Introduction to Artificial Intelligence

Bayesian Networks

Jianmin Li

Department of Computer Science and Technology
Tsinghua University

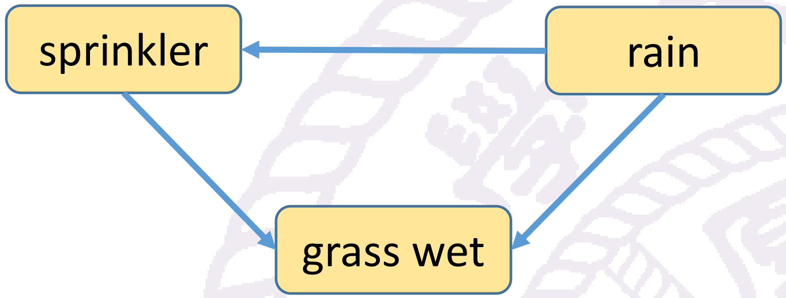
Spring, 2024

Outline

- Syntax
- Semantics
- Parameterized distributions
- Exact inference

Bayesian networks

- A simple, graphical notation for conditional independence assertions
- Compact specification of full joint distributions

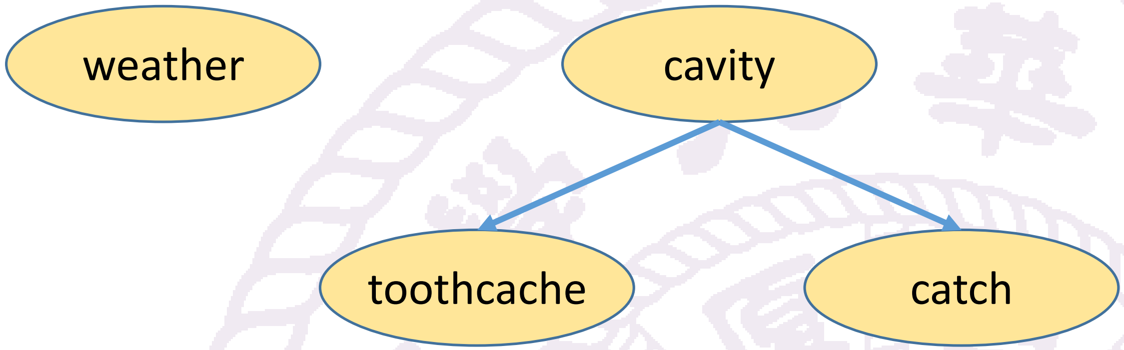


- Syntax:
  - a set of nodes, one per variable
  - a directed, acyclic graph (link "directly influences")
  - a conditional distribution for each node given its parents

$$\mathbf{P}(X_i \mid Parents(X_i))$$

Topology

- Topology of network encodes conditional independence assertions



- *Weather* is independent of the other variables
- *Toothache* and *Catch* are conditionally independent given *Cavity*

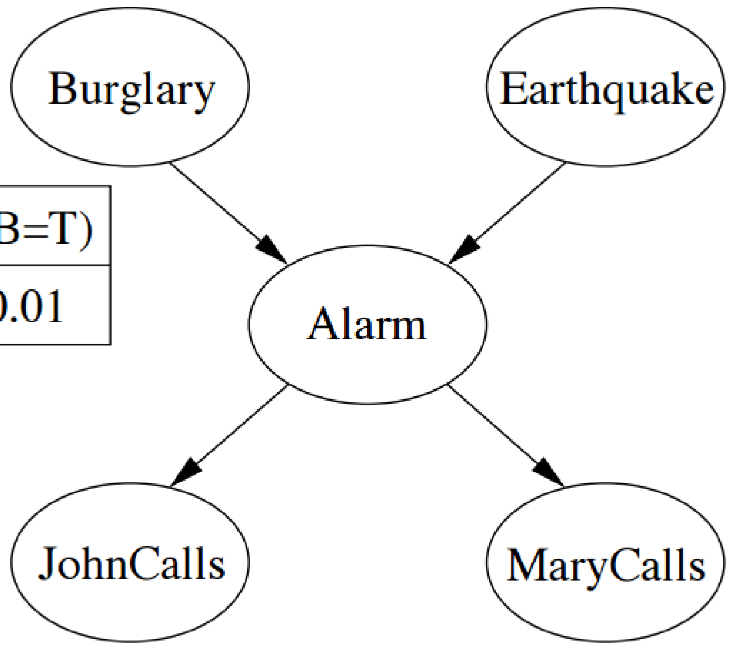
Conditional distribution

- Conditional probability table (CPT)
  - giving the distribution over X_i for each combination of parent values
- Canonical distributions

Example

- I’m at work, neighbor John calls to say my alarm is ringing, but neighbor Mary doesn’t call. Sometimes it’s set off by minor earthquakes. Is there a burglar?
- Variables: *Burglary*, *Earthquake*, *Alarm*, *JohnCalls*, *MaryCalls*
- Network topology reflects "causal" knowledge:
  - A burglar can set the alarm off
  - An earthquake can set the alarm off
  - The alarm can cause Mary to call
  - The alarm can cause John to call

Example



P(E=T)
0.02

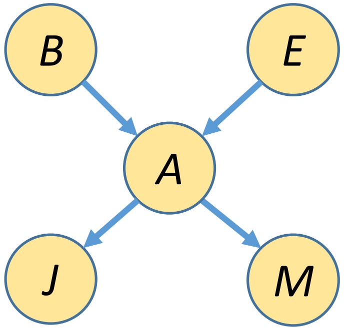
P(B=T)
0.01

B	E	P(A=T|B,E)
T	T	0.95
T	F	0.94
F	T	0.29
F	F	0.001

A	P(J=T|A)
T	0.90
F	0.05

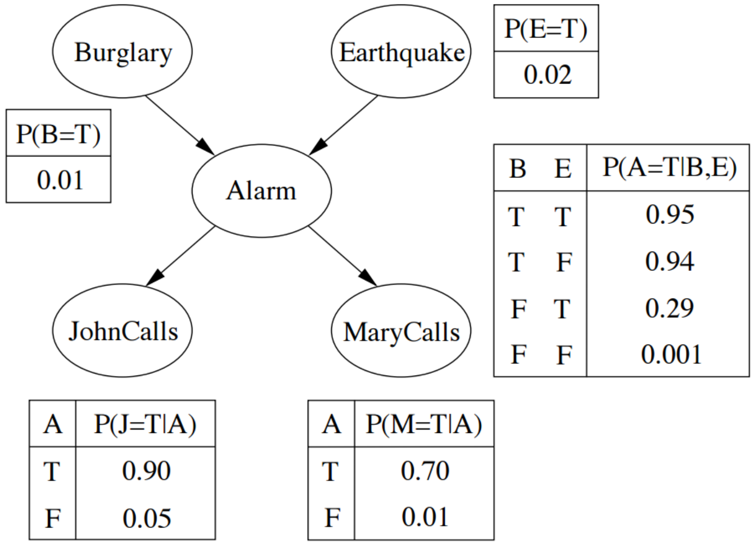
A	P(M=T|A)
T	0.70
F	0.01

Compactness



- A CPT for Boolean X_i with k Boolean parents has 2^k rows for the combinations of parent values
- Each row requires one number p for X_i = true (the number for X_i = false is just $1-p$)
- If each variable has no more than k parents, the complete network requires $O(n*2^k)$ numbers
- I.e., grows linearly with n, vs. $O(2^n)$ for the full joint distribution
- For burglary net, 1+1+4+2+2=10 numbers (vs. $2^5-1 = 31$)

Global semantics



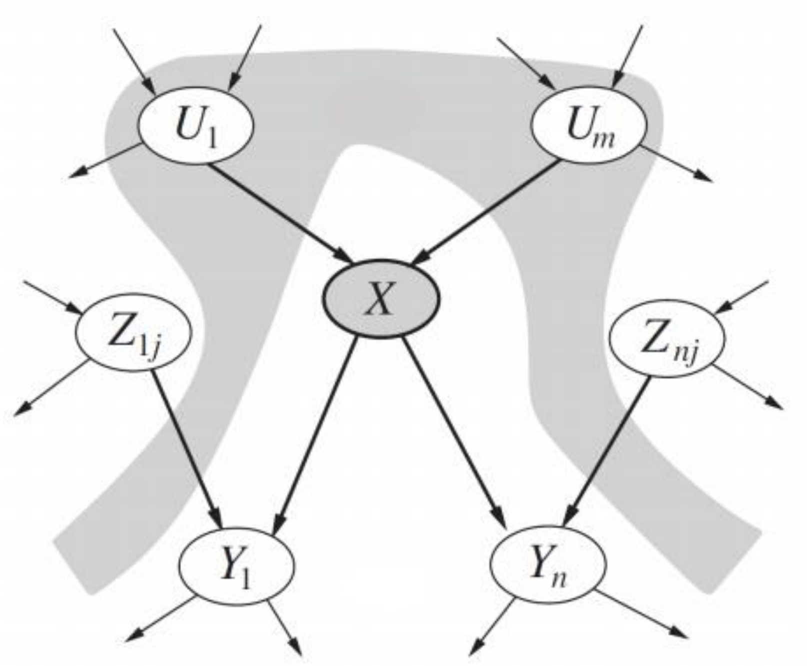
- Defines the full joint distribution as the product of the local conditional distributions

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

- e.g

$$
\begin{aligned}
& P(j \wedge m \wedge a \wedge \neg b \wedge \neg e) \\
= \quad & P(j|a)P(m|a)P(a|\neg b, \neg e)P(\neg b)P(\neg e) \\
= \quad & 0.9 \times 0.7 \times 0.001 \times 0.999 \times 0.998 \\
\approx \quad & 0.000628
\end{aligned}
$$

Local semantics

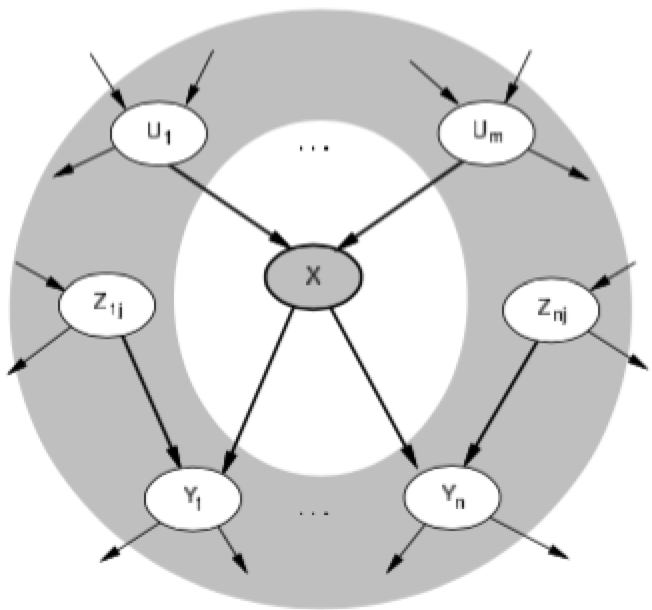


- Each node is conditionally independent of its nondescendants given its parents
- Theorem: Local semantics $\Leftrightarrow$ global semantics

Markov blanket

- Each node is conditionally independent of all others given its Markov blanket

parents + children + children's parents



Constructing Bayesian networks

- A series of locally testable assertions of conditional independence
- Guarantees the required global semantics
- Algorithm
  - Choose an ordering of variables $X_1, \dots, X_n$
  - For *i* = 1 to *n*
    - add X_i to the network
    - select parents from $X_1, \dots, X_{i-1}$ such that
      $\mathbf{P}(X_i | Parents(X_i)) = \mathbf{P}(X_i | X_1, \dots, X_{i-1})$

Constructing Bayesian networks

- The choice of parents guarantees the global semantics:

Chain rule

$$\mathbf{P}(X_1, \ldots, X_n) = \prod_{i=1}^{n} \mathbf{P}(X_i | X_1, \ldots, X_{i-1})$$

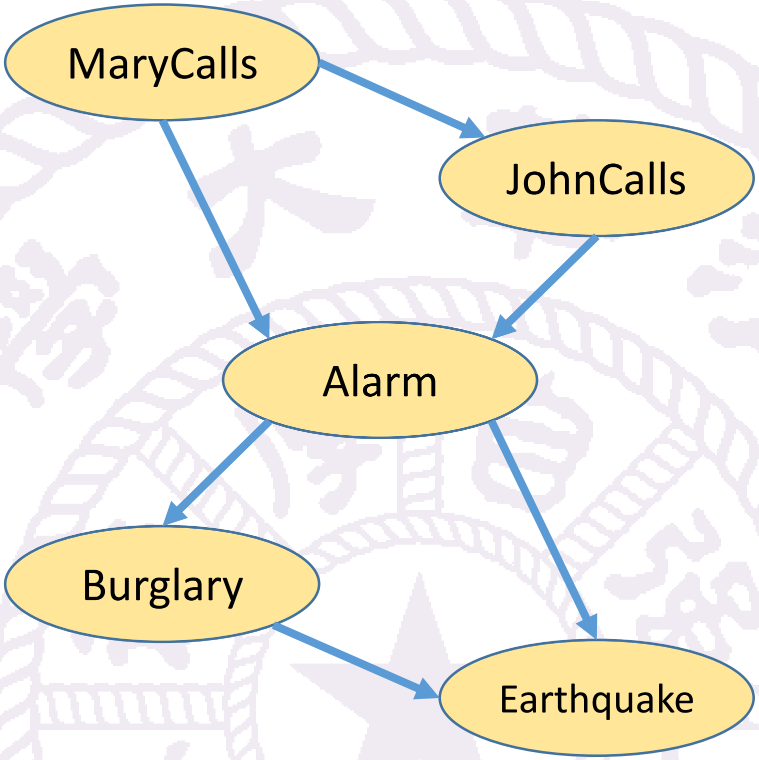
$$= \prod_{i=1}^{n} \mathbf{P}(X_i | Parents(X_i))$$

By construction

Example

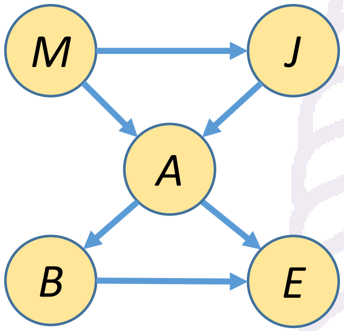
- Suppose we choose the ordering *M, J, A, B, E*

$P(J \mid M)=P(J)$?	No
$P(A \mid J, M)=P(A \mid J)$? $P(A \mid J, M)=P(A)$?	No
$P(B \mid A, J, M)=P(B \mid A)$?	Yes
$P(B \mid A, J, M)=P(B)$?	No
$P(E \mid B, A, J, M)=P(E \mid A, B)$?	Yes
$P(E \mid B, A, J, M)=P(E \mid A)$?	No

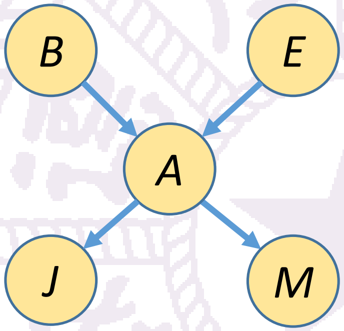


Example

- Deciding conditional independence is hard in noncausal directions
- Assessing conditional probabilities is hard in noncausal directions
- Network is less compact: 1+2+4+2+4=13 numbers needed

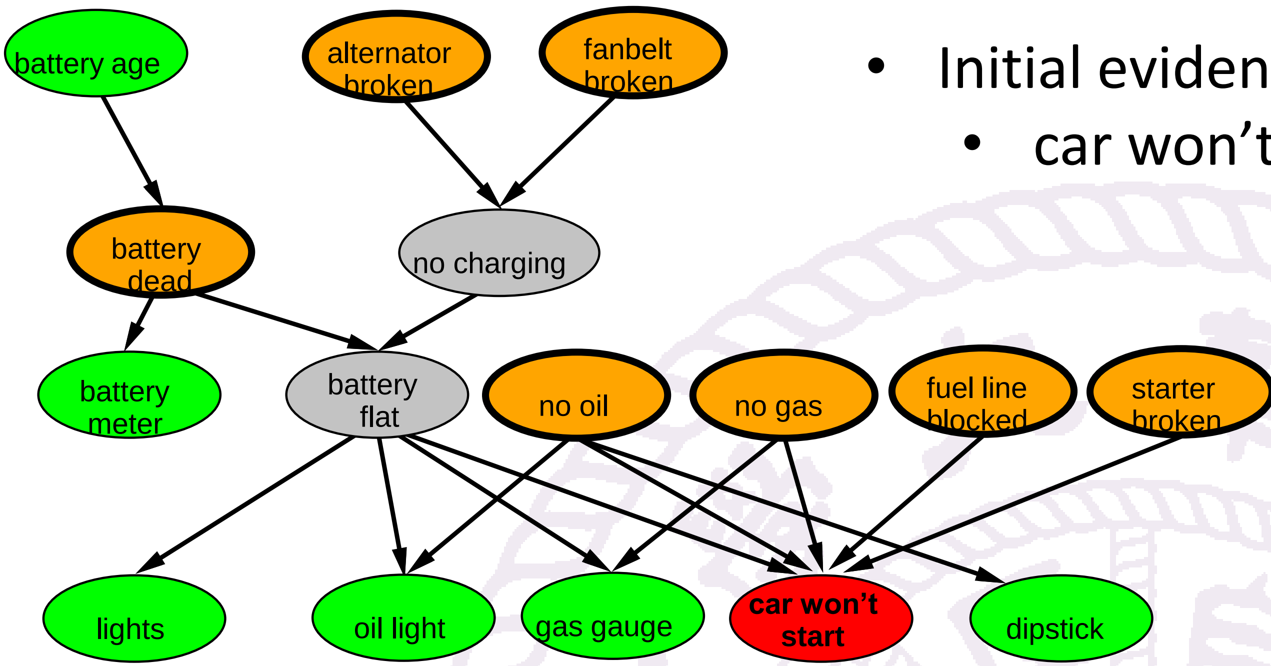


M, J, A, B, E



B, E, A, J, M

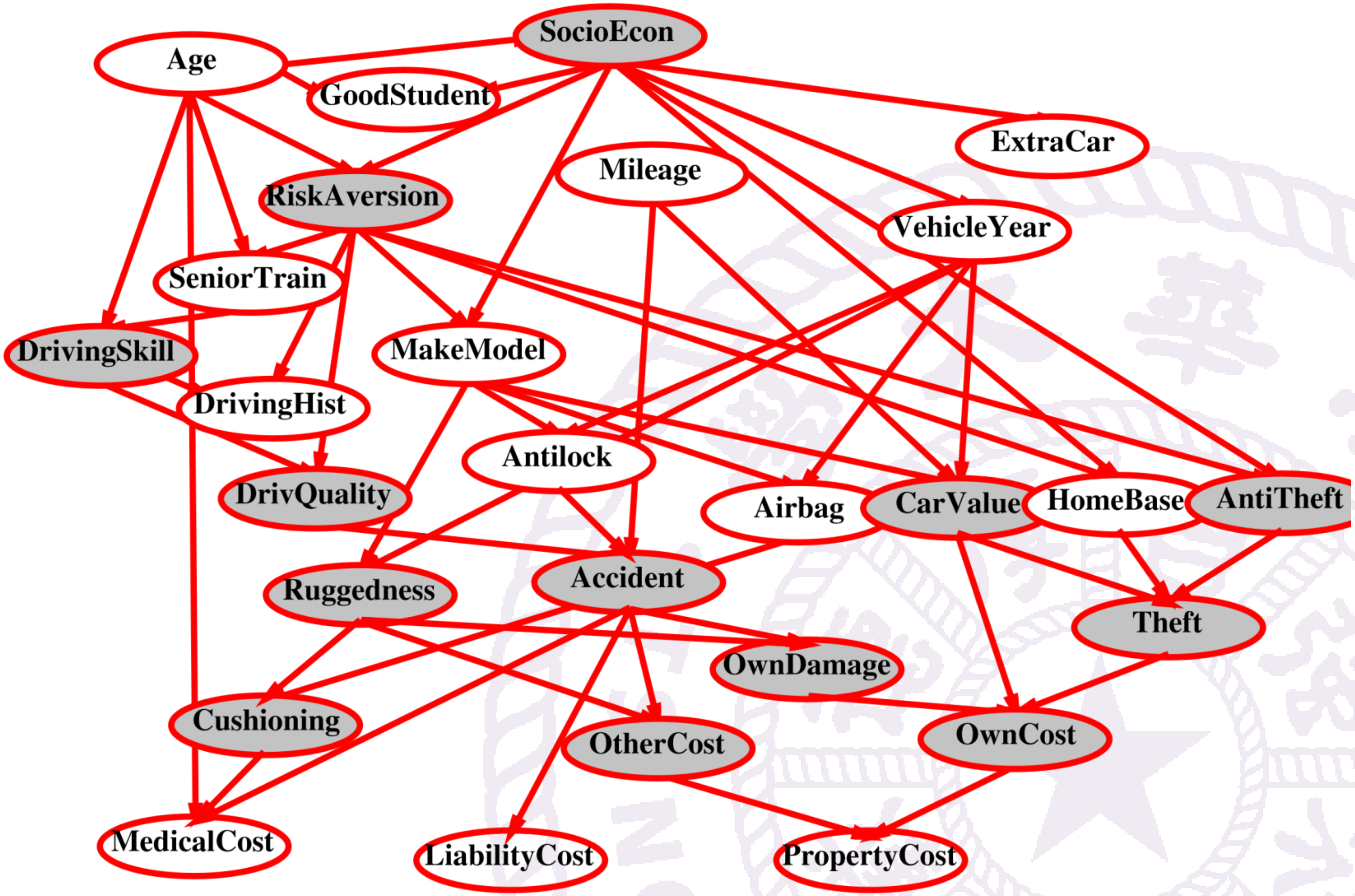
Example: Car diagnosis



- Initial evidence
  - car won't start

- Testable variables (green)
- "broken, so fix it" variables (orange)
- Hidden variables (gray)
  - ensure sparse structure, reduce parameters

Example: Car insurance



Compact conditional distributions

- CPT
  - grows exponentially with number of parents
  - becomes infinite with continuous-valued parent or child
- Solution
  - canonical distributions that are defined compactly

Deterministic models

- the simplest case

$X = f(Parents(X))$ for some function f

- e.g., Boolean functions

$NorthAmerican \Leftrightarrow Candian \vee US \vee Mexican$

- e.g., numerical relationships among continuous variables

$$\frac{\partial Level}{\partial t} = inflow + precipitation - outflow - evaporation$$

Noisy-OR distributions

- Multiple noninteracting causes
  - Parents $U_1, \ldots, U_k$ include all causes (can add leak node)
  - Negated $\neg U_i$ causes do not have any influence on X
  - Independent failure probability q_i for each cause alone

$$P(\neg X \mid U_1, \ldots, U_j, \neg U_{j+1}, \ldots, \neg U_k) = \prod_{i=1}^{j} q_i$$

$$P(X \mid U_1, \ldots, U_j, \neg U_{j+1}, \ldots, \neg U_k) = 1 - \prod_{i=1}^{j} q_i$$

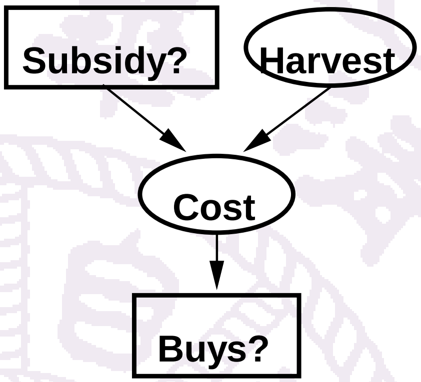
- Number of parameters linear in number of parents

Noisy-OR distributions

Cold	*Flu*	*Malaria*	*P*(*Fever*)	*P*(¬*Fever*)
F	F	F	0.0	1.0
F	F	T	0.9	0.1
F	T	F	0.8	0.2
F	T	T	0.98	0.02=0.2x0.1
T	F	F	0.4	0.6
T	F	T	0.94	0.06=0.6x0.1
T	T	F	0.88	0.12=0.6x0.2
T	T	T	0.988	0.012=0.6x0.2x0.1

Hybrid (discrete+continuous) networks

- Discrete (Subsidy? and Buys?)
- Continuous (Harvest and Cost)
- Option 1: discretization—possibly large errors, large CPTs
- Option 2: finitely parameterized canonical families
  - Continuous variable, discrete+continuous parents (e.g., Cost)
  - Discrete variable, continuous parents (e.g., Buys?)



Continuous child variables

- One conditional density function for child variable given continuous parents, for each possible assignment to discrete parents
- Linear Gaussian model, e.g.,

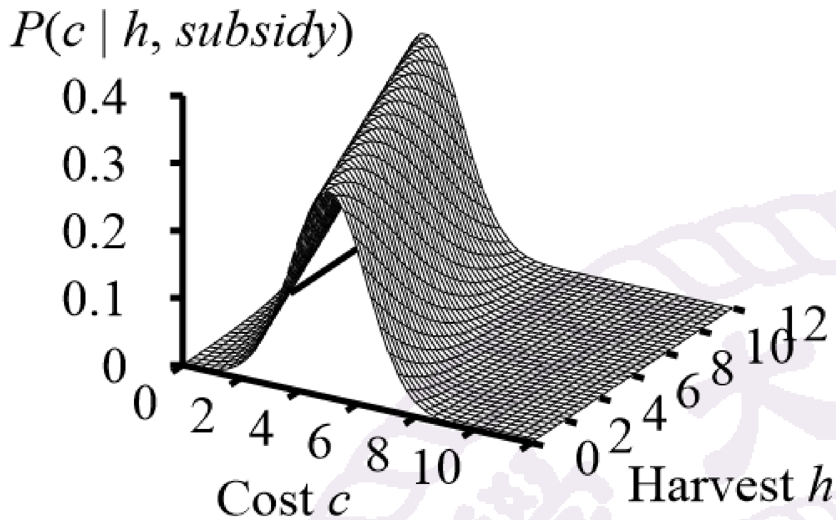
$$p(Cost = c | Harvest = h, Subsidy = true)$$

$$= N(c | a_t h + b_t, \sigma_t)$$

$$= \frac{1}{\sigma_t \sqrt{2\pi}} e^{\left(-\frac{1}{2}\left(\frac{c-(a_t h + b_t)}{\sigma_t}\right)^2\right)}$$

- Mean *Cost* varies linearly with *Harvest*, variance is fixed

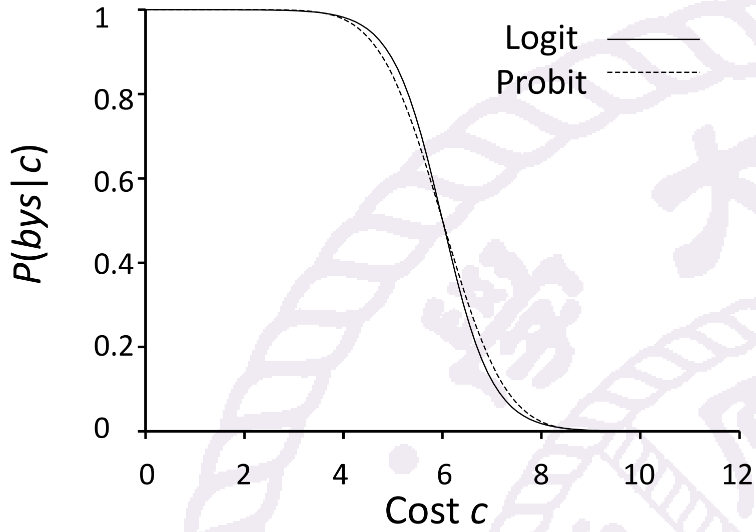
Continuous child variables



- All-continuous network with LG distributions
  - full joint distribution is a multivariate Gaussian
- Discrete+continuous LG network is a conditional Gaussian network
  - a multivariate Gaussian over all continuous variables for each combination of discrete variable values

Discrete variable w/ continuous parents

- Probability of *Buys*? given *Cost* should be a "soft" threshold



- Probit distribution uses integral of Gaussian:

$$\Phi(x) = \int_{-\infty}^{x} N(0,1)(x)dx$$

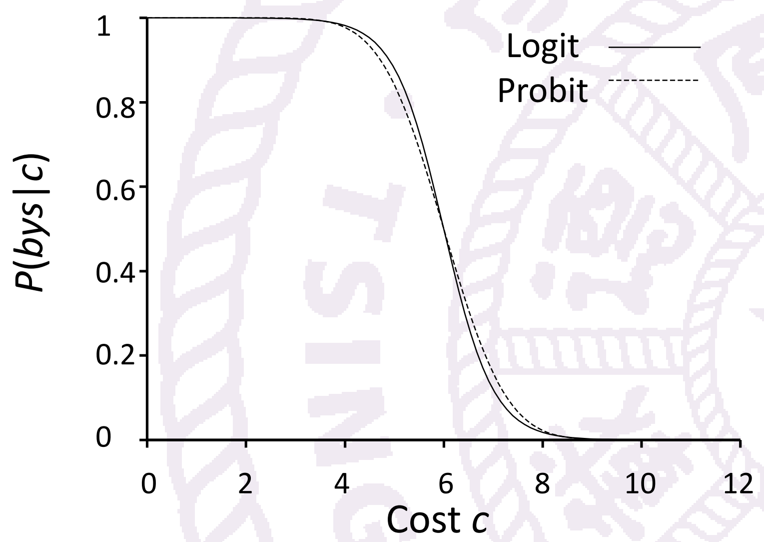
$$P(Buys = true|Cost = c) = \Phi((-c + \mu)/\sigma)$$

Discrete variable w/ continuous parents

- Sigmoid (or logit) distribution also used in neural networks

$$P(Buys = true|Cost = c) = \frac{1}{1 + e^{-2\frac{-c+\mu}{\sigma}}}$$

- Sigmoid has similar shape to probit but much longer tails:



Inference tasks

- 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

$$\mathbf{P}(X_i, X_j | \mathbf{E}=\mathbf{e}) = \mathbf{P}(X_i | \mathbf{E}=\mathbf{e})\mathbf{P}(X_i | X_j, \mathbf{E}=\mathbf{e})$$

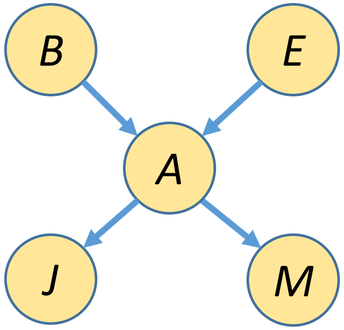
- Optimal decisions
  - decision networks include utility information
  - probabilistic inference required for

$$P(outcome | action, evidence)$$

Inference tasks

- 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



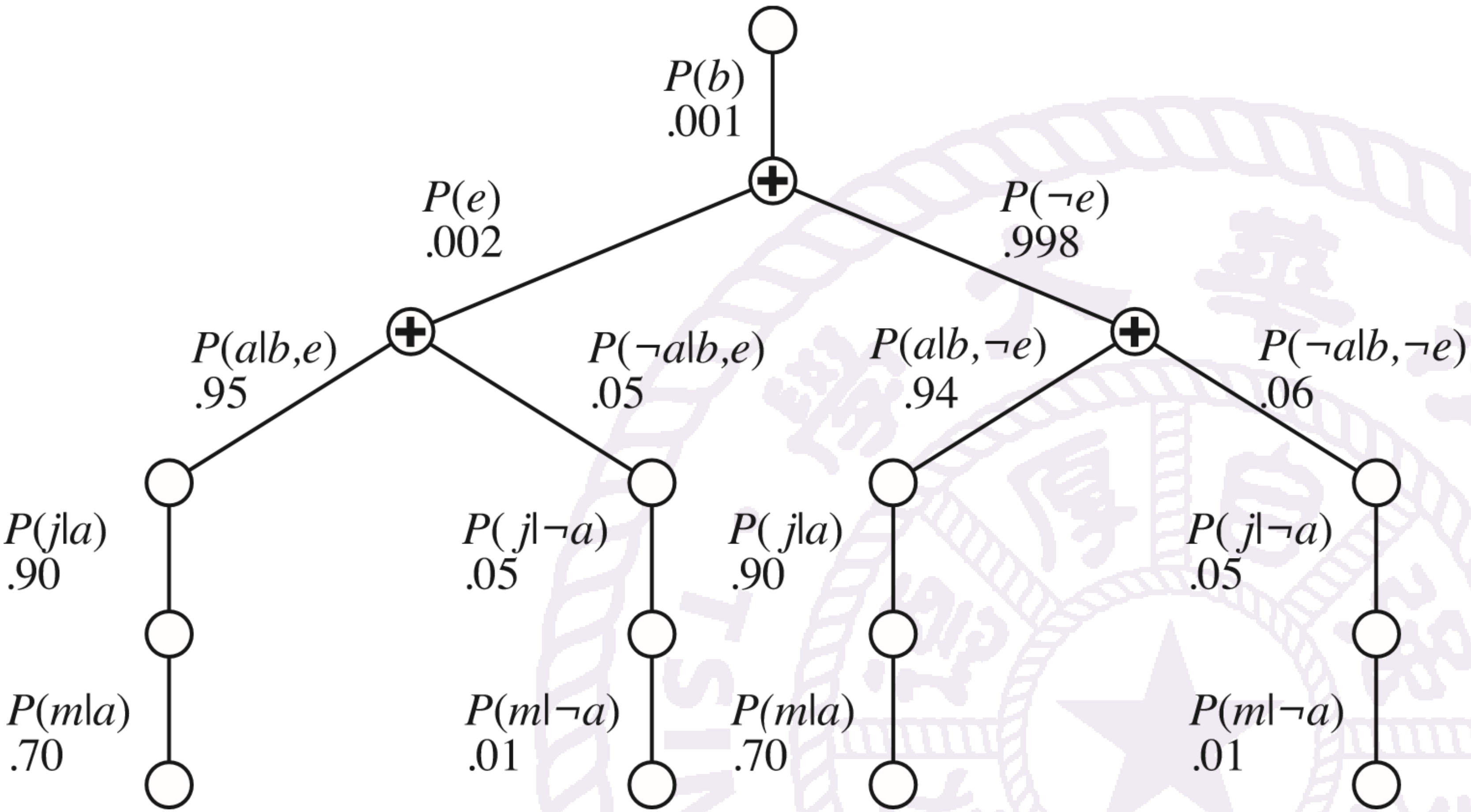
- sum out variables from the joint without actually constructing its explicit representation
- Simple query on the burglary network:

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

- Rewrite full joint entries using product of CPT entries

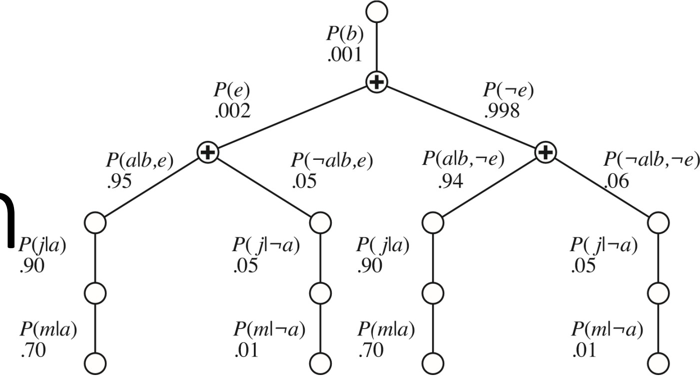
$$\mathbf{P}(B|j,m) = \alpha \sum_e \sum_a \mathbf{P}(B)P(e)\mathbf{P}(a|B,e)P(j|a)P(m|a)$$
$$= \alpha\mathbf{P}(B) \sum_e P(e) \sum_a \mathbf{P}(a|B,e)P(j|a)P(m|a)$$

Evaluation tree



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

Enumeration algorithm



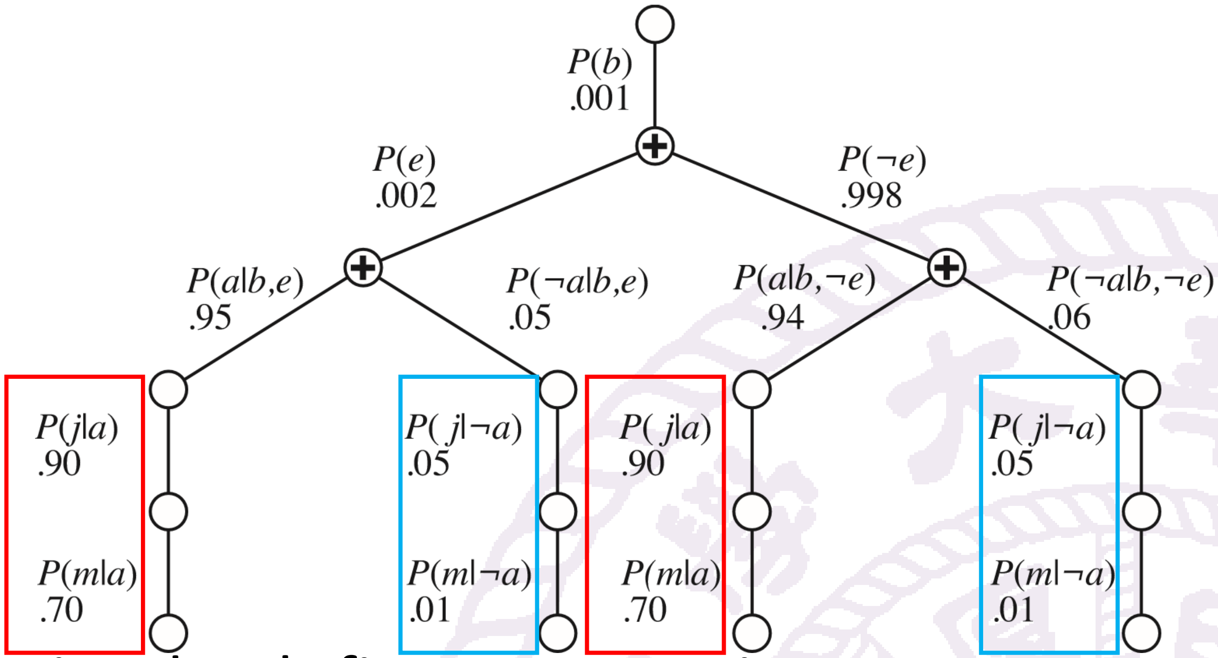
function ENUMERATION-ASK(X, **e**, bn) **returns** a distribution over X
  inputs: X, the query variable
    e, observed values for variables **E**
    bn, a Bayes net with variables $\{X\} \cup \mathbf{E} \cup \mathbf{Y}$ /* **Y** = *hidden variables* */

  $\mathbf{Q}(X) \leftarrow$ a distribution over X, initially empty
  for each value x_i of X **do**
    $\mathbf{Q}(x_i) \leftarrow$ ENUMERATE-ALL(bn.VARS, $\mathbf{e}_{x_i}$)
      where $\mathbf{e}_{x_i}$ is **e** extended with $X = x_i$
  return NORMALIZE($\mathbf{Q}(X)$)

---

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

Inference by enumeration



- Recursive depth-first enumeration
  - $O(n)$ space, $O(d^n)$ time
- Enumeration is inefficient: repeated computation
  - e.g., computes $P(j|a)P(m|a)$ for each value of E

Inference by variable elimination

- Variable elimination: carry out summations right-to-left, storing intermediate results (factors) to avoid recomputation

$$\mathbf{P}(B|j,m) = \underbrace{\alpha\mathbf{P}(B)}_{B} \sum_e \underbrace{P(e)}_{E} \sum_a \underbrace{\mathbf{P}(a|B,e)}_{A} \underbrace{P(j|a)}_{J} \underbrace{P(m|a)}_{M}$$

$$= \alpha\mathbf{P}(B) \sum_e P(e) \sum_a \mathbf{P}(a|B,e) P(j|a) f_M(a)$$

$$= \alpha\mathbf{P}(B) \sum_e P(e) \sum_a \mathbf{P}(a|B,e) f_J(a) f_M(a)$$

$$= \alpha\mathbf{P}(B) \sum_e P(e) \sum_a f_A(a,b,e) f_J(a) f_M(a)$$

sum out *A*

$$= \alpha\mathbf{P}(B) \sum_e f_E(e) f_{\bar{A}JM}(b,e)$$

sum out *E*

$$= \alpha\mathbf{P}(B) f_{\bar{E}\bar{A}JM}(b)$$

$$= \alpha f_B(b) \times f_{\bar{E}\bar{A}JM}(b)$$

Variable elimination: Basic operations

- Pointwise product of factors f_1 and f_2:

$$f_1(x_1, \ldots, x_j, y_1, \ldots, y_k) \times f_2(y_1, \ldots, y_k, z_1, \ldots, z_l)$$
$$= f(x_1, \ldots, x_j, y_1, \ldots, y_k, z_1, \ldots, z_l)$$

- E.g., $f_1(a, b) \times f_2(b, c) = f(a, b, c)$

Example

A	B	$f_1(A, B)$	B	C	$f_2(B, C)$	A	B	C	$f(A, B, C)$
T	T	0.3	T	T	0.2	T	T	T	0.3*0.2
T	F	0.7	T	F	0.8	T	T	F	0.3*0.8
F	T	0.9	F	T	0.6	T	F	T	0.7*0.6
F	F	0.1	F	F	0.4	T	F	F	0.7*0.4
						F	T	T	0.9*0.2
						F	T	F	0.9*0.8
						F	F	T	0.1*0.6
						F	F	F	0.1*0.4

Variable elimination: Basic operations

- 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

$$\sum_X f_1 \times \cdots \times f_k$$
$$= f_1 \times \cdots \times f_i \sum_X f_{i+1} \times \cdots \times f_k$$
$$= f_1 \times \cdots \times f_i \times f_{\bar{X}}$$

assuming $f_1, \ldots, f_i$ do not depend on X

Example

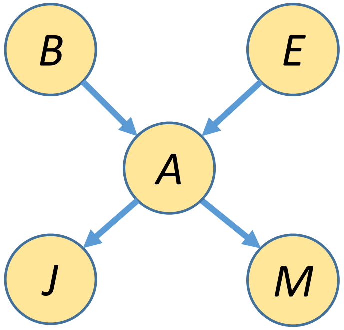
$$
\begin{aligned}
& \mathbf{f}(B, C) \\
= \quad & \sum_a \mathbf{f}_3(A, B, C) \\
= \quad & \mathbf{f}_3(a, B, C) + \mathbf{f}_3(\neg a, B, C) \\
= \quad & \begin{pmatrix} 0.06 & 0.24 \\ 0.42 & 0.28 \end{pmatrix} + \begin{pmatrix} 0.18 & 0.72 \\ 0.06 & 0.04 \end{pmatrix} \\
= \quad & \begin{pmatrix} 0.24 & 0.96 \\ 0.48 & 0.32 \end{pmatrix}
\end{aligned}
$$

Variable elimination algorithm

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 $\mathbf{P}(X_1, \ldots, X_n)$

  $factors \leftarrow [\,]$
  for each var **in** ORDER(bn.VARS) **do**
    $factors \leftarrow$ [MAKE-FACTOR(var, **e**)|$factors$]
    if var is a hidden variable **then** $factors \leftarrow$ SUM-OUT(var, $factors$)
  return NORMALIZE(POINTWISE-PRODUCT($factors$))

Irrelevant variables



- Consider the query **P**(*JohnCalls*|*Burglary* = true)
  $\mathbf{P}(J|b)$
  $= \alpha P(b) \sum_e P(e) \sum_a \mathbf{P}(a|b,e) P(J|a) \sum_m P(m|a)$
- Sum over *m* is identically 1; *M* is irrelevant to the query
- Theorem:

  Y is irrelevant unless $Y \in Ancestors(\{X\} \cup \mathbf{E})$

- *X*=*JohnCalls*, **E**={*Burglary*}, and $Ancestors(\{X\} \cup \mathbf{E}) = \{Alarm, Earthquake\}$, so *MaryCalls* is irrelevant

Complexity of exact inference

- Singly connected networks (or polytrees):
  - any two nodes are connected by at most one (undirected) path
  - Complexity of variable elimination are linear in CPT sizes
- Multiply connected networks:
  - can reduce 3-SAT to exact inference => NP-hard
  - equivalent to counting 3-SAT models => #P-complete

Summary

- Bayes nets provide a natural representation for (causally induced) conditional independence
- Topology + CPTs = compact representation of joint distribution
- Canonical distributions (e.g., noisy-OR) = compact representation of CPTs
- Continuous variables => parameterized distributions (e.g., linear Gaussian)
- Exact inference by variable elimination

谢谢！