}$$

factor

where

- $\mathbf{f_4}(A) = \begin{pmatrix} P(j \mid a) \\ P(j \mid \neg a) \end{pmatrix} = \begin{pmatrix} 0.90 \\ 0.05 \end{pmatrix}$, $\mathbf{f_5}(A) = \begin{pmatrix} P(m \mid a) \\ P(m \mid \neg a) \end{pmatrix} = \begin{pmatrix} 0.70 \\ 0.01 \end{pmatrix}$
- $\mathbf{f_3}(A,B,E)$ is a 2×2×2 matrix

- In terms of factors, the query expression is written as

$$\boldsymbol{P}(B \mid j,m) = \alpha\, \mathbf{f_1}(B) \times \sum_e \mathbf{f_2}(E) \times \sum_a \mathbf{f_3}(A,B,E) \times \mathbf{f_4}(A) \times \mathbf{f_5}(A)$$

  - where × is the pointwise product operation.

Variable elimination: Factorization

- First, we sum out A from the product of $\mathbf{f_3}$, $\mathbf{f_4}$, and $\mathbf{f_5}$.

$$\mathbf{f_6}(B,E) = \sum_a \mathbf{f_3}(A,B,E) \times \mathbf{f_4}(A) \times \mathbf{f_5}(A)$$

$$= \mathbf{f_3}(a,B,E) \times \mathbf{f_4}(a) \times \mathbf{f_5}(a) + \mathbf{f_3}(\neg a,B,E) \times \mathbf{f_4}(\neg a) \times \mathbf{f_5}(\neg a)$$

Now we are left with $\boldsymbol{P}(B \mid j,m) = \alpha\, \mathbf{f_1}(B) \times \sum_e \mathbf{f_2}(E) \times \mathbf{f_6}(B,E)$

- Next, we sum out E from the product of $\mathbf{f_2}$ and $\mathbf{f_6}$

$$\mathbf{f_7}(B) = \sum_e \mathbf{f_2}(E) \times \mathbf{f_6}(B,E) = \mathbf{f_2}(e) \times \mathbf{f_6}(B,e) + \mathbf{f_2}(\neg e) \times \mathbf{f_6}(B,\neg e)$$

- The following expression can be evaluated by taking pointwise product and normalizing the result.

$$\boldsymbol{P}(B \mid j,m) = \alpha\, \mathbf{f_1}(B) \times \mathbf{f_7}(B)$$

Variable elimination: Operators

- The pointwise product of two factors $\mathbf{f_1}$ and $\mathbf{f_2}$ gives a new factor $\mathbf{f}$ whose
  - Variables are the union of the variables in $\mathbf{f_1}$ and $\mathbf{f_2}$
  - Elements are given by the product of the corresponding elements in the two factors.
- Suppose the two factors have variables $Y_1, \ldots, Y_k$ in common.
- Then, $\mathbf{f_3}(X_1 \ldots X_j, Y_1 \ldots Y_k, Z_1 \ldots Z_l) = \mathbf{f_1}(X_1 \ldots X_j, Y_1 \ldots Y_k) \times \mathbf{f_2}(Y_1 \ldots Y_k, Z_1 \ldots Z_l)$

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	$.1 \times .4 = .04$

if all the variables are binary, $\mathbf{f_1}$ and $\mathbf{f_2}$ have 2^{j+k} and 2^{k+l} entries, respectively, and their product has 2^{j+k+l}.

Variable elimination: Operators

- Summing out a variable from a product of factors is done by adding up the submatrices formed by fixing the variable to each of its values in turn.

$$\mathbf{f}(A,B) = \sum_{a} \mathbf{f_3}(A,B,C) = \mathbf{f_3}(a,B,C) + \mathbf{f_3}(\neg a,B,C)$$

- Notice that any factor that does not depend on the variable to be summed out can be moved outside the summation.

$$\sum_{e} \mathbf{f_2}(E) \times \mathbf{f_3}(A,B,E) \times \mathbf{f_4}(A) \times \mathbf{f_5}(A) = \mathbf{f_4}(A) \times \mathbf{f_5}(A) \times \sum_{e} \mathbf{f_2}(E) \times \mathbf{f_3}(A,B,E)$$

Variable elimination: Ordering

- Every choice of ordering yields a valid algorithm
- Different orderings cause different intermediate factors to be generated during the calculation.

$$\boldsymbol{P}(B \mid j, m) = \alpha\, \mathbf{f_1}(B) \times \sum_e \mathbf{f_2}(E) \times \sum_a \mathbf{f_3}(A, B, E) \times \mathbf{f_4}(A) \times \mathbf{f_5}(A)$$

$\mathbf{f_6}$

$$= \alpha\, \mathbf{f_1}(B) \times \sum_a \mathbf{f_4}(A) \times \mathbf{f_5}(A) \times \sum_e \mathbf{f_2}(E) \times \mathbf{f_3}(A, B, E)$$

- A heuristic for variable elimination

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

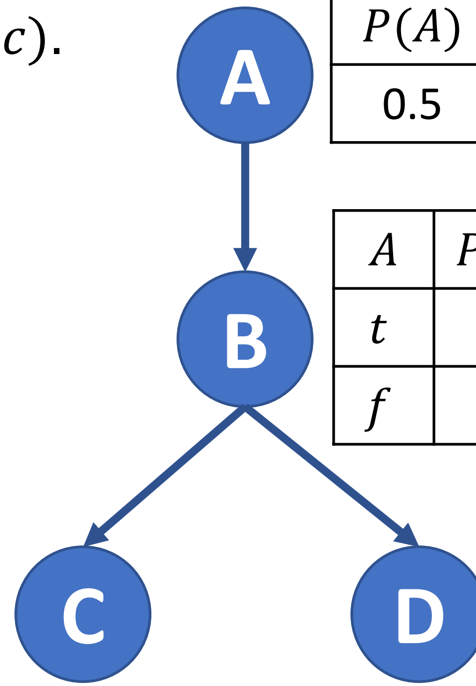
Variable elimination: An example

- Consider the following network. Calculate $P(B \mid \neg c)$.
- $\boldsymbol{P}(B \mid \neg c) = \alpha\, \underbrace{P(\neg c \mid B)}_{\mathbf{f}_1(B)} \times \sum_a \underbrace{P(B \mid a)}_{\mathbf{f}_2(A,B)} \times \underbrace{P(a)}_{\mathbf{f}_3(A)}$

  factor

- Irrelevant variable: D. Observed variable: $C = \neg c$.
- Sum out A to have $\boldsymbol{f}_4(B) = \sum_a P(B \mid a) \times P(a)$
- Join $\boldsymbol{f}_1$ and $\boldsymbol{f}_4$: $\boldsymbol{f}_5(B, \neg c) = \boldsymbol{f}_1(B) \times \boldsymbol{f}_4(B)$
- Finally, we have $\boldsymbol{P}(B \mid \neg c) = \alpha\, \boldsymbol{f}_5(B, \neg c)$
- Assume that $B = b$. Normalize for B:

$$\boldsymbol{P}(b \mid \neg c) = \frac{\boldsymbol{f}_5(b, \neg c)}{\boldsymbol{f}_5(b, c) + \boldsymbol{f}_5(\neg b, \neg c)}$$



$P(A)$
0.5

A	$P(B)$
t	0.5
f	0

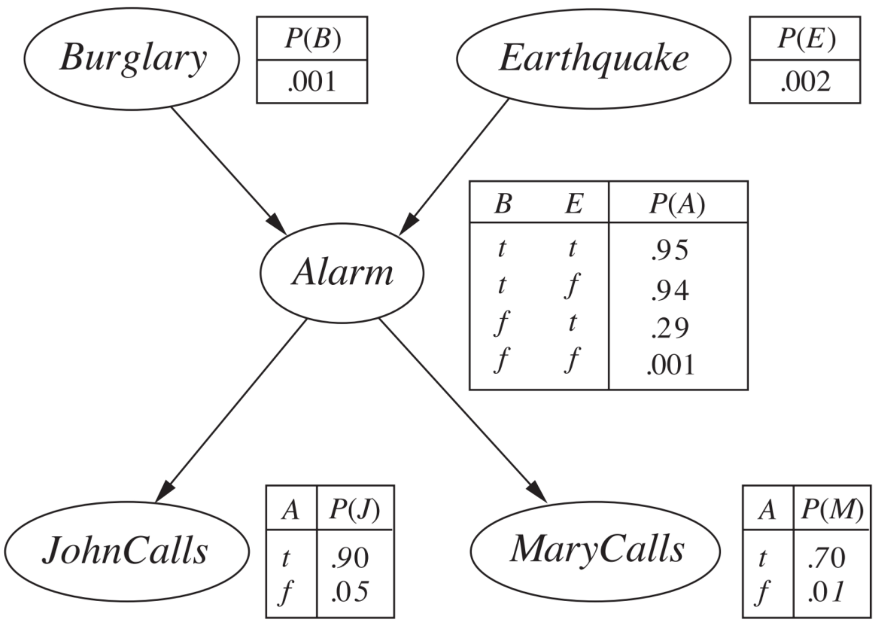
B	$P(C)$
t	0.5
f	0

B	$P(D)$
t	0.5
f	0

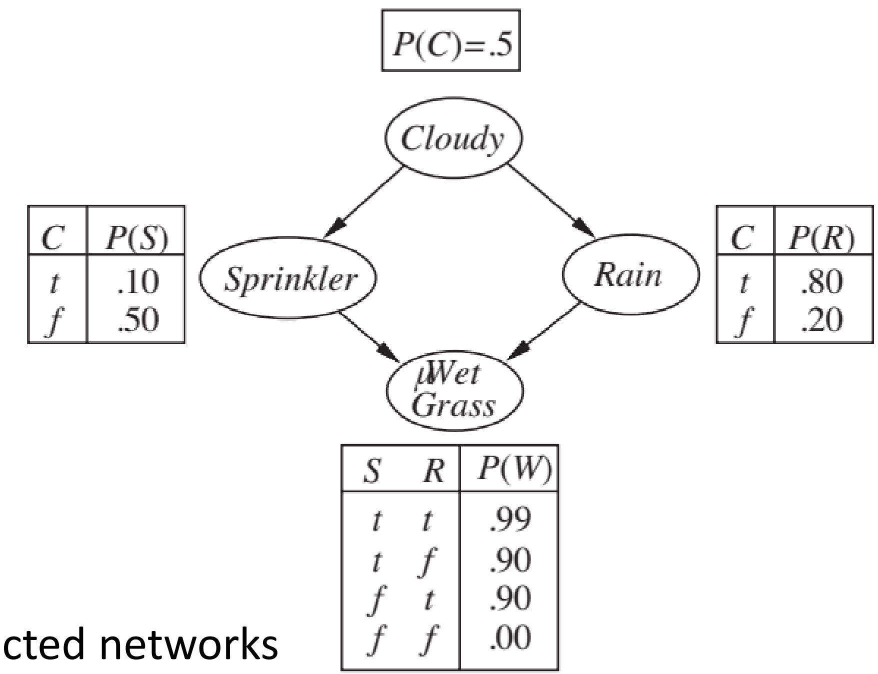
The complexity of exact inference

- The complexity of exact inference depends strongly on the structure of the network.
- Singly connected networks or polytrees: Linear time and space complexity to the network size.
  - The size is defined as the number of CPT entries.
  - If the number of parents of each node is bounded by a constant, the complexity will also be linear in the number of nodes.
- Multiply connected networks: Exponential time and space complexity in the worst case, even when the number of parents per node is bounded
  - Inference in Bayesian networks is NP-hard.

The complexity of exact inference



Singly connected networks or polytrees: there is at most one undirected path between any two nodes in the network



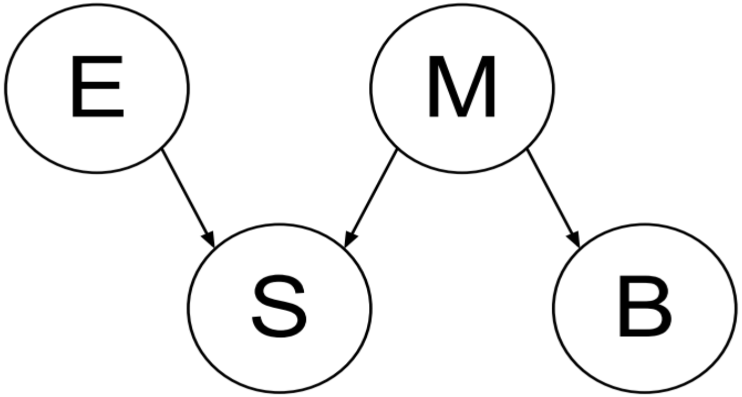
Multiply connected networks

Quiz 02: Inference in BN

- A smell of sulfur (S) can be caused either by rotten eggs (E) or as a sign of the doom brought by the Mayan Apocalypse (M). The Mayan Apocalypse also causes the oceans to boil (B).
- The Bayesian network and corresponding CPTs are shown below.

$P(E)$	
$+e$	0.4
$-e$	0.6

$P(S \mid E, M)$			
$+e$	$+m$	$+s$	1.0
$+e$	$+m$	$-s$	0.0
$+e$	$-m$	$+s$	0.8
$+e$	$-m$	$-s$	0.2
$-e$	$+m$	$+s$	0.3
$-e$	$+m$	$-s$	0.7
$-e$	$-m$	$+s$	0.1
$-e$	$-m$	$-s$	0.9



$P(M)$	
$+m$	0.1
$-m$	0.9

$P(B \mid M)$		
$+m$	$+b$	1.0
$+m$	$-b$	0.0
$-m$	$+b$	0.1
$-m$	$-b$	0.9

Quiz 02: Inference in BN

- Write down the joint probability distribution from the given network

  P(E, S, M, B) =

- Compute the following entry from the joint distribution

  P(+e, +s, –m, +b) =

- What is the probability that the oceans boil?

  P(+b) =

- What is the probability that the Mayan Apocalypse is occurring, given that the oceans are boiling?

  P(+m | +b) =

- What is the probability that rotten eggs are present, given that the Mayan Apocalypse is occurring?

  P(+e | +m) =

- What is the probability that the Mayan Apocalypse is occurring, given that there is a smell of sulfur, the oceans are boiling, and there are rotten eggs?

  P(+m | +s, +b, +e) =

Constructing a Bayesian network

Constructing a Bayesian network

- Scenario 1: Network structure known and all variables observable
  - Compute only the CPT entries
- Scenario 2: Network structure known while some variables hidden
  - Gradient descent (greedy hill-climbing) method, i.e., search for a solution along the steepest descent of a criterion function
- Scenario 3: Network structure unknown, all variables observable
  - Search through the model space to reconstruct network topology
- Scenario 4: Network structure unknown and all variables hidden
  - No good algorithms known for this purpose
- *D. Heckerman. A Tutorial on Learning with Bayesian Networks. In Learning in Graphical Models, M. Jordan, ed.. MIT Press, 1999.*

Constructing a Bayesian network

- Certain conditional independence relationships can guide the knowledge engineer to build the topology of the network.
- The Chain Rule holds for any set of random variables.

$$\mathbf{P}(x_1, \dots, x_n) = \prod_{i=1}^{n} \mathbf{P}(x_i \mid x_{i-1}, \dots, x_1)$$

$$= P(x_n \mid x_{n-1}, \dots, x_1) P(x_{n-1}, \dots, x_1)$$

$$= P(x_n \mid x_{n-1}, \dots, x_1) P(x_2 \mid x_{n-2}, \dots, x_1) \dots P(x_2 \mid x_1) P(x_1)$$

- We generally assert that, for every variable X_i in the network

$$\mathbf{P}(X_i \mid X_{i-1}, \dots, X_1) = \mathbf{P}(X_i \mid Parent(X_i)) \quad *$$

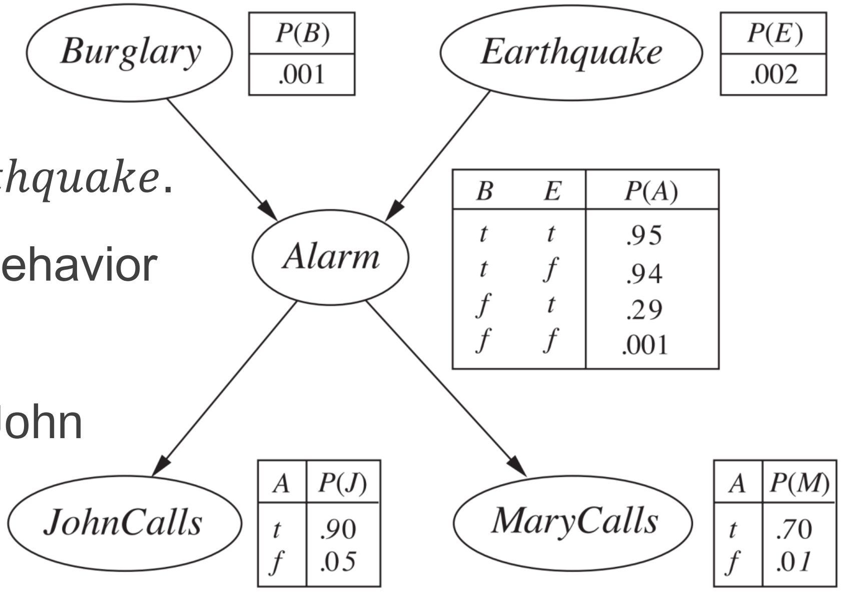
provided that $Parent(X_i) \subseteq \{X_{i-1}, \dots, X_1\}$.

Construct a Bayesian network

- Each node must be conditionally independent of its other predecessors in the node ordering, given its parents.
- *Nodes:* Identify the set of variables required to model the domain and order them, $\{X_1, \dots, X_n\}$.
  - Any order will work, but the resulting network will be more compact if the variables are ordered such that causes precede effects.
- *Links:* For $i = 1$ to n do:
  - Choose, from $\{X_1, \dots, X_{i-1}\}$, a minimal set of parents for X_i such that Equation * is satisfied.
  - For each parent insert a link from the parent to X_i.
  - CPTs: Write down the conditional probability table, $P(X_i | Parent(X_i))$.

Construct a Bayesian network

- Intuitively, the parents of node X_i should contain all those nodes in $\{X_1, \ldots, X_{i-1}\}$ that *directly influence* X_i.



- $MaryCalls$ is indirectly influenced by whether there is a $Burglary$ or an $Earthquake$.
- These events influence Mary's calling behavior only through their effect on the $Alarm$
- Given the state of the $Alarm$, whether John calls has no influence on Mary's calling
- That is,

$$P(MaryCalls \mid JohnCalls, Alarm, Earthquake, Burglary) = P(MaryCalls \mid Alarm)$$

- Thus, $Alarm$ will be the only parent node for $MaryCalls$.

Construct a Bayesian network

- The network is guaranteed to be acyclic.
  - Each node is connected only to earlier nodes.
- Bayesian networks contain no redundant probability values.
  - If there is no redundancy, then there is no chance for inconsistency.
- It is impossible for the domain expert to create a Bayesian network that violates the axioms of probability.

Example: Fire diagnosis

- You want to diagnose whether there is a fire in a building
- You can receive reports (possibly noisy) about whether everyone is leaving the building
- If everyone is leaving, this may have been caused by a fire alarm.
- If there is a fire alarm, it may have been caused by a fire or by tampering.
- If there is a fire, there may be smoke.

Fire diagnosis: Define variables

- Start by choosing the random Boolean variables for this domain
- $Tampering\ (T)$: the alarm has been tampered with
- $Fire\ (F)$: there is a fire
- $Alarm\ (A)$: there is an alarm
- $Smoke\ (S)$: there is smoke
- $Leaving\ (L)$: there are lots of people leaving the building
- $Report\ (R)$: the sensor reports that everyone are leaving the building

Fire diagnosis: Chain rule

- Define a total ordering of variables
  - Choose an order that follows the causal sequence of events
  - Fire (F) Tampering (T) Alarm (A) Smoke (S) Leaving (L) Report (R)

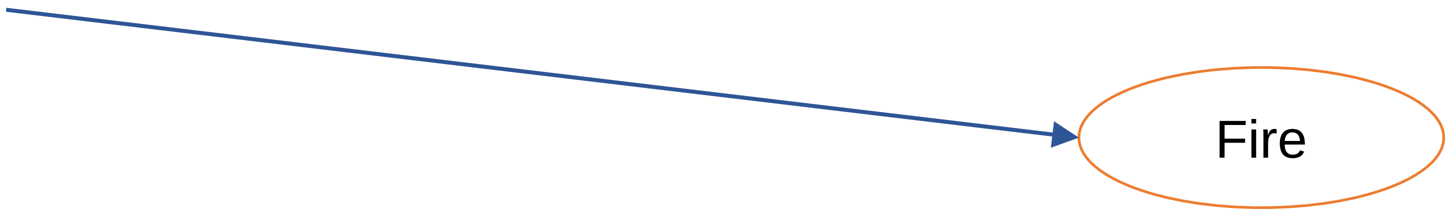
- Consider the following chain rule and use given clues to simplify it

$$\mathbf{P}(F, T, A, S, L, R) = \mathbf{P}(F)\, \mathbf{P}(T \mid F)\, \mathbf{P}(A \mid F, T)\, \mathbf{P}(S \mid F, T, A)\, \mathbf{P}(L \mid F, T, A, S)\, \mathbf{P}(R \mid F, T, A, S, L)$$

Fire diagnosis: Build a topology

- $Fire\ (F)$ is the first variable in the ordering, X_1, which has no parent.

$$\mathbf{P}(F)\ \mathbf{P}(T \mid F)\ \mathbf{P}(A \mid F, T)\ \mathbf{P}(S \mid F, T, A)\ \mathbf{P}(L \mid F, T, A, S)\ \mathbf{P}(R \mid F, T, A, S, L)$$



Fire diagnosis: Build a topology

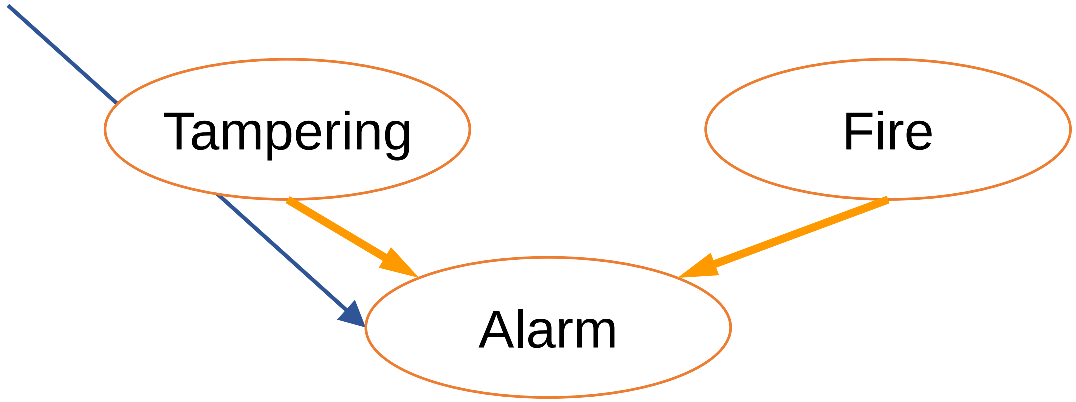
- $Tampering\ (T)$ is independent of fire
  - Learning that one is true/false would not change your beliefs about the probability of the other.

$$\mathbf{P}(F)\ \mathbf{P}(\boldsymbol{T})\ \mathbf{P}(A \mid F, T)\ \mathbf{P}(S \mid F, T, A)\ \mathbf{P}(L \mid F, T, A, S)\ \mathbf{P}(R \mid F, T, A, S, L)$$

Fire diagnosis: Build a topology

- $Alarm\ (A)$ depends on both $Fire$ and $Tampering$: it could be caused by either or both.

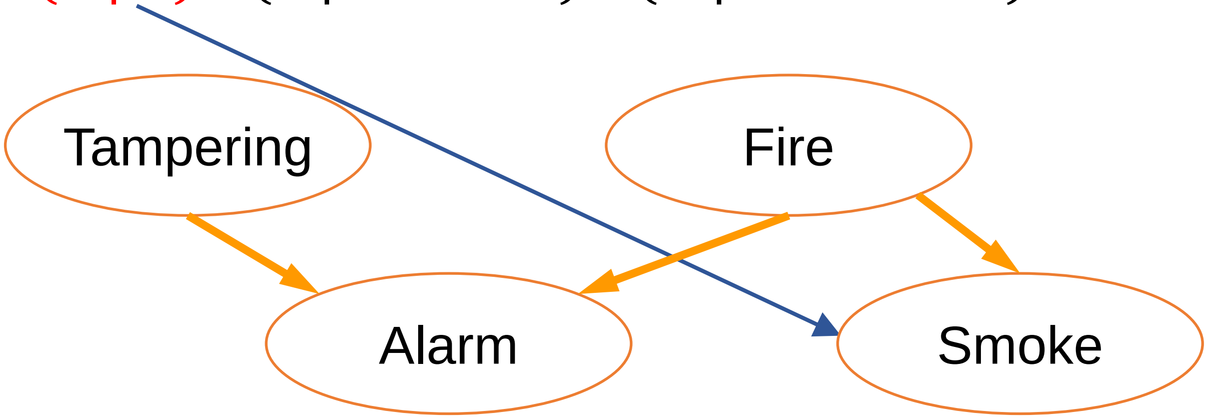
$$\mathbf{P}(F)\ \mathbf{P}(T)\ \mathbf{P}(A \mid F, T)\ \mathbf{P}(S \mid F, T, A)\ \mathbf{P}(L \mid F, T, A, S)\ \mathbf{P}(R \mid F, T, A, S, L)$$



Fire diagnosis: Build a topology

- $Smoke$ (S) is caused by $Fire$, and so is independent of $Tampering$ and $Alarm$, given whether there is a $Fire$.

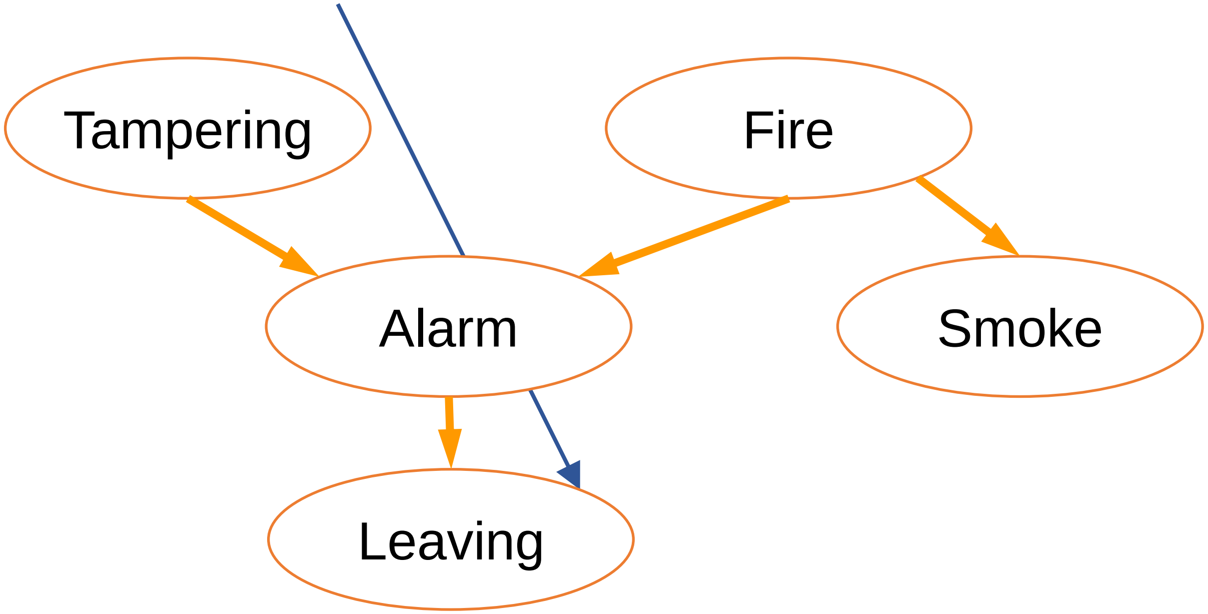
$$\mathbf{P}(F)\ \mathbf{P}(T)\ \mathbf{P}(A \mid F, T)\ \mathbf{P}(S \mid F)\ \mathbf{P}(L \mid F, T, A, S)\ \mathbf{P}(R \mid F, T, A, S, L)$$



Fire diagnosis: Build a topology

- $Leaving\ (L)$ is caused by $Alarm$, and thus is independent of the other variables, given $Alarm$.

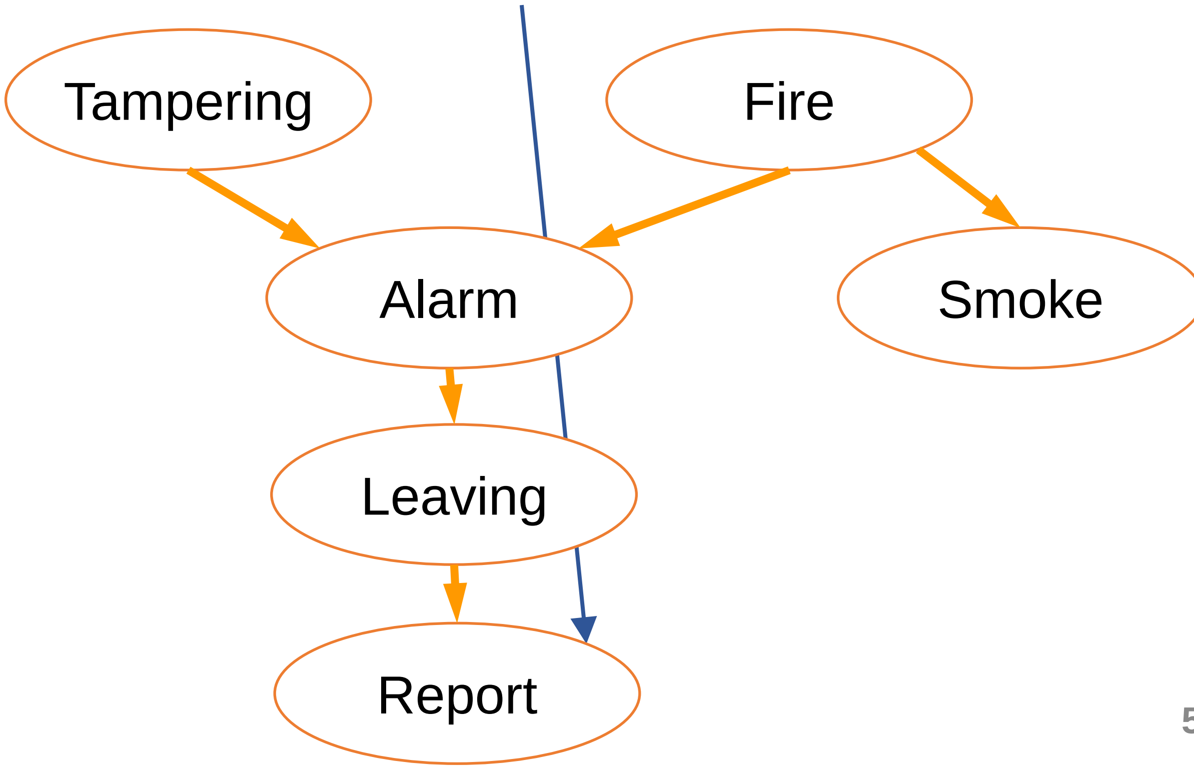
$$\mathbf{P}(F)\ \mathbf{P}(T)\ \mathbf{P}(A \mid F, T)\ \mathbf{P}(S \mid F)\ \mathbf{P}(\boldsymbol{L} \mid \boldsymbol{A})\ \mathbf{P}(R \mid F, T, A, S, L)$$



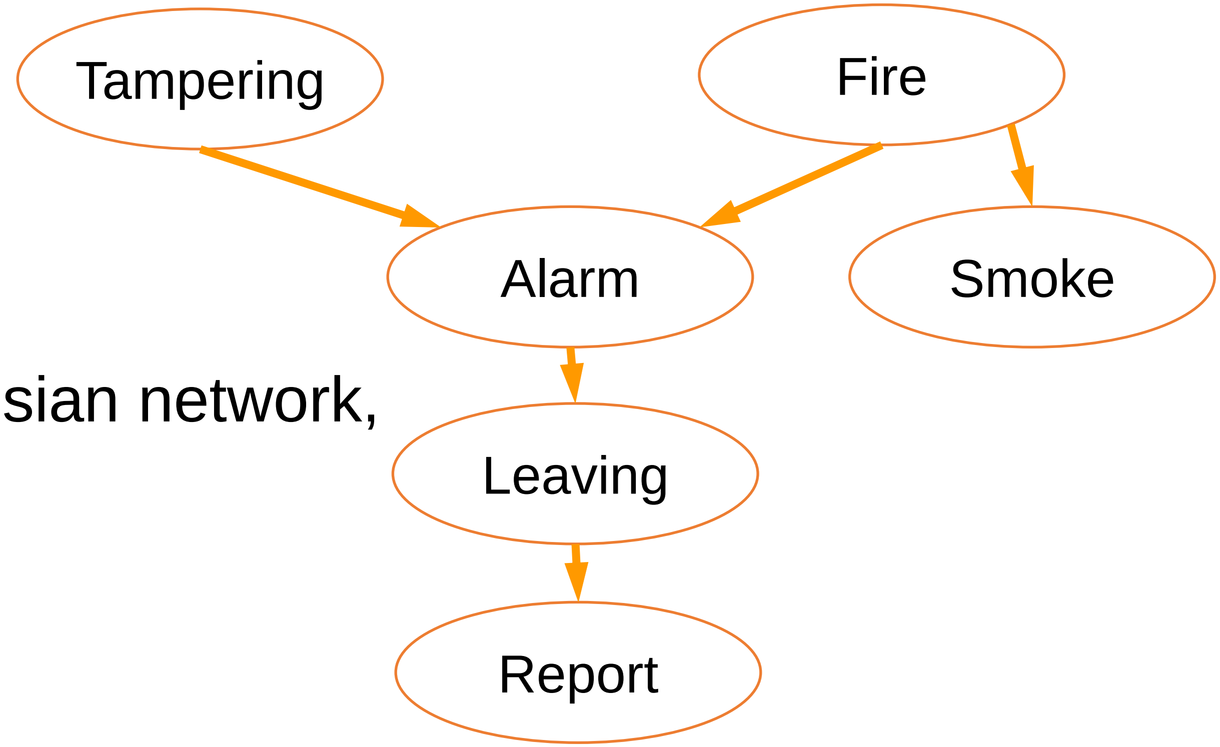
Fire diagnosis: Build a topology

- $Report\ (R)$ is caused by $Leaving$, and thus is independent of the other variables given $Leaving$

$$\mathbf{P}(F)\ \mathbf{P}(T)\ \mathbf{P}(A \mid F, T)\ \mathbf{P}(S \mid F)\ \mathbf{P}(L \mid A)\ \mathbf{P}(\boldsymbol{R} \mid \boldsymbol{L})$$



Fire diagnosis: Build a topology



- The resulting Bayesian network, and

- The corresponding compact factorization of the original FJPD

$$\mathbf{P}(F,T,A,S,L,R) = \mathbf{P}(F)\,\mathbf{P}(T)\,\mathbf{P}(A \mid F,T)\,\mathbf{P}(S \mid F)\,\mathbf{P}(L \mid A)\,\mathbf{P}(R \mid L)$$

Fire diagnosis: Specify CPTs

- How many probabilities do we need to specify for this Bayesian network?
- How many probabilities do we explicitly specify for $Fire$?

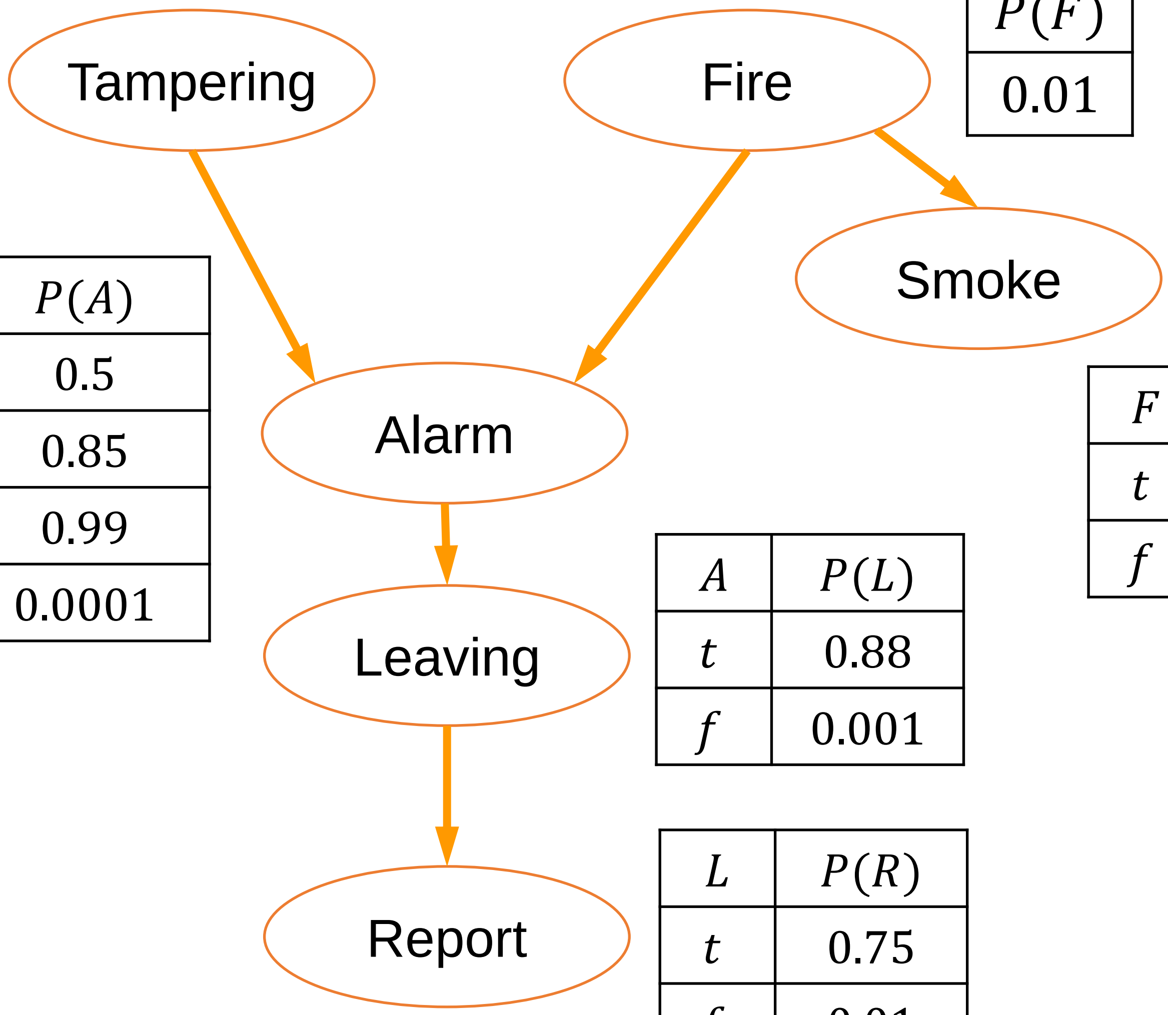
A. 1 B. 2 C. 4 D. 8

- How many probabilities do we explicitly specify for $Alarm$?

Fire diagnosis: Specify CPTs

$P(T)$
0.02

$P(F)$
0.01



T	F	$P(A)$
t	t	0.5
t	f	0.85
f	t	0.99
f	f	0.0001

F	$P(S)$
t	0.9
f	0.01

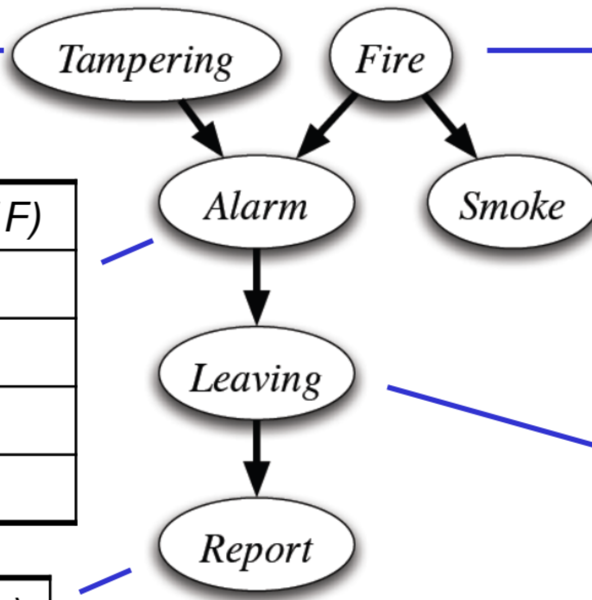
A	$P(L)$
t	0.88
f	0.001

L	$P(R)$
t	0.75
f	0.01

Fire diagnosis: Calculations

- $P(T = t, F = f, A = t, S = f, L = t, R = t) = ?$
- $P(T = t) \times P(F = f) \times P(A = t \mid T = t, F = f) \times P(S = f \mid F = f)$
  $\times P(L = t \mid A = t) \times P(R = t \mid L = t)$

P(Tampering=t)
0.02



P(Fire=t)
0.01

Tampering T	Fire F	P(Alarm=t|T,F)
t	t	0.5
t	f	0.85
f	t	0.99
f	f	0.0001

Fire F	P(Smoke=t |F)
t	0.9
f	0.01

Alarm	P(Leaving=t|A)
t	0.88
f	0.001

Leaving	P(Report=t|L)
t	0.75
f	0.01

$= 0.02 \times (1 - 0.01) \times 0.85 \times (1 - 0.01) \times 0.88 \times 0.75$

= 0.126

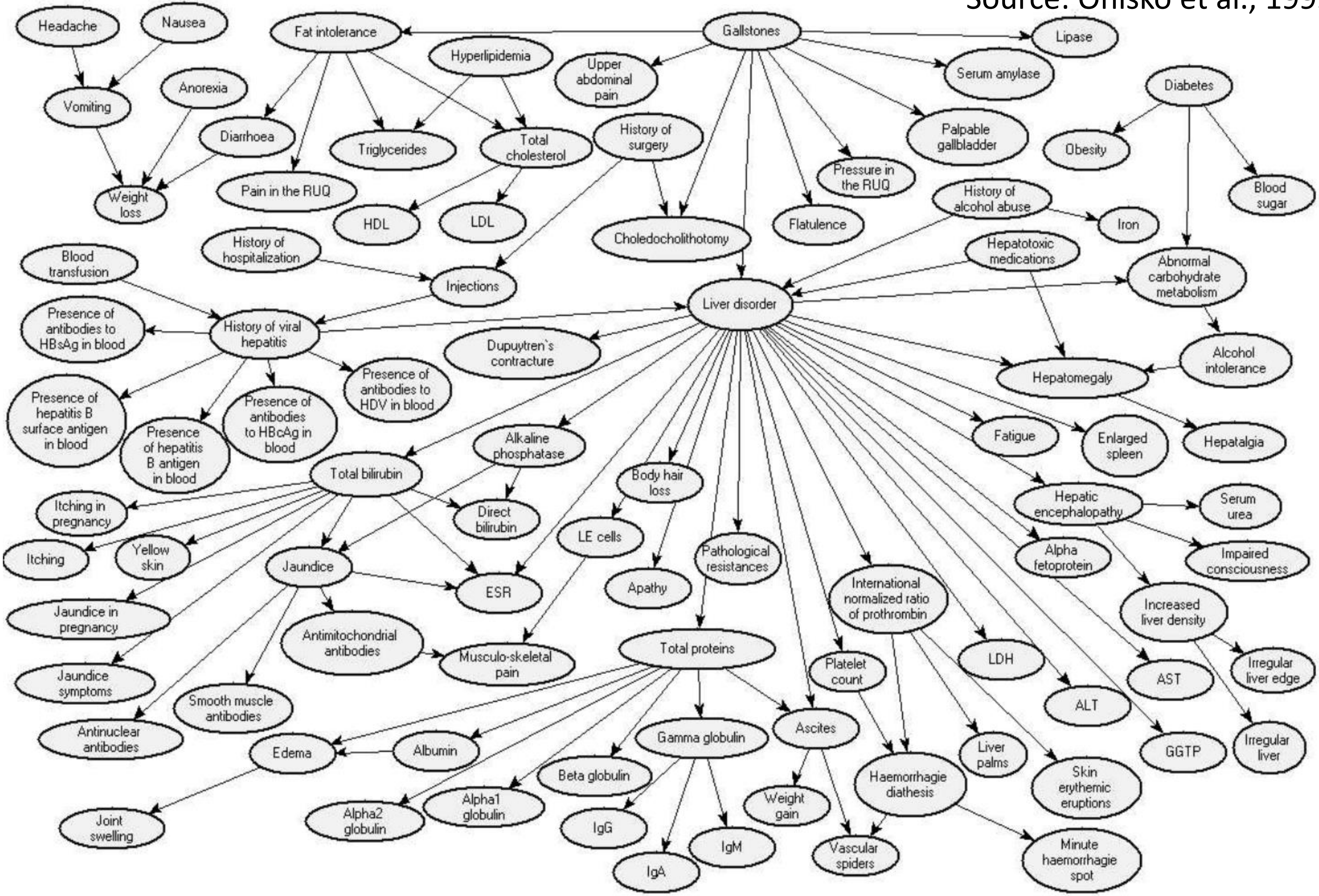
Fire diagnosis: Specify CPTs

- How many probabilities do we need to specify for this Bayesian network?
- $P(Tampering)$: 1 probability $P(T = t)$
- $P(Alarm \mid Tampering, Fire)$: 4 (independent)
  - 1 probability for each of the 4 instantiations of the parents
- For all other variables with only one parent: 2 probabilities: one for the parent being true and one for otherwise
- In total: 1+1+4+2+2+2 = **12** (compared to 2^6-1=63 for FJPD)

Bayesian networks vs. FJPD

- A CPT for a Boolean variable X_i with k Boolean parents has 2^k rows for the combinations of parent values.
- If each variable has no more than k parents, the complete network requires to specify $n2^k$ numbers.
  - For $k \ll n$, this is a substantial improvement.
  - The numbers required grow linearly with n, vs. $O(2^n)$ for the FJPD.
- For example, a Bayesian network with 30 Boolean variables, each with 5 parents, needs 30×2^5 probabilities.
  - Meanwhile, a JPD requires 2^{30} probabilities.

Source: Onisko et al., 1999



~60 nodes, max 4 parents per node

Need ~60 $\times$ 2^4 = 15 $\times$ 2^6 probabilities instead of 2^{60} probabilities for the FJPD

Bayesian networks vs. JPD

- What happens if the network is fully connected, or $k \approx n$?
  - Not much saving compared to the numbers needed for FJPD.
- Bayesian networks are useful in sparse domains (or locally structured domains).
  - A domain in which each component interacts with (is related to) a small fraction of other components
- What if this is not the case in a domain we reason about?
  - We may need to make simplifying assumptions to reduce the dependencies in a domain.

Where do the CPTs come from?

- From experts: tedious, costly, not always reliable
- From data: Machine Learning
  - There are algorithms to learn both structures and numbers.
  - It can be hard to get enough data.
- Still, usually better than specifying the FJPD.

THE END