Independence

Absolute independence :

- Inference from joint distributions: huge space (and thus time) complexity, but
- Two random variables A B are (absolutely) independent iff $P(A|B) = P(A)$, i.e., $P(A,B) = P(A|B)P(B) = P(A)P(B)$, and
- If n Boolean variables are independent, the full joint is $\mathbb{P}(X_1, \ldots, X_n) = \prod_i \mathbb{P}(X_i)$, i.e., can be specified by just n numbers; but
- Absolute independence is a very strong requirement, rarerly met, so:

Relative independence :

- If I have a cavity, the probability that the probe catches does not depend on whether I have a toothache:

$$P(Catch|Toothache, Cavity) = P(Catch|Cavity)$$

  i.e., $Catch$ is conditionally independent of $Toothache$ given $Cavity$.

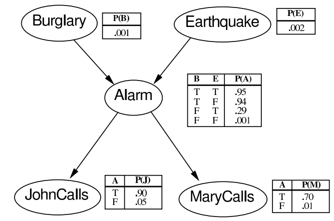
- The same independence holds if I haven't got a cavity:

$$P(Catch|Toothache, \neg Cavity) = P(Catch|\neg Cavity)$$

Belief networks

- A simple, graphical notation for conditional independence assertions and hence for compact specification of full joint distributions.
  - a set of nodes, one per variable
  - a directed, acyclic graph (of "direct influences")
  - a conditional distribution for each node given its parents: $\mathbb{P}(X_i|Parents(X_i))$
  - In the simplest case, conditional distribution represented as a conditional probability table.

I'm at work, neighbor John calls to say my alarm is ringing, but neighbor Mary doesn't call. Sometimes the alarm is set off by minor earthquakes. Is there a burglar?



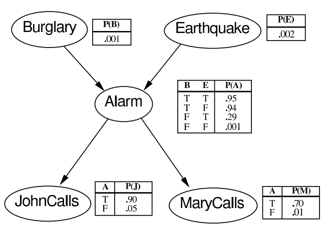
Belief networks (cont'd)

- A belief network provides a complete description of the domain; if X_j is not a parent of X_i then they are conditionally independent, thus:

$$\mathbb{P}(X_i|X_1, \ldots, X_{i-1}) = \mathbb{P}(X_i|Parents(X_i))$$

- More compact than a matrix, so we solve the space problem.
- Computing probabilities:

$$P(J \wedge M \wedge A \wedge \neg B \wedge \neg E) =$$



Belief networks (cont'd)

- A belief network provides a complete description of the domain; if X_j is not a parent of X_i then they are conditionally independent, thus:

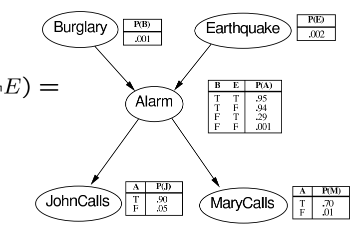
$$\mathbb{P}(X_i|X_1, \ldots, X_{i-1}) = \mathbb{P}(X_i|Parents(X_i))$$

- More compact than a matrix, so we solve the space problem.
- Computing probabilities:

$$P(J \wedge M \wedge A \wedge \neg B \wedge \neg E) =$$

$$P(J|A)P(M|A)P(A|\neg B, \neg E)P(\neg B)P(\neg E) =$$

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



Incremental construction of belief networks

- A belief network is a correct representation of the domain only if each node is conditionally independent of it's predecessors (in node ordering), given its parents.
  - e.g., the fact that Mary calls certainly depends on whether there is a burglary, but is not directly influenced by it (influenced only by the alarm sounding or not).

$$\mathbb{P}(M|J, A, E, B) = \mathbb{P}(M|A)$$

in general,

$$\mathbb{P}(X_i|X_1, \ldots, X_{i-1}) = \mathbb{P}(X_i|Parents(X_i))$$

- Incremental construction:
  1. Choose the set of variables $\mathbf{X}$ that describes the domain.
  2. Choose an ordering $\langle X_1, X_2, \ldots, X_n\rangle$ for $\mathbf{X}$.
  3. For i from 1 to n do
     (a) add a node for X_i to the network.
     (b) choose as parents for this node some minimal set of nodes such that it holds that $\mathbb{P}(X_i|X_1, \ldots, X_{i-1}) = \mathbb{P}(X_i|Parents(X_i))$.

Incremental construction (cont'd)

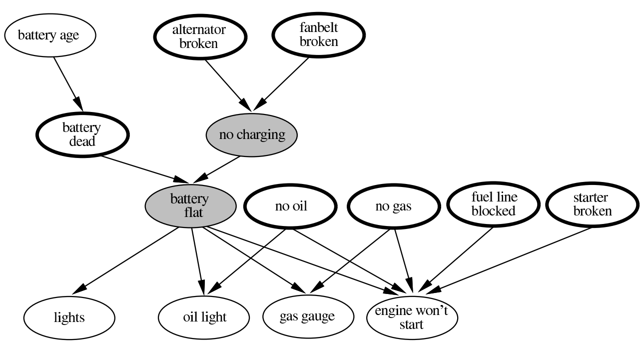
- The node ordering does matter.
  - Compare the orderings

B, E, A, J, M	original construction
M, J, A, B, E	two more edges
M, J, E, B, A	same complexity as the full joint distribution!!

  - All the above networks represent the same joint distribution, one better than the others.
- The correct order of nodes is to cosider the "root causes" first, then the variables they influence directly, and so on.

Hidden varialbles

- Initial evidence: engine won't start
- Testable variables (thin ovals)
- Diagnosis variables (thick ovals)
- Hidden variables (shaded) ensure sparse structure, reduce parameters



Exact inference in belief networks

- Simple queries:
  compute posterior marginal $\mathbb{P}(X_i|\mathbf{E} = \mathbf{e})$.
  - e.g., $P(NoGas|Gauge = empty, Lights = on, Starts = false)$.
- Inference by enumeration: rewrite full joint entries using products of entries in the node tables.
  Simple query on the burglary network:

$$\begin{aligned}\mathbb{P}(B|J = true, M = true) &= \mathbb{P}(B, J = true, M = true)/P(J = true, M = true)\\ &= \alpha\mathbb{P}(B, J = true, M = true)\\ &= \alpha\sum_e\sum_a \mathbb{P}(B, e, a, J = true, M = true)\end{aligned}$$

Rewrite full joint entries using product of CPT entries:

$$\begin{aligned}&P(B = true|J = true, M = true)\\ &= \alpha\sum_e\sum_a P(B = true)P(e)P(a|B = true, e)P(J = true|a)P(M = true|a)\\ &= \alpha P(B = true)\sum_e P(e)\sum_a P(a|B = true, e)P(J = true|a)P(M = true|a)\end{aligned}$$

Inference by enumeration

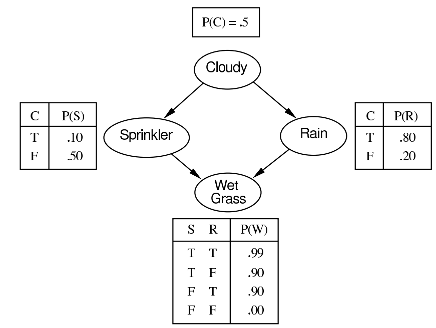
function ENUMERATIONASK(X,e,bn) **returns** a distribution over X
inputs: X, the query variable
  e, evidence specified as an event
  bn, a belief network specifying joint distribution $\mathbb{P}(X_1, \ldots, X_n)$

 $Q(X) \leftarrow$ a distribution over X
 for each value x_i of X **do**
  extend e with value x_i for X
  $Q(x_i) \leftarrow$ ENUMERATEALL(VARS[bn],e)
 return NORMALIZE($Q(X)$)

ENUMERATEALL($vars$,e) **returns** a real number
 if EMPTY?($vars$) **then return** 1.0
 else do
  $Y \leftarrow$ FIRST($vars$)
  if Y has value y in e
   then return $P(y \mid Parents(Y)) \times$ ENUMERATEALL(REST($vars$),e)
   else return $\sum_y P(y \mid Parents(Y)) \times$ ENUMERATEALL(REST($vars$),e_y)
    where e_y is e extended with $Y = y$

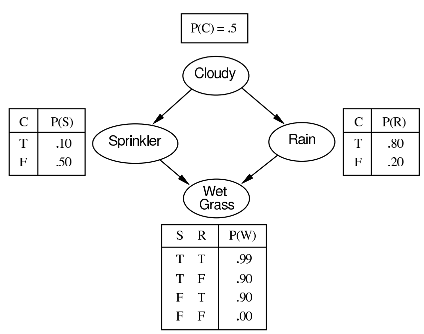
The complexity of exact inference

- For polytrees (at most one path between any two nodes): linear in the size of the network
- For multiply connected networks (dags): exponential!



The complexity of exact inference

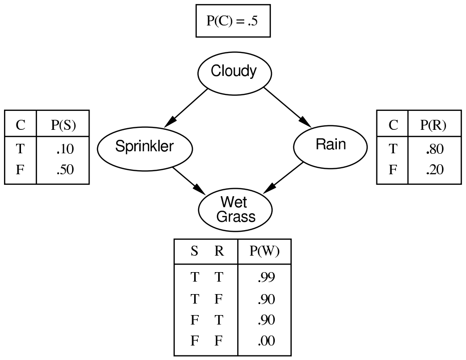
- For polytrees (at most one path between any two nodes): linear in the size of the network
- For multiply connected networks (dags): exponential!
  - special case: inference in propositional logic
  - so it is NP-hard



Clustering algorithms

- Variable elimination is simple and efficient
- It can be however less efficient than possible in multiply connected networks (repeat computations)
- Improvement: clustering
  - Basic idea: join individual nodes so that the network becomes a polytree
  - Example: two nodes with boolean variables are replaced by a “meganode” with one variable that can take the values tt, tf, ft, ff.

Clustering algorithms (cont'd)



- $Sprinkler + Rain$:

C	$P(S+R)$ tt	tf	ft	ff
t	.08	.02	.72	.18
f	.10	.40	.10	.40

- $Wet\ grass$:

$S+R$	$P(W)$
$t\ t$	.99
$t\ f$	.90
$f\ t$	.90
$f\ f$	.00

Clustering algorithms (cont'd)

- Meganodes can have shared variables
- A special purpose inference algorithm is needed
  - Takes a form similar to constraint propagation
  - Linear time (with careful bookkeeping)
  - Still an NP-hard problem though