

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 Bayesian Networks

Knowledge in Uncertain Domain

- *Full joint probability distribution*
- *Bayesian Networks*



Full joint probability distribution

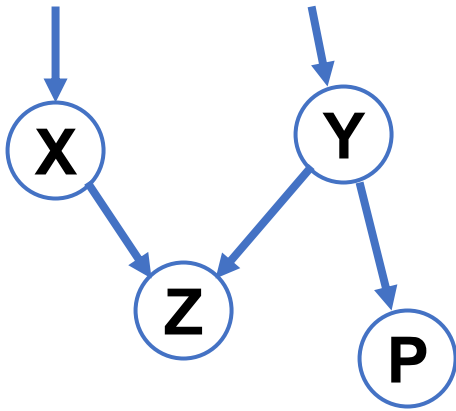
- The full joint probability distribution (JPD) can answer any question about the domain.
 - It can become intractably large as the number of variables grows.
 - Specifying probabilities for possible worlds one by one is unnatural and tedious.
- (Conditional) independence relationships among variables can greatly reduce the number of probabilities required.
- **Bayesian networks** can **represent essentially**, and in many cases very concisely, **any full JPD**.
 - Belief network, probabilistic network, causal network, knowledge map

Bayesian networks

- A **Bayesian network** is a **directed graph** in which each node is annotated with **quantitative probability information**.
- Each **node** presents a **random discrete/continuous variable**.
- A set of **directed links** or arrows **connects pairs of nodes**.
 - If there is an arrow from node X to node Y , X is a parent of Y .
 - The graph has no directed cycles, and hence is a DAG.
- Each node X_i has a probability distribution $P(X_i | Parent(X_i))$ that quantifies the **effect of the parents** on the node.

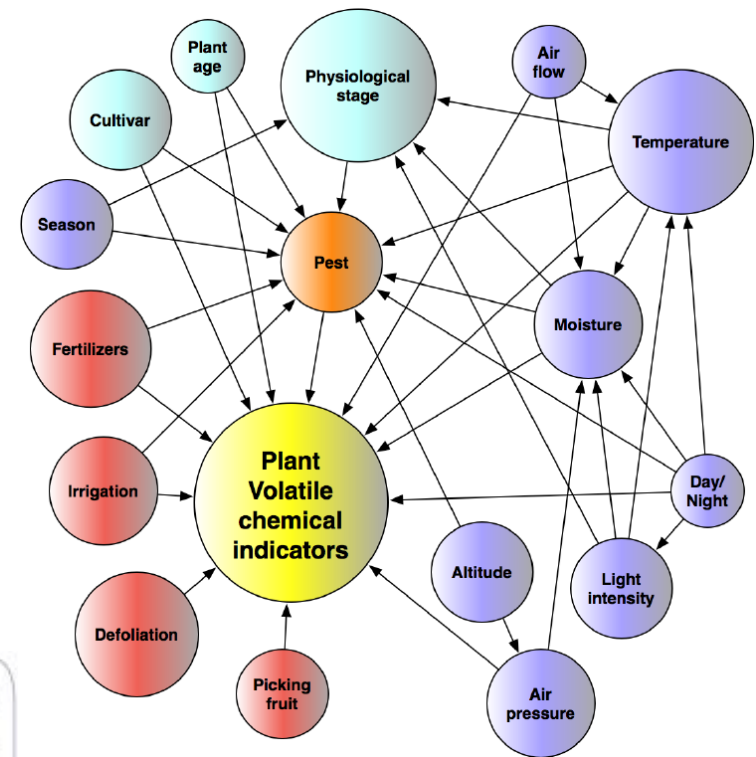
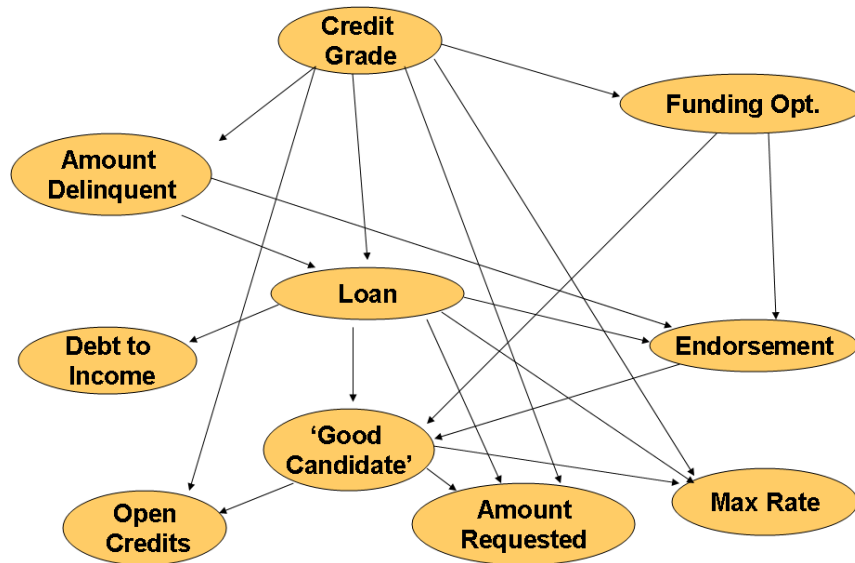
Bayesian networks

- A **Bayesian network (BN)** is a directed graph in which each node is annotated with quantitative probability information.

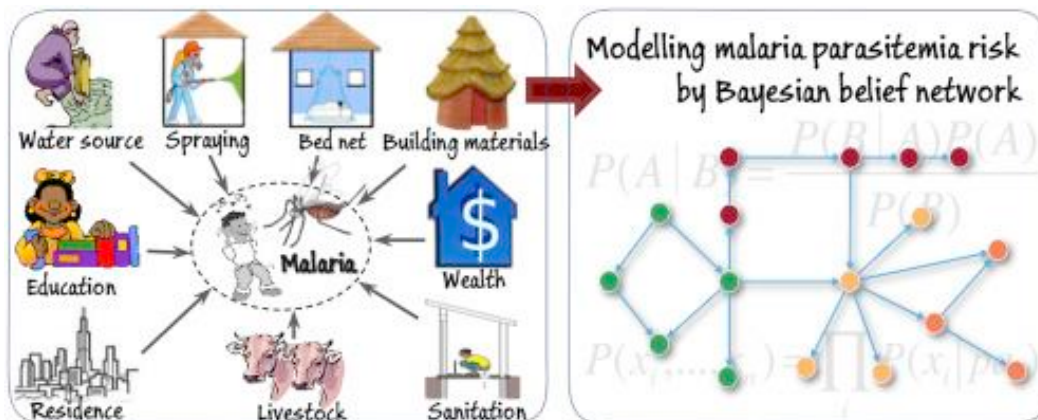


- X and Y are the parents of Z . Y is the parent of P
- No dependency between Z and P
- No loops/cycles

Applications of Bayesian network



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

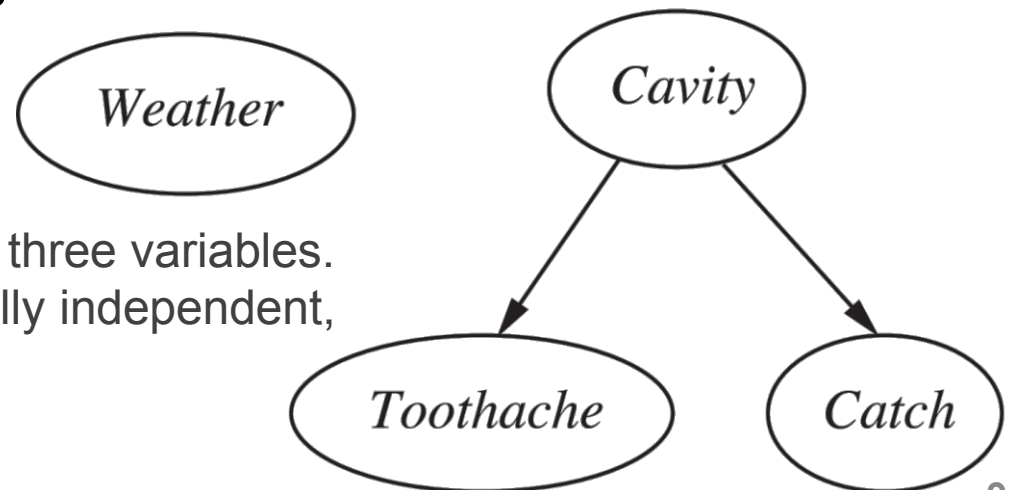


A model of household factors influencing the risk of malaria

Bayesian network topology

- The network topology defines the conditional independence relationships that hold in the domain.
 - An arrow means that X has a direct influence on Y , which suggests that causes should be parents of effects.
 - A domain expert decides what direct influences exist.
- Then, specify a conditional probability distribution for each variable, given its parents

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



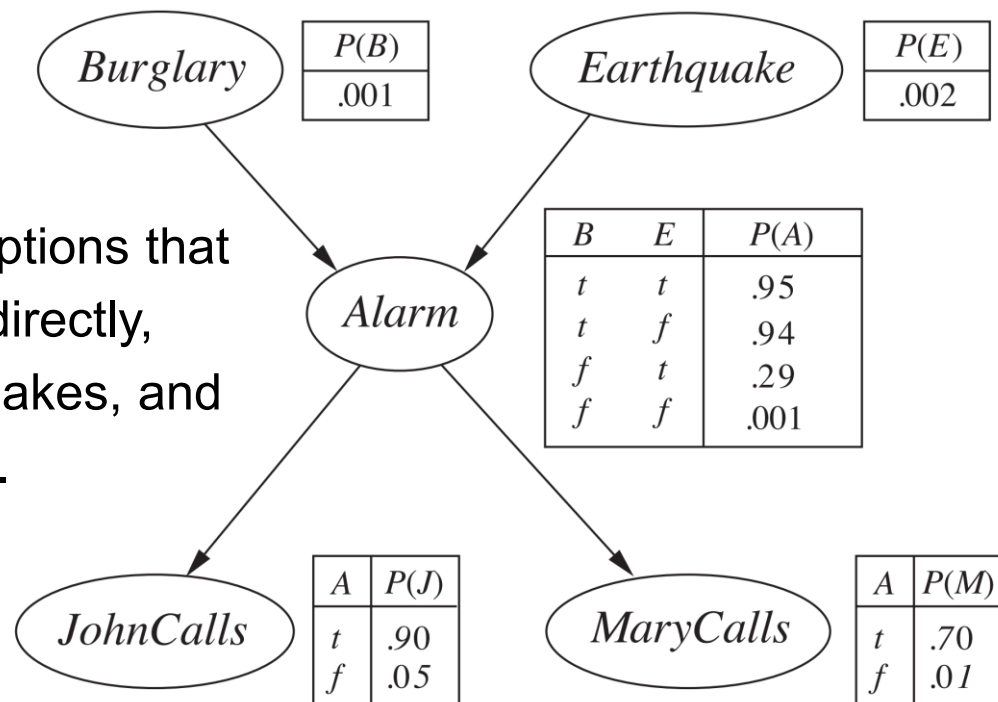
Another example of Bayesian network

The network structure shows that

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

The network thus expresses assumptions that

- They do not perceive burglaries directly,
- They do not notice minor earthquakes, and
- They do not confer before calling.



Conditional probability table (CPT)

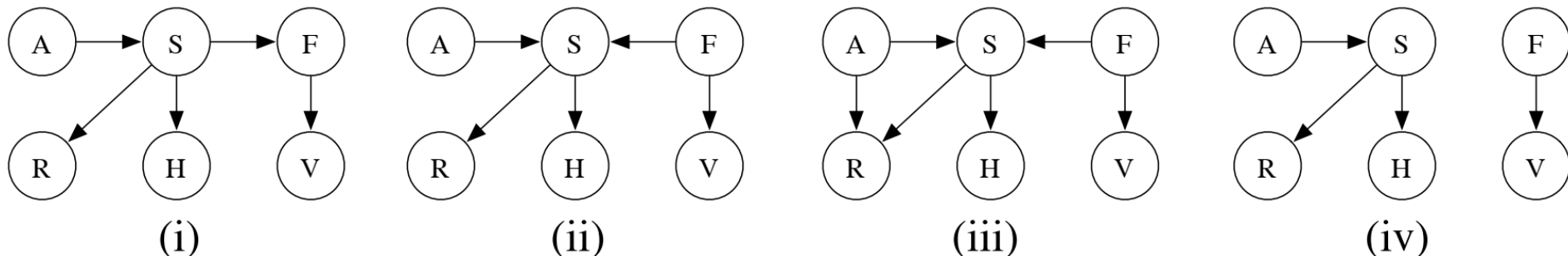
- Each row contains the conditional probability of each node value for a **conditioning case**.
 - A conditioning case is a possible combination of values for the parent nodes, or a miniature possible world.
- Each row must **sum to 1**.
 - The entries represent an exhaustive set of cases for the variable.
 - For Boolean variables, given that the probability of a true value is p , the second value $1 - p$ is omitted.
- **A node with no parents** has only one row, representing the **prior probabilities** of each possible value of the variable.

Conditional probability table (CPT)

- 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.).
- In this way, a small agent can cope with a very large world, at least approximately.
 - The degree of approximation can be improved with more relevant information introduced.

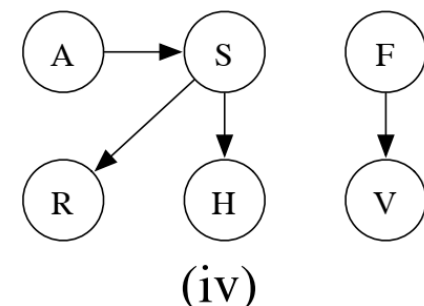
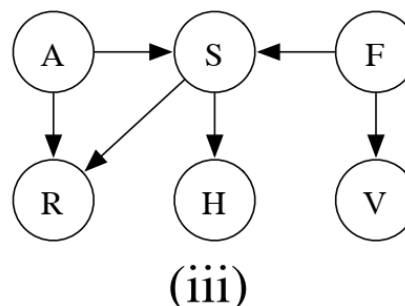
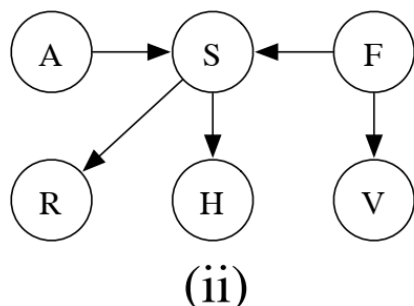
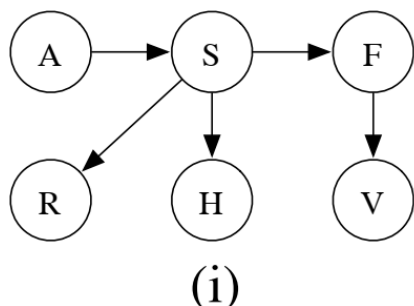
Quiz 01: Bayesian nets: Snuffles

- 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.



Quiz 01: Bayesian nets: Snuffles

- Consider the four Bayes Nets shown. Choose the one which models the domain (as described in the previous slide) best.



- For each network, do the following
 - If it models the domain exactly as above, write correct.
 - If it has too many conditional independence properties, write extra independence and state one that it has but should not have.
 - If it has too few conditional independence properties, write missing independence and state one that it should have but does not have.

Quiz 01: Bayesian nets: Snuffles

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

Inference in Bayesian Network

- *Representing the full joint probability distribution*
- *Exact inference in Bayesian Networks*
- *The complexity of exact inference*



Represent the full joint distribution

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

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

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

Represent the full joint distribution

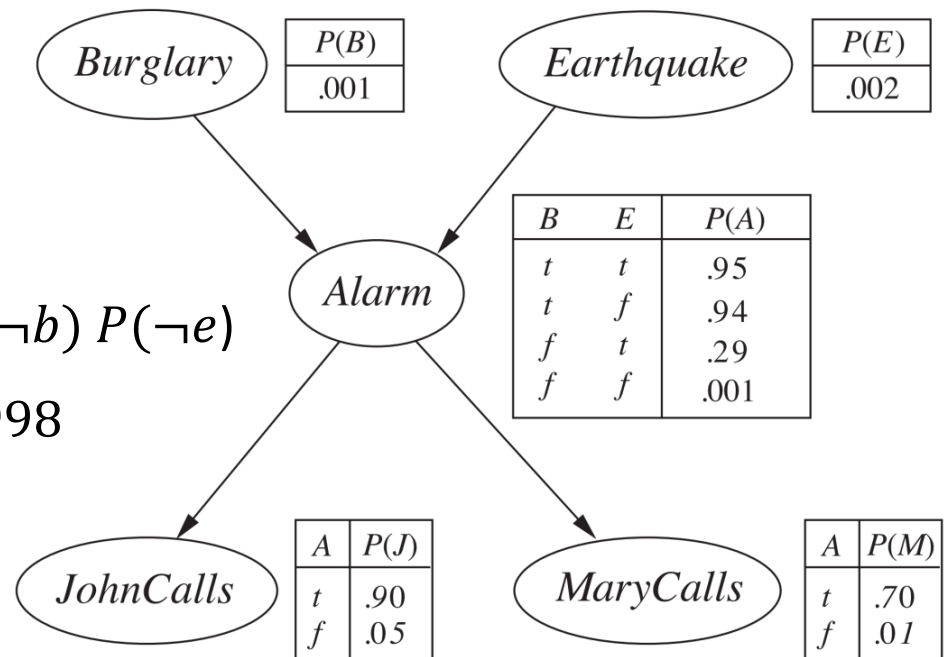
- For example, calculate the probability that the alarm has sounded, but neither a burglary nor an earthquake has occurred, and both John and Mary call

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

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

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

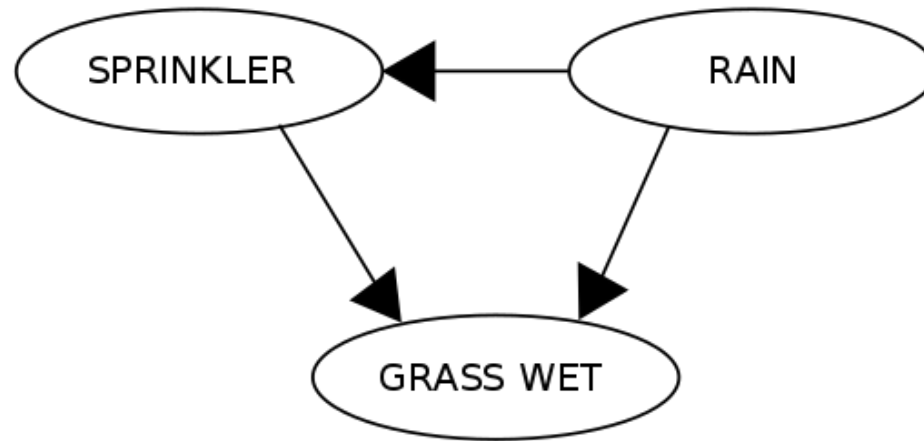
$$= 0.000628$$



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

The wet grass example

RAIN	SPRINKLER	
	T	F
F	0.4	0.6
T	0.01	0.99



	RAIN	
	T	F
	0.2	0.8

SPRINKLER	RAIN	GRASS WET	
		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

$$\begin{aligned} P(G = T, S = T, R = T) &= P(G = T | S = T, R = T) P(S = T | R = T) P(R = T) \\ &= 0.99 \times 0.01 \times 0.2 = 0.00198 \end{aligned}$$

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

$$\begin{aligned} P(R = T | G = T) &= \frac{0.00198_{TTT} + 0.1584_{TFT}}{0.00198_{TTT} + 0.288_{TTF} + 0.1584_{TFT} + 0.0_{TFF}} \\ &= \frac{891}{2491} \approx 35.77\% \end{aligned}$$

Notations

- X denotes the **query variable**.
- E denotes the set of **evidence variables** E_1, \dots, E_m , and e is a particular **observed event**.
- Y denotes the nonevidence, non-query variables Y_1, \dots, Y_l (called the **hidden variables**).
- Thus, the complete set of variables is $\mathbf{X} = \{X\} \cup E \cup Y$.
- A typical query asks for the posterior probability $P(X \mid e)$.
 - E.g., $P(\text{Burglary} \mid \text{JohnCalls} = \text{true}, \text{MaryCalls} = \text{true})$
 $= \langle 0.284, 0.716 \rangle$

Inference by enumeration

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

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

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

Inference by enumeration

- Consider the following query

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

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

$$P(B \mid j, m) = \alpha P(B, j, m) = \alpha \sum_e \sum_a P(B, j, m, e, a)$$

- 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: **$O(n2^n)$** for a network with n Boolean variables

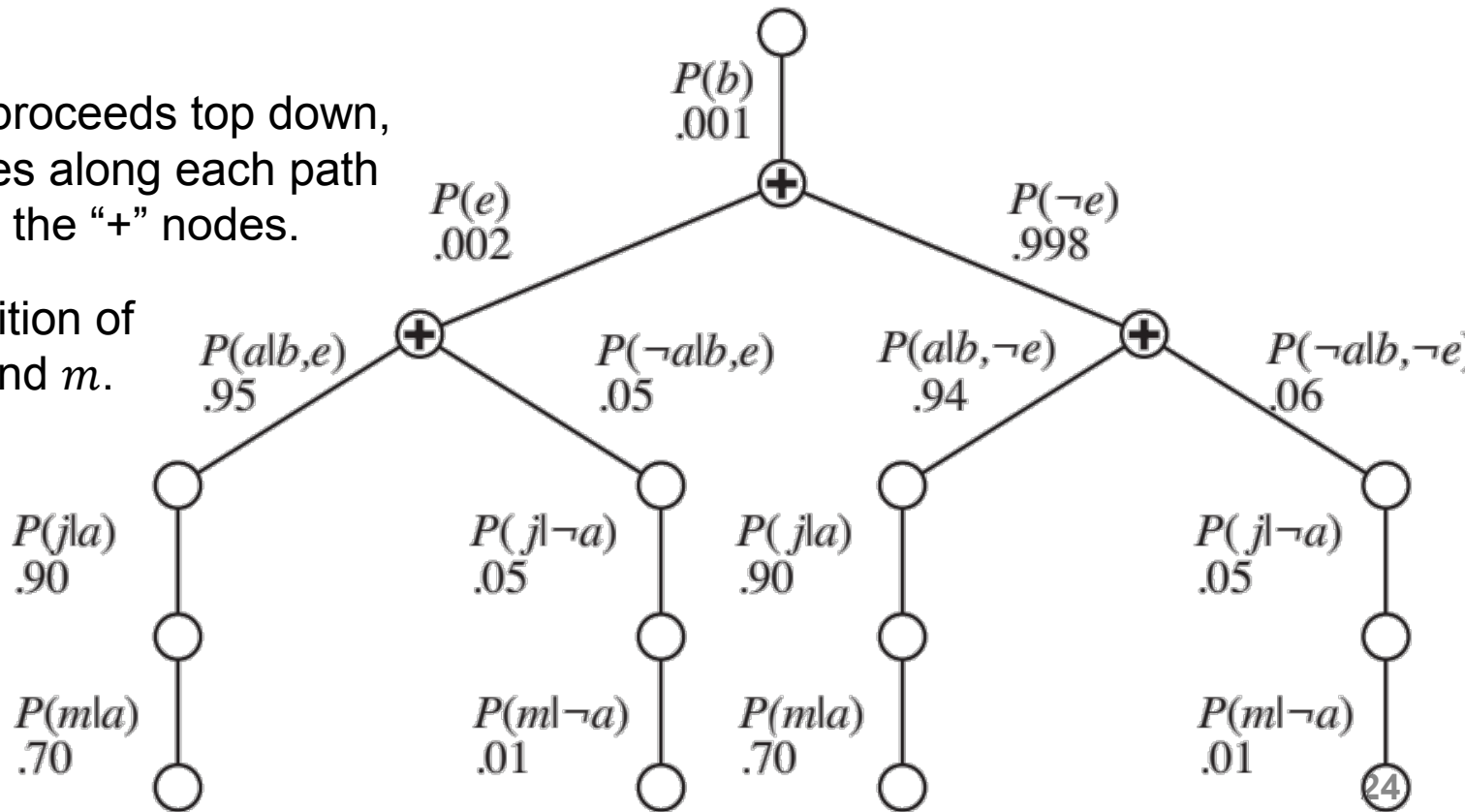
Inference by enumeration

- An improvement can be obtained from the following

$$P(b \mid j, m) = \alpha P(b) \sum_e P(e) \sum_a P(a \mid b, e) P(j \mid a) P(m \mid a)$$

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

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


function ENUMERATE-ALL($vars, \mathbf{e}$) **returns** a real number
if EMPTY?($vars$) **then return** 1.0
 $Y \leftarrow$ FIRST($vars$)
if Y has value y in \mathbf{e}
 then return $P(y \mid \text{parents}(Y)) \times \text{ENUMERATE-ALL}(\text{REST}(vars), \mathbf{e})$
 else return $\sum_y P(y \mid \text{parents}(Y)) \times \text{ENUMERATE-ALL}(\text{REST}(vars), \mathbf{e}_y)$
 where \mathbf{e}_y is \mathbf{e} extended with $Y = y$

Inference by enumeration

- The space complexity of ENUMERATION-ASK is only linear in the number of variables.
 - The algorithm sums over the full JPD without ever 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

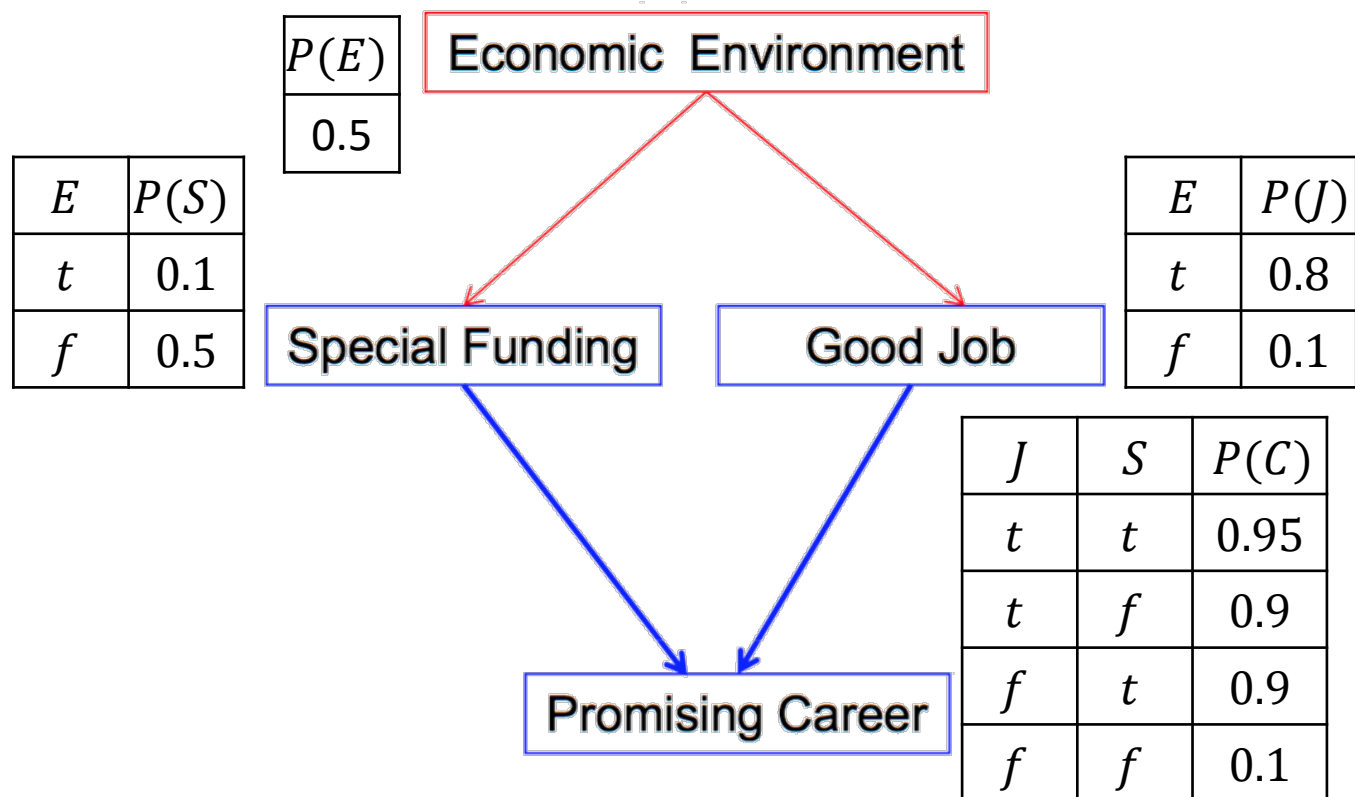
- This is a form of dynamic programming.
- Do the calculation once and save the results for later use
 - Evaluate expressions in *right-to-left* order (i.e., bottom up in the tree)
 - Store intermediate results and do summations over each variable only for portions of the expression that depend on the variable.
- Consider the burglary network. We evaluate the following

$$P(B \mid j, m) = \alpha P(B) \sum_e P(e) \sum_a P(a \mid B, e) P(j \mid a) P(m \mid a)$$


$f_1(B)$ $f_2(E)$ $f_3(A, B, E)$ $f_4(A)$ $f_5(A)$

Quiz 02: Inference in Bayesian net

- Consider the following Bayesian network. Compute the causal inference $P(C|E)$ and diagnostic inference $P(E|C)$.



Bayesian Network Construction



Construct 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.*

Construct 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.

$$\begin{aligned}P(x_1, \dots, x_n) &= \prod_{i=1}^n P(x_i | x_{i-1}, \dots, x_1) \\&= P(x_n | x_{n-1}, \dots, x_1) P(x_{n-1}, \dots, x_1) \\&= P(x_n | x_{n-1}, \dots, x_1) P(x_2 | x_{n-2}, \dots, x_1) \dots P(x_2 | x_1) P(x_1)\end{aligned}$$

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

$$P(X_i | X_{i-1}, \dots, X_1) = P(X_i | \text{Parent}(X_i)) *$$

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

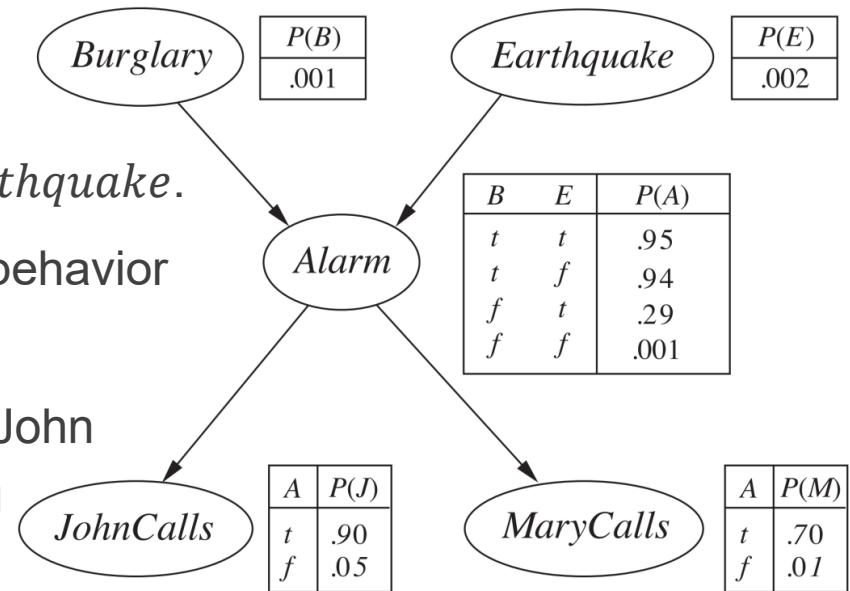
Construct a Bayesian network

- 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, $P(X_i | \text{Parent}(X_i))$.

Construct a Bayesian network

- Intuitively, the parents of node X_i should contain all those nodes in $\{X_1, \dots, X_{i-1}\}$ 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,

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

- 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.

Example: Fire diagnosis

- 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

Example: Fire diagnosis

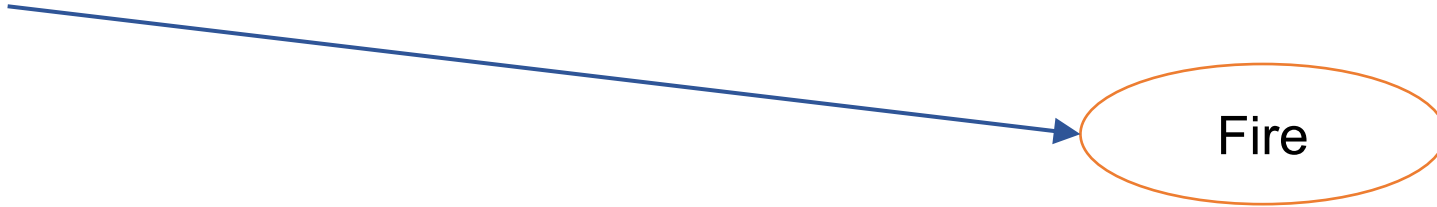
- Define a total ordering of variables
 - Choose an order that follows the causal sequence of events
 - Fire (F) Tampering (T) Alarm (A) Smoke (S) Leaving (L) Report (R)
- Consider the following chain rule and use given clues to simplify it

$$\begin{aligned} P(F, T, A, S, L, R) = & P(F) P(T | F) P(A | F, T) P(S | F, T, A) \\ & P(L | F, T, A, S) P(R | F, T, A, S, L) \end{aligned}$$

Example: Fire diagnosis (Topology)

- *Fire* (F) is the first variable in the ordering, X_1 , which does not have parents.

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



Example: Fire diagnosis (Topology)

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

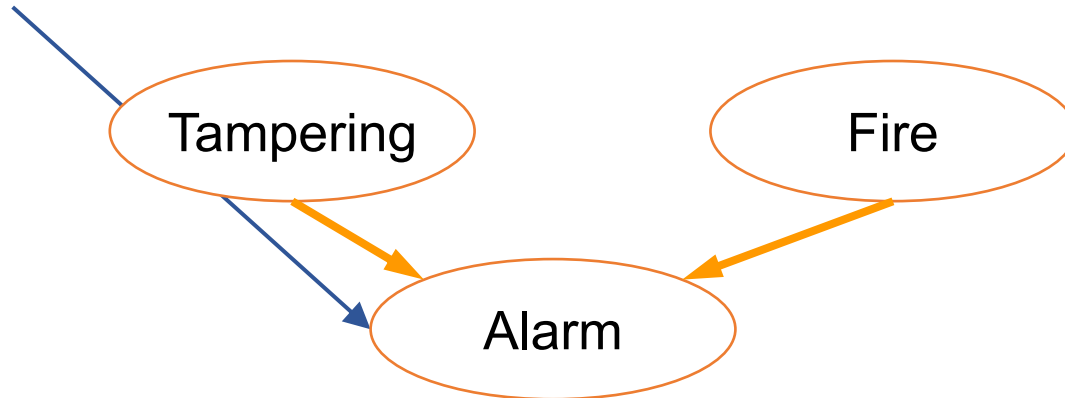
$P(F)$ **$P(T)$** $P(A \mid F, T)$ $P(S \mid F, T, A)$ $P(L \mid F, T, A, S)$ $P(R \mid F, T, A, S, L)$



Example: Fire diagnosis (Topology)

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

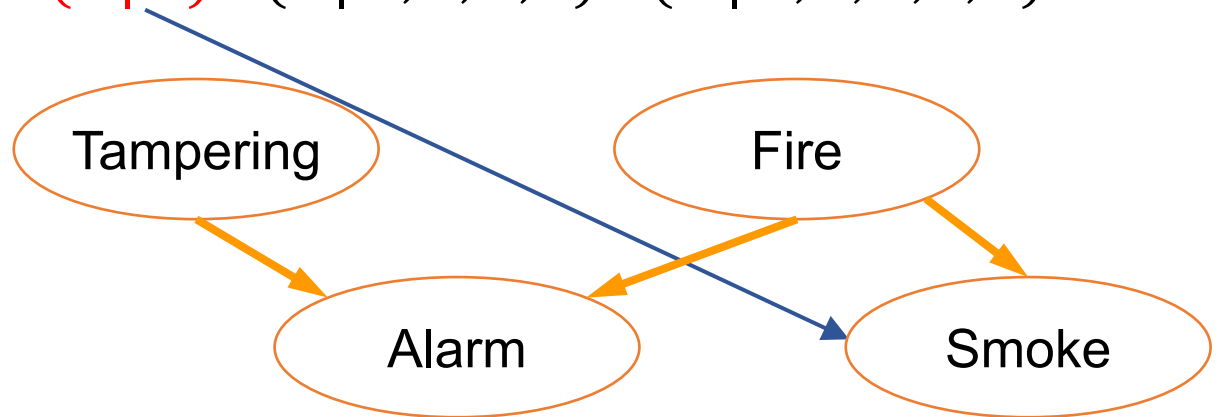
$$P(F) P(T) \mathbf{P(A \mid F, T)} P(S \mid F, T, A) P(L \mid F, T, A, S) P(R \mid F, T, A, S, L)$$



Example: Fire diagnosis (Topology)

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

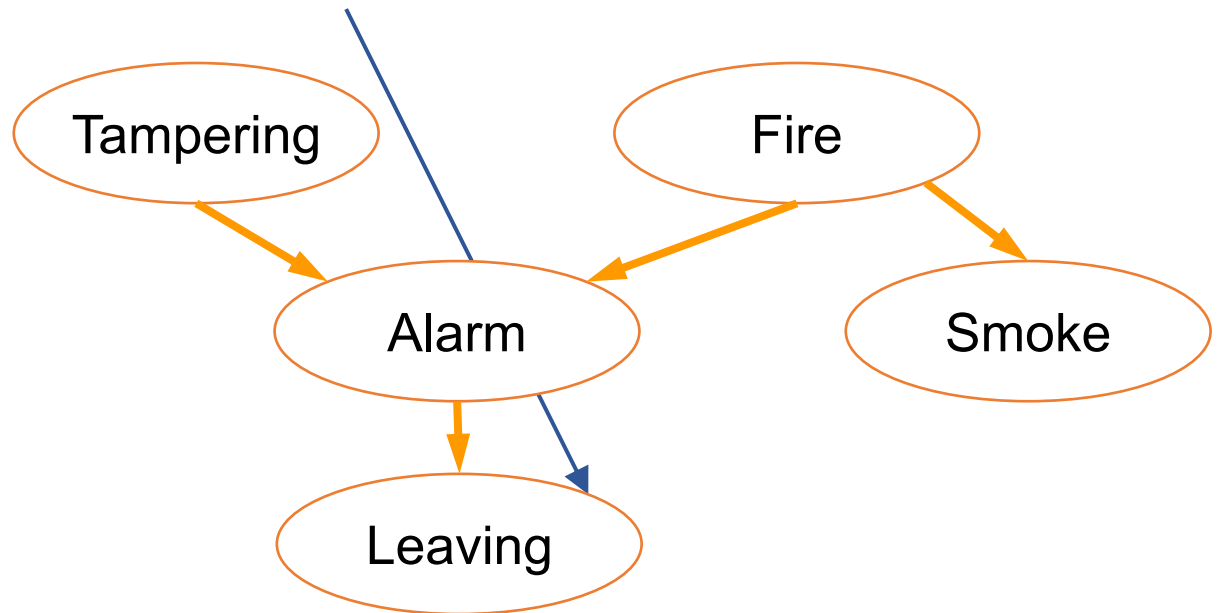
$$P(F) P(T) P(A | F, T) \mathbf{P(S | F)} P(L | F, T, A, S) P(R | F, T, A, S, L)$$



Example: Fire diagnosis (Topology)

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

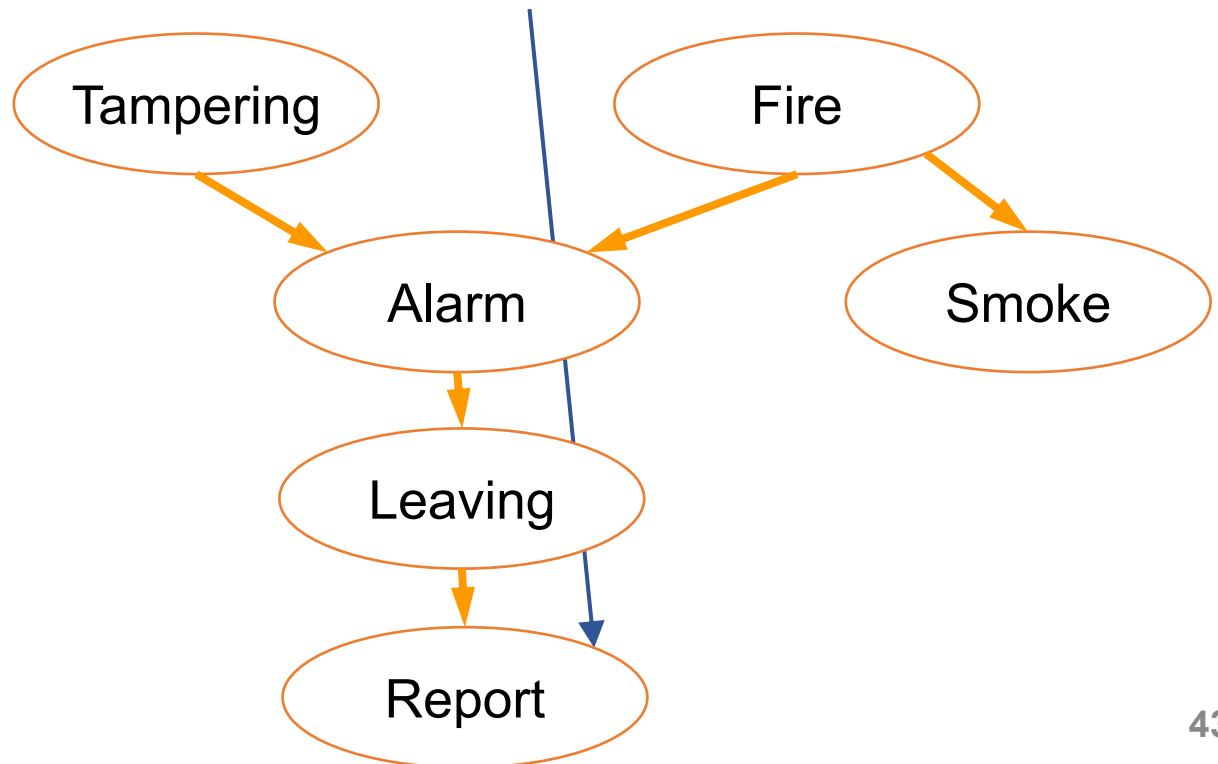
$$P(F) P(T) P(A | F, T) P(S | F) \mathbf{P(L | A)} P(R | F, T, A, S, L)$$



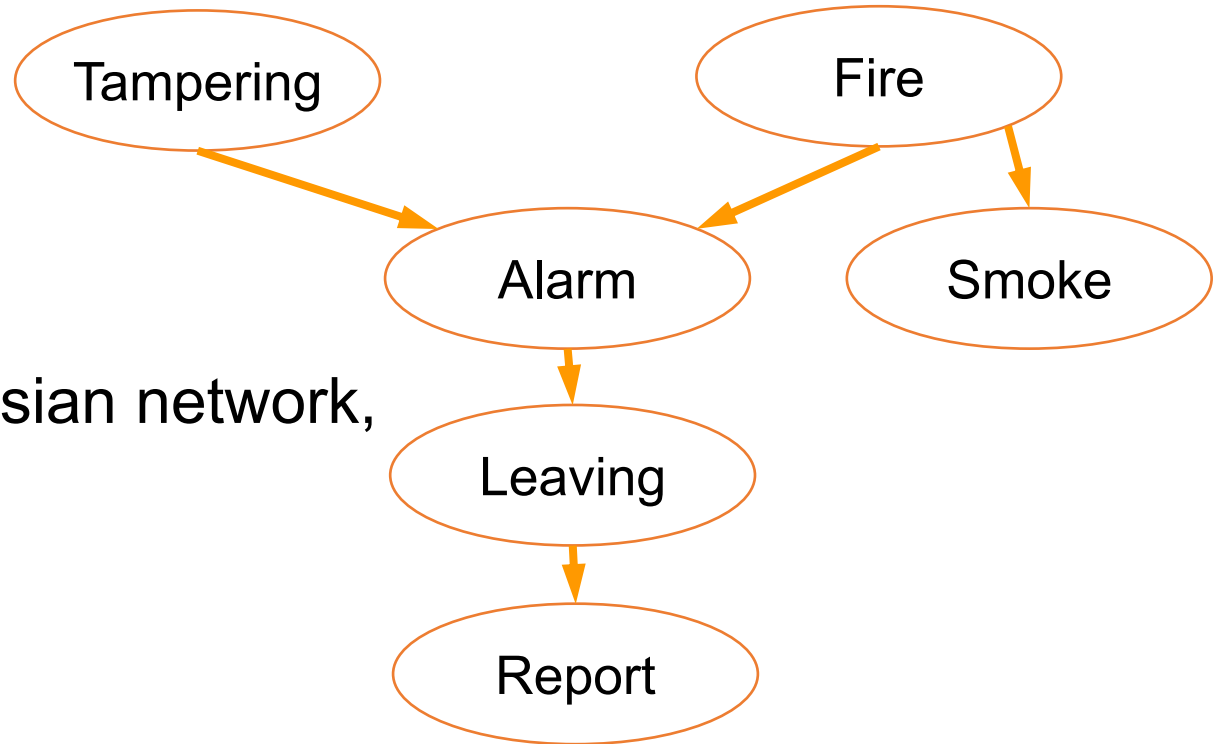
Example: Fire diagnosis (Topology)

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

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



Example: Fire diagnosis (Topology)



- The resulting Bayesian network, and

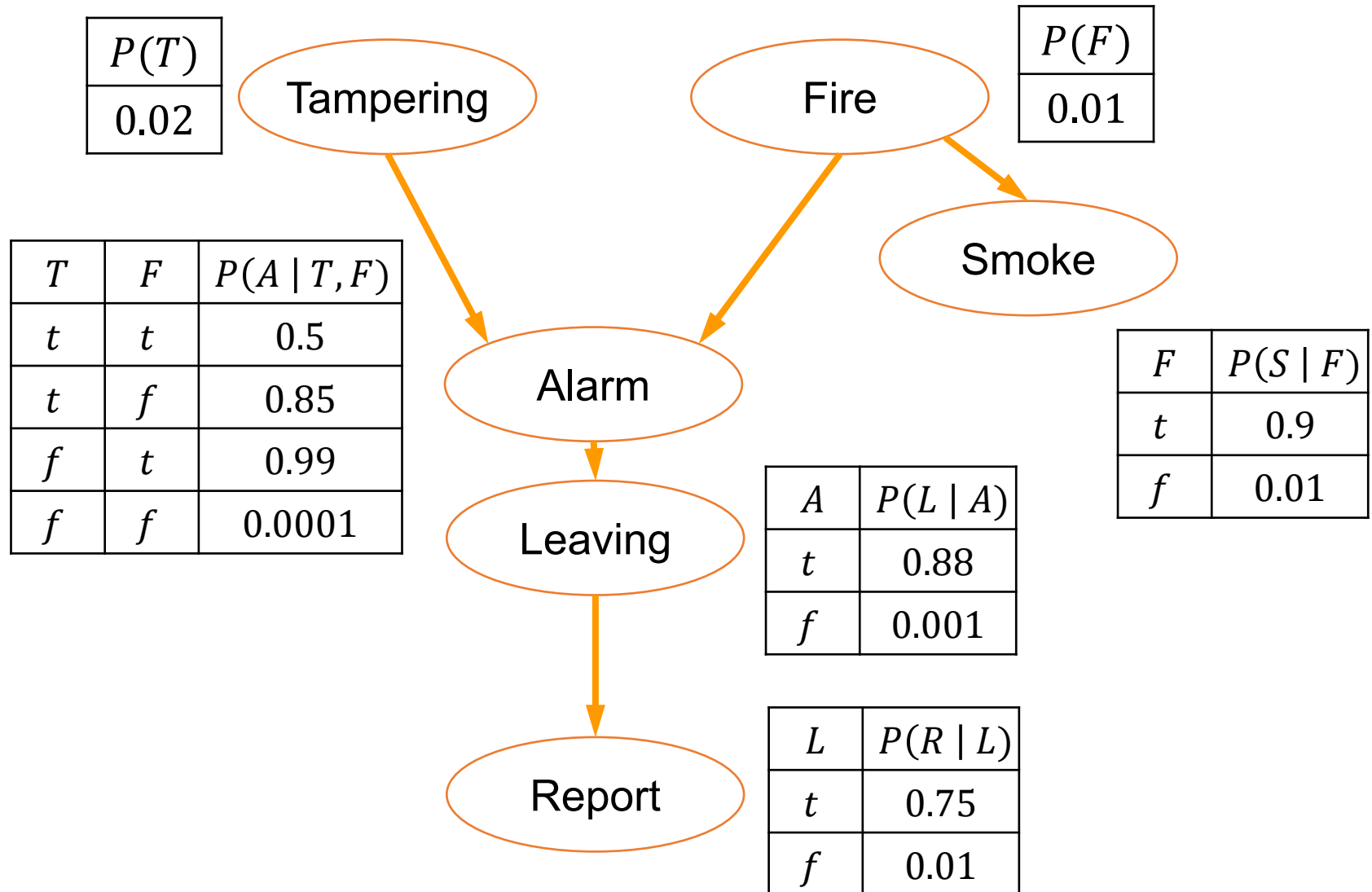
- the corresponding compact factorization of the original JPD

$$P(F, T, A, S, L, R) = P(F) P(T) P(A | F, T) P(S | F) P(L | A) P(R | L)$$

Example: Fire diagnosis (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*?

Example: Fire diagnosis (CPTs)

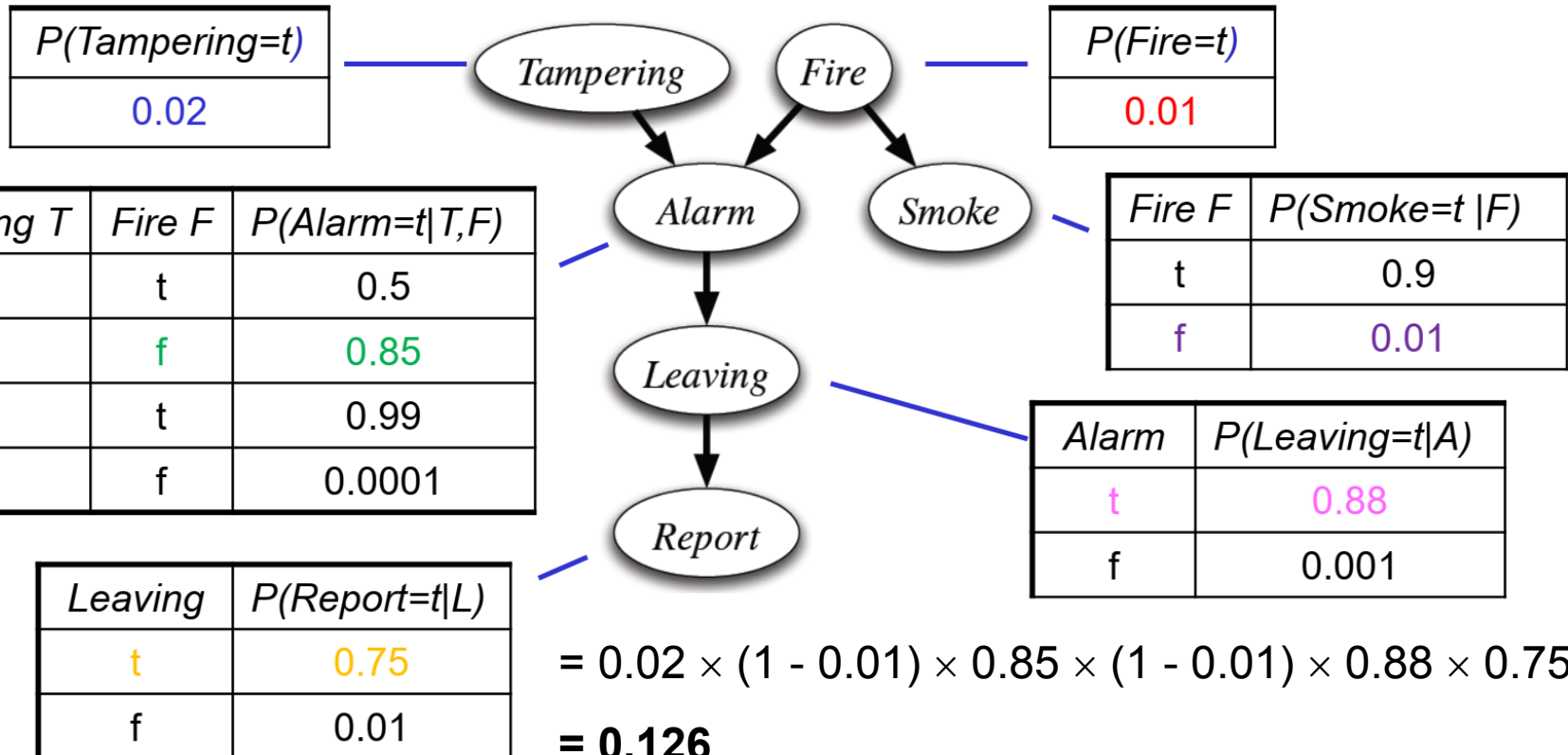


Example: Fire diagnosis (CPTs)

- *How many probabilities do we need to specify for this Bayesian network?*
- $P(\textit{Tampering})$: 1 probability $P(T = t)$
- $P(\textit{Alarm} \mid \textit{Tampering}, \textit{Fire})$: 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: $1+1+4+2+2+2 = \mathbf{12}$ (compared to $2^6-1= \mathbf{63}$ for full JPD!)

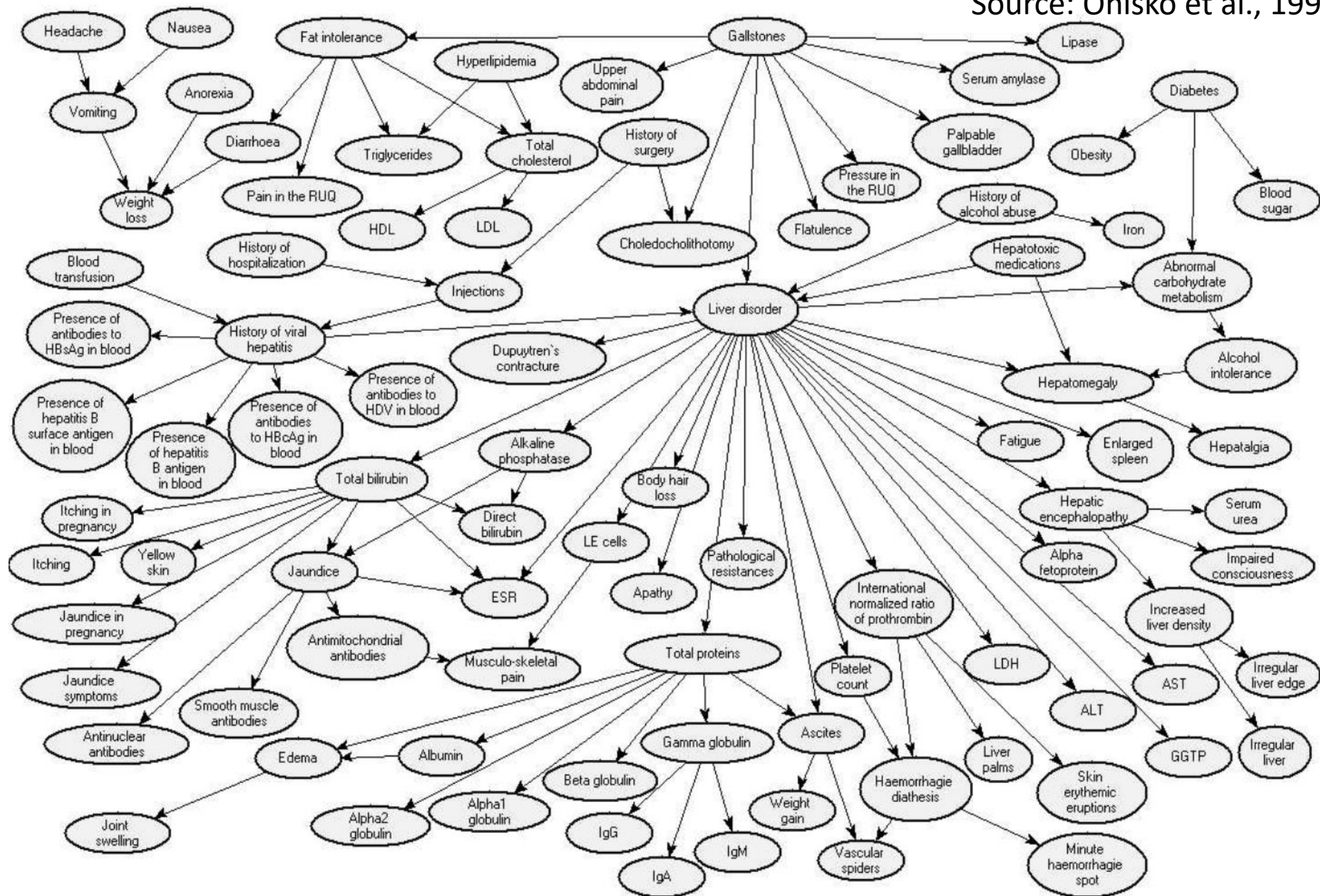
Example: Fire diagnosis

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



Bayesian networks vs. JPD

- 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 JPD
- 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.



~60 nodes, max 4 parents per node

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

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 the full JPD
- Bayesian networks are **useful in sparse domains** (or locally structured domains).
 - That is, domains 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?
 - 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 full JPD



THE END