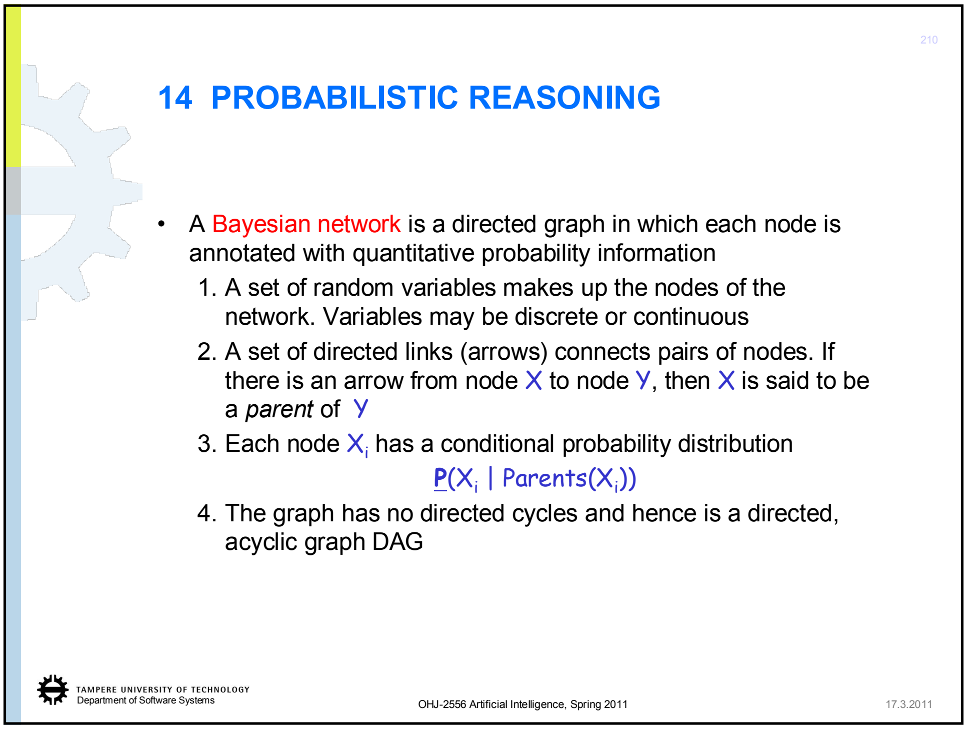
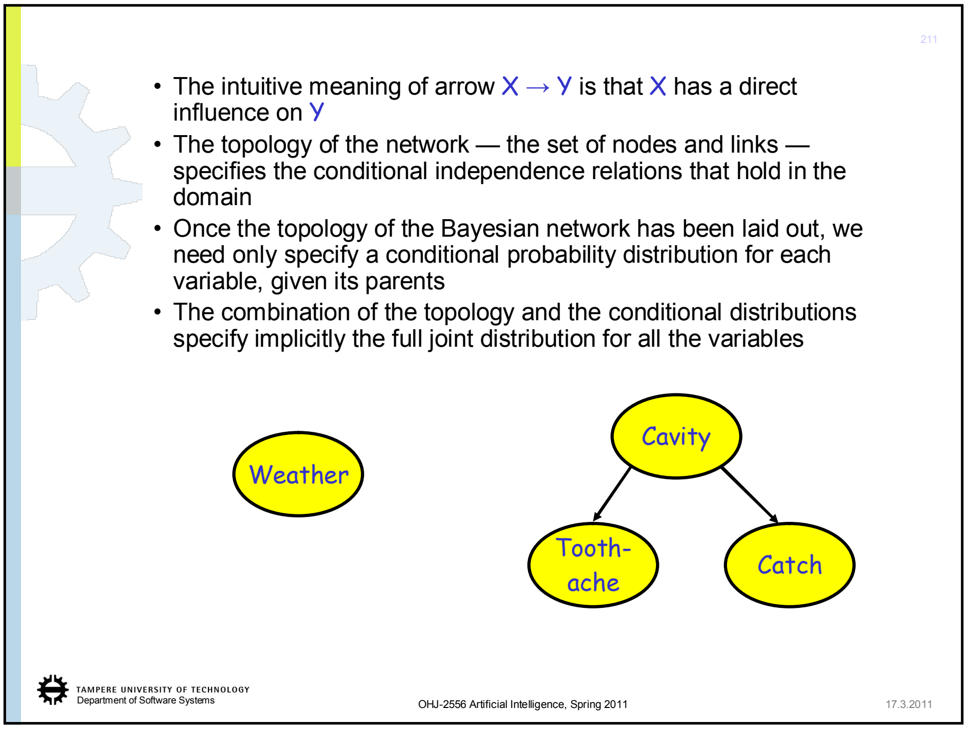


14 PROBABILISTIC REASONING

- A Bayesian network is a directed graph in which each node is annotated with quantitative probability information
  1. A set of random variables makes up the nodes of the network. Variables may be discrete or continuous
  2. A set of directed links (arrows) connects pairs of nodes. If there is an arrow from node X to node Y, then X is said to be a *parent* of Y
  3. Each node X_i has a conditional probability distribution

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

  4. The graph has no directed cycles and hence is a directed, acyclic graph DAG



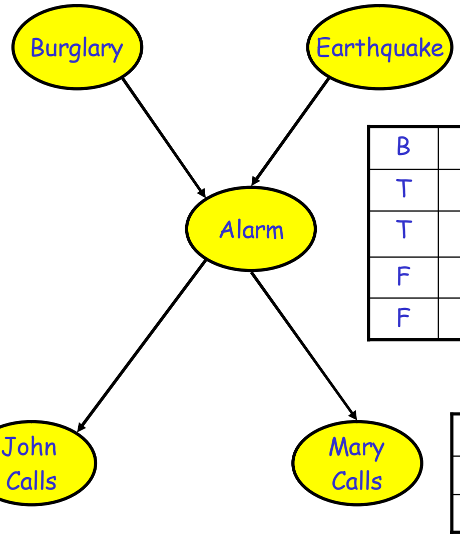
- The intuitive meaning of arrow $X \rightarrow Y$ is that X has a direct influence on Y
- The topology of the network — the set of nodes and links — specifies the conditional independence relations that hold in the domain
- Once the topology of the Bayesian network has been laid out, we need only specify a conditional probability distribution for each variable, given its parents
- The combination of the topology and the conditional distributions specify implicitly the full joint distribution for all the variables

Example (from LA)

- A new burglar alarm has been installed at home. It is fairly reliable at detecting a burglary, but also responds on occasion to minor earthquakes
- Neighbors John and Mary have promised to call you at work when they hear the alarm
- John always calls when he hears the alarm, but sometimes confuses the telephone ringing with the alarm and calls then, too
- Mary, on the other hand, likes loud music and sometimes misses the alarm altogether
- Given the evidence of who has or has not called, we would like to estimate the probability of a burglary

P(B)
.001

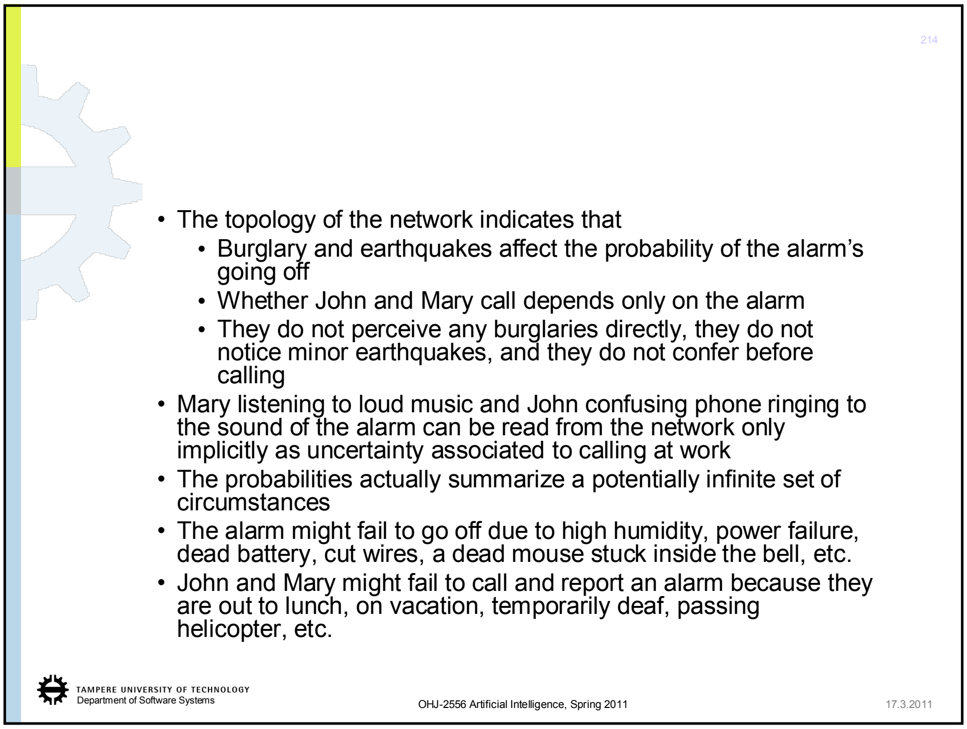
P(E)
.002



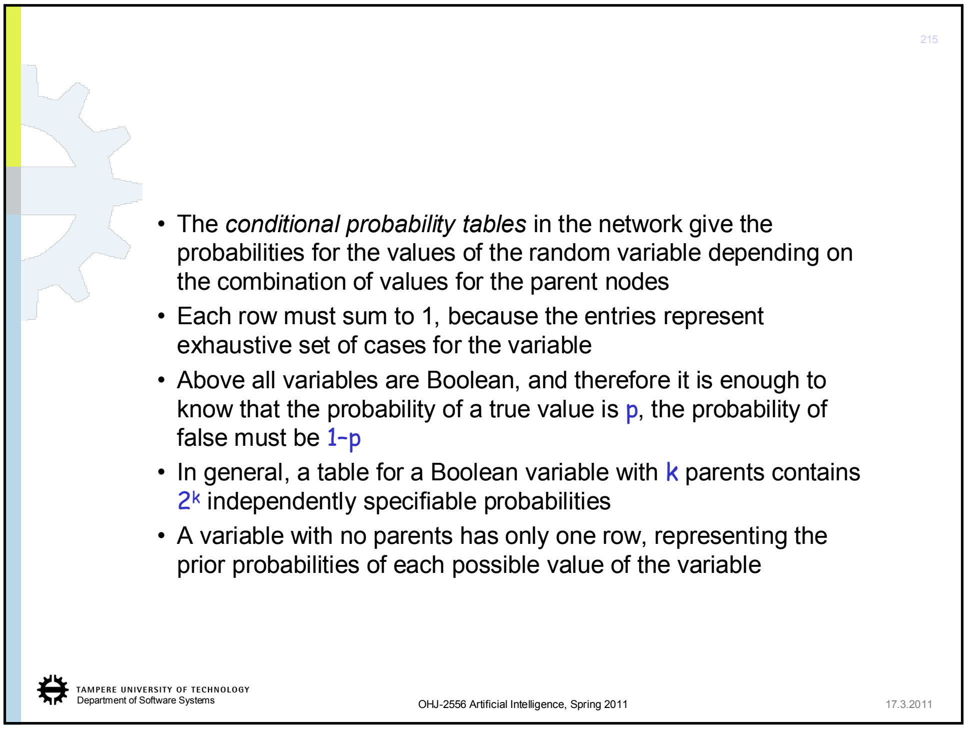
B	E	P(A)
T	T	.95
T	F	.94
F	T	.29
F	F	.001

A	P(J)
T	.90
F	.05

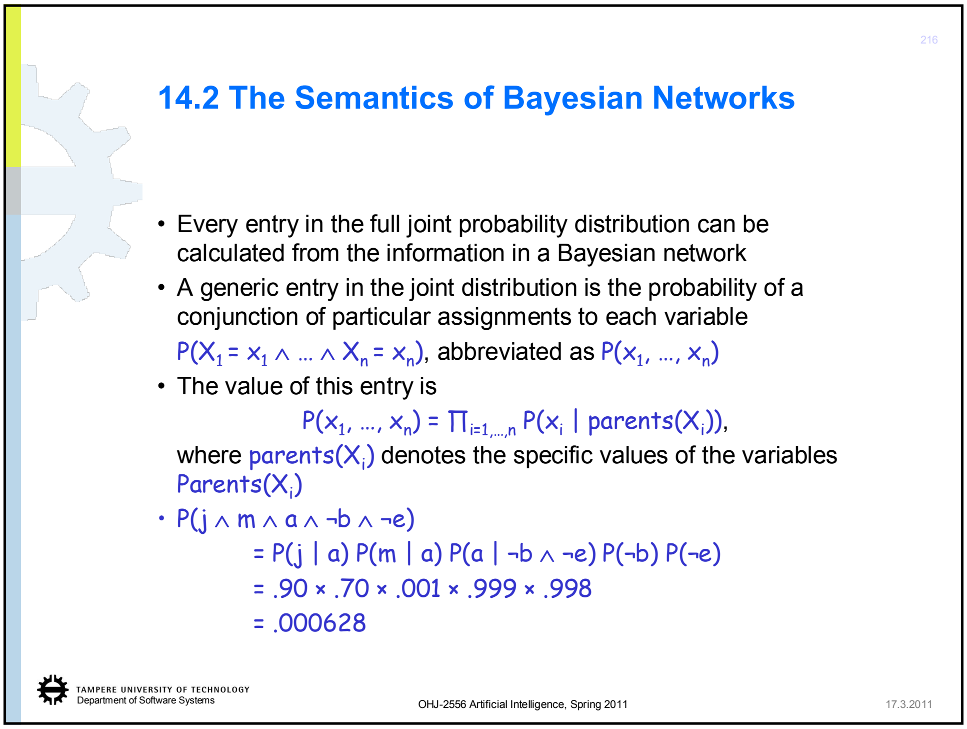
A	P(M)
T	.70
F	.01



- The topology of the network indicates that
  - Burglary and earthquakes affect the probability of the alarm's going off
  - Whether John and Mary call depends only on the alarm
  - They do not perceive any burglaries directly, they do not notice minor earthquakes, and they do not confer before calling
- Mary listening to loud music and John confusing phone ringing to the sound of the alarm can be read from the network only implicitly as uncertainty associated to calling at work
- The probabilities actually summarize a potentially infinite set of circumstances
- The alarm might fail to go off due to high humidity, power failure, dead battery, cut wires, a dead mouse stuck inside the bell, etc.
- John and Mary might fail to call and report an alarm because they are out to lunch, on vacation, temporarily deaf, passing helicopter, etc.



- The *conditional probability tables* in the network give the probabilities for the values of the random variable depending on the combination of values for the parent nodes
- Each row must sum to 1, because the entries represent exhaustive set of cases for the variable
- Above all variables are Boolean, and therefore it is enough to know that the probability of a true value is p, the probability of false must be $1-p$
- In general, a table for a Boolean variable with k parents contains 2^k independently specifiable probabilities
- A variable with no parents has only one row, representing the prior probabilities of each possible value of the variable



14.2 The Semantics of Bayesian Networks

- Every entry in the full joint probability distribution can be calculated from the information in a Bayesian network
- A generic entry in the joint distribution is the probability of a conjunction of particular assignments to each variable $P(X_1 = x_1 \land \ldots \land X_n = x_n)$, abbreviated as $P(x_1, \ldots, x_n)$
- The value of this entry is

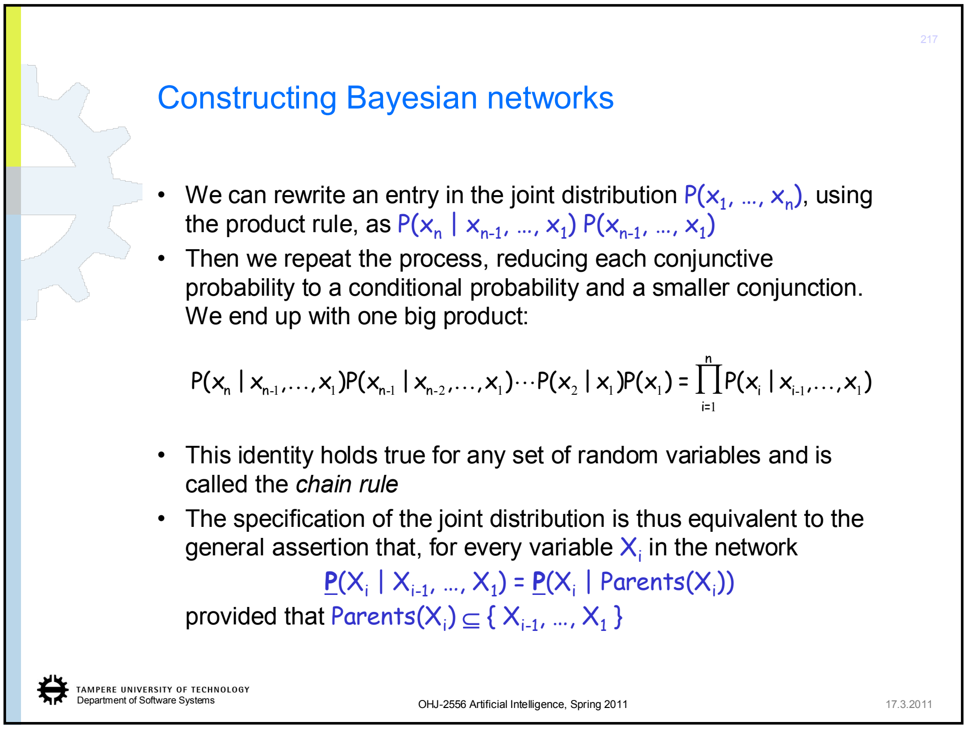
  $$P(x_1, \ldots, x_n) = \prod_{i=1,\ldots,n} P(x_i \mid parents(X_i)),$$

  where $parents(X_i)$ denotes the specific values of the variables $Parents(X_i)$
- $P(j \land m \land a \land \neg b \land \neg e)$

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

  $= .90 \times .70 \times .001 \times .999 \times .998$

  $= .000628$



Constructing Bayesian networks

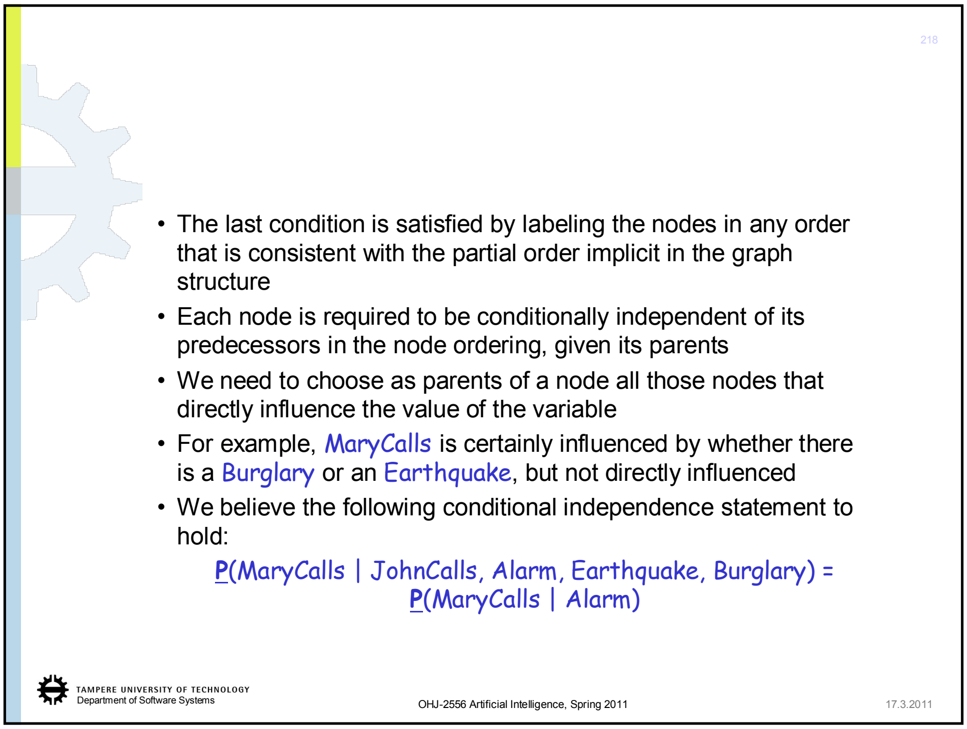
- We can rewrite an entry in the joint distribution $P(x_1, \ldots, x_n)$, using the product rule, as $P(x_n \mid x_{n-1}, \ldots, x_1)\, P(x_{n-1}, \ldots, x_1)$
- Then we repeat the process, reducing each conjunctive probability to a conditional probability and a smaller conjunction. We end up with one big product:

$$P(x_n \mid x_{n-1}, \ldots, x_1) P(x_{n-1} \mid x_{n-2}, \ldots, x_1) \cdots P(x_2 \mid x_1) P(x_1) = \prod_{i=1}^{n} P(x_i \mid x_{i-1}, \ldots, x_1)$$

- This identity holds true for any set of random variables and is called the *chain rule*
- The specification of the joint distribution is thus equivalent to the general assertion that, for every variable X_i in the network

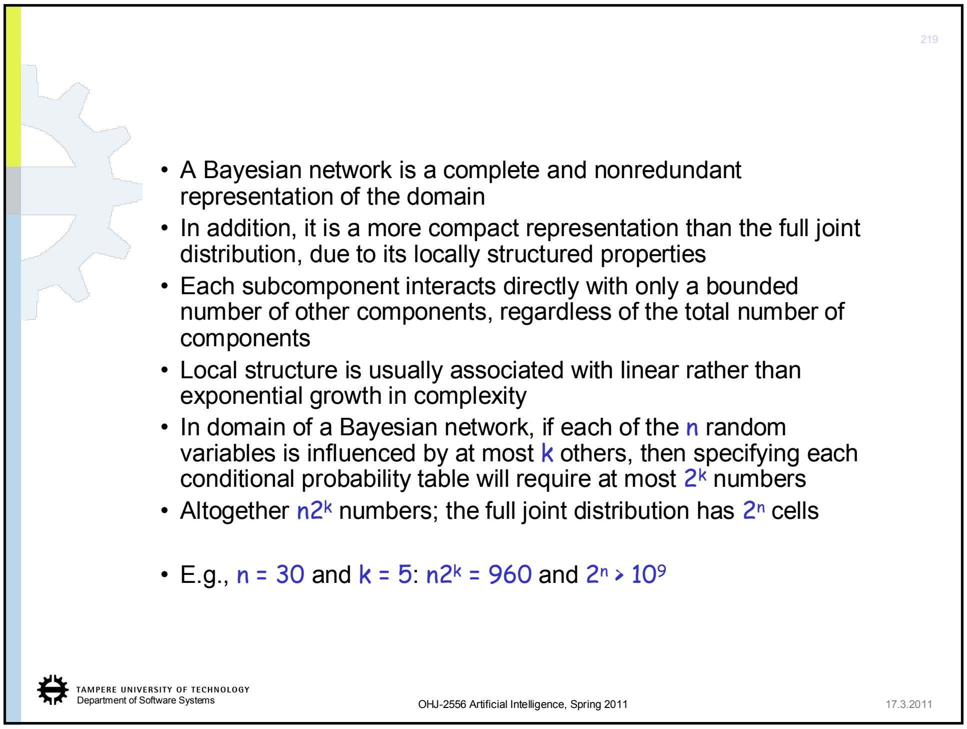
  $$\underline{\mathbf{P}}(X_i \mid X_{i-1}, \ldots, X_1) = \underline{\mathbf{P}}(X_i \mid Parents(X_i))$$

  provided that $Parents(X_i) \subseteq \{ X_{i-1}, \ldots, X_1 \}$



- The last condition is satisfied by labeling the nodes in any order that is consistent with the partial order implicit in the graph structure
- Each node is required to be conditionally independent of its predecessors in the node ordering, given its parents
- We need to choose as parents of a node all those nodes that directly influence the value of the variable
- For example, MaryCalls is certainly influenced by whether there is a Burglary or an Earthquake, but not directly influenced
- We believe the following conditional independence statement to hold:

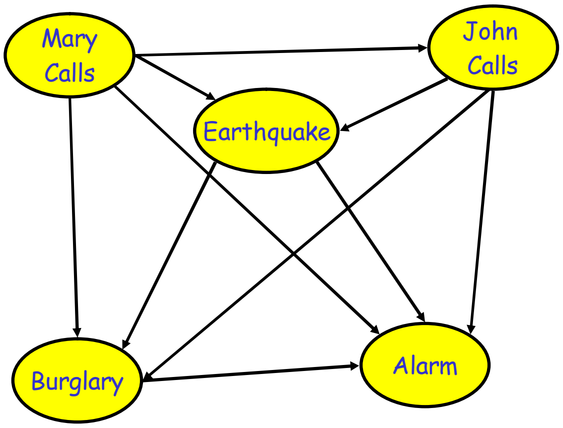
$$\underline{P}(MaryCalls \mid JohnCalls, Alarm, Earthquake, Burglary) = \underline{P}(MaryCalls \mid Alarm)$$



- A Bayesian network is a complete and nonredundant representation of the domain
- In addition, it is a more compact representation than the full joint distribution, due to its locally structured properties
- Each subcomponent interacts directly with only a bounded number of other components, regardless of the total number of components
- Local structure is usually associated with linear rather than exponential growth in complexity
- In domain of a Bayesian network, if each of the n random variables is influenced by at most k others, then specifying each conditional probability table will require at most 2^k numbers
- Altogether $n2^k$ numbers; the full joint distribution has 2^n cells
- E.g., $n = 30$ and $k = 5$: $n2^k = 960$ and $2^n > 10^9$

- In a Bayesian network we can exchange accuracy with complexity
- It may be wiser to leave out very weak dependencies from the network in order to restrict the complexity, but this yields a lower accuracy

- Choosing the right topology for the network is a hard problem
- The correct order in which to add nodes is to add the “root causes“ first, then the variables they influence, and so on, until we reach the “leaves,” which have no direct causal influence on other variables
- Adding nodes in the false order makes the network unnecessarily complex and unintuitive

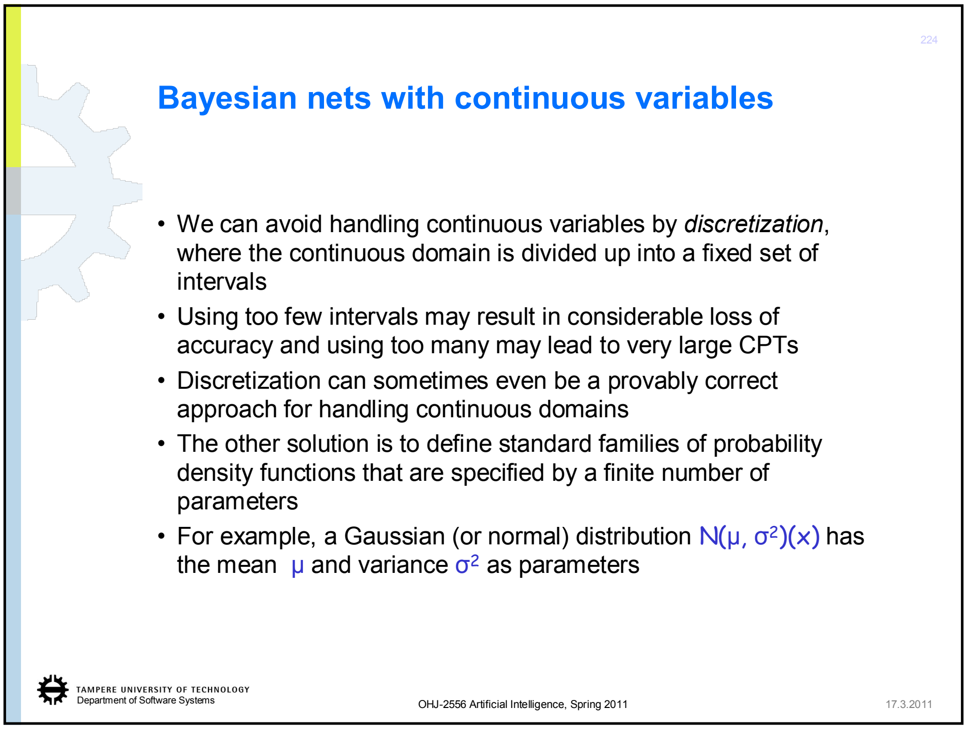


14.3 Representation of Conditional Distributions

- Filling in the CPT for a node requires up to $O(2^k)$ numbers
- Usually the relationship between the parents and the child is described by an easier-to-calculate *canonical distribution*
- The simplest example is a deterministic node, which has its value specified exactly by the values of its parents, with no uncertainty
- An example of a logical relationship is for instance a node . Scandinavian whose value is a disjunction of its parents Swedish, Norwegian, Danish and Icelandic
- Uncertain relationships can often be characterized by so-called "noisy" logical relationships
- In the noisy-OR relation the causal relationship between parent and child may be inhibited
- It is assumed that the inhibition of each parent is independent of the inhibition of any other parents

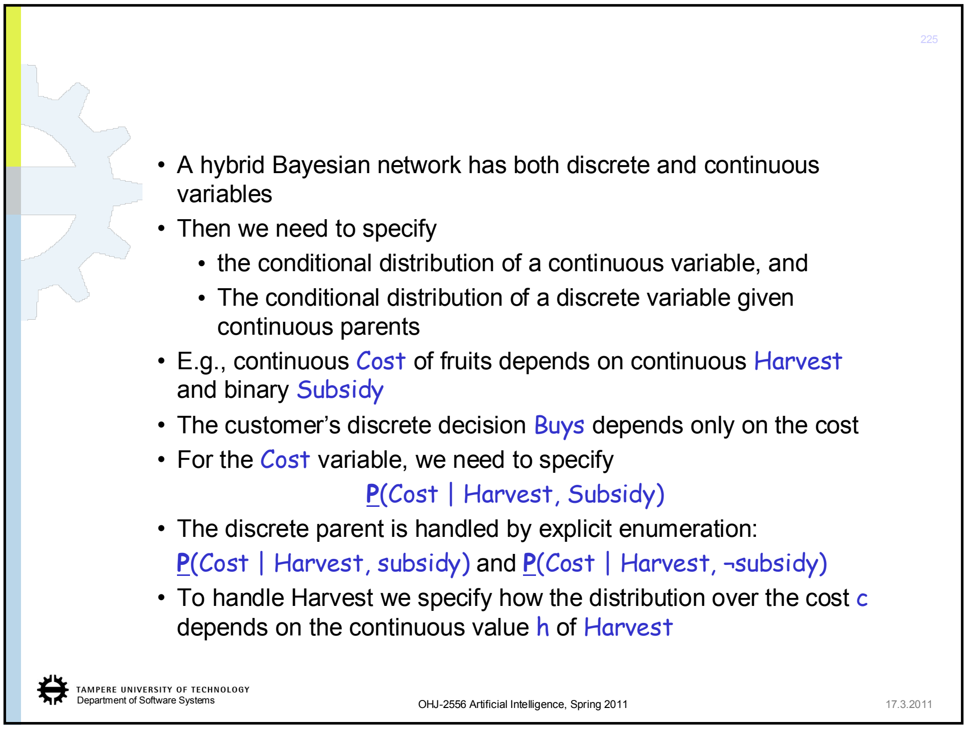
P(¬fever | cold, ¬flu, ¬malaria) = 0.6
P(¬fever | ¬cold, flu, ¬malaria) = 0.2
P(¬fever | ¬cold, ¬flu, malaria) = 0.1

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 = 0.2 × 0.1
T	F	F	0.4	0.6
T	F	T	0.94	0.06 = 0.6 × 0.1
T	T	F	0.88	0.12 = 0.6 × 0.2
T	T	T	0.988	.012 = .6 × .2 × .1



Bayesian nets with continuous variables

- We can avoid handling continuous variables by *discretization*, where the continuous domain is divided up into a fixed set of intervals
- Using too few intervals may result in considerable loss of accuracy and using too many may lead to very large CPTs
- Discretization can sometimes even be a provably correct approach for handling continuous domains
- The other solution is to define standard families of probability density functions that are specified by a finite number of parameters
- For example, a Gaussian (or normal) distribution $N(\mu, \sigma^2)(x)$ has the mean μ and variance σ^2 as parameters

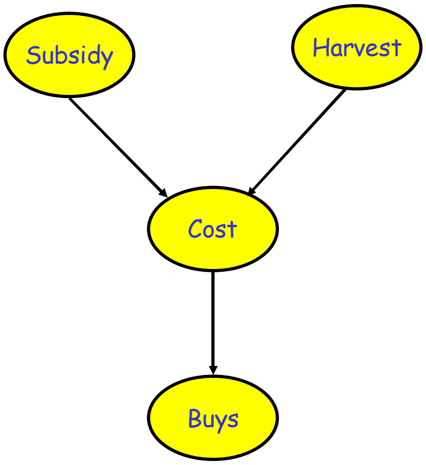


- A hybrid Bayesian network has both discrete and continuous variables
- Then we need to specify
  - the conditional distribution of a continuous variable, and
  - The conditional distribution of a discrete variable given continuous parents
- E.g., continuous Cost of fruits depends on continuous Harvest and binary Subsidy
- The customer's discrete decision Buys depends only on the cost
- For the Cost variable, we need to specify

$$\underline{\mathbf{P}}(Cost \mid Harvest, Subsidy)$$

- The discrete parent is handled by explicit enumeration:

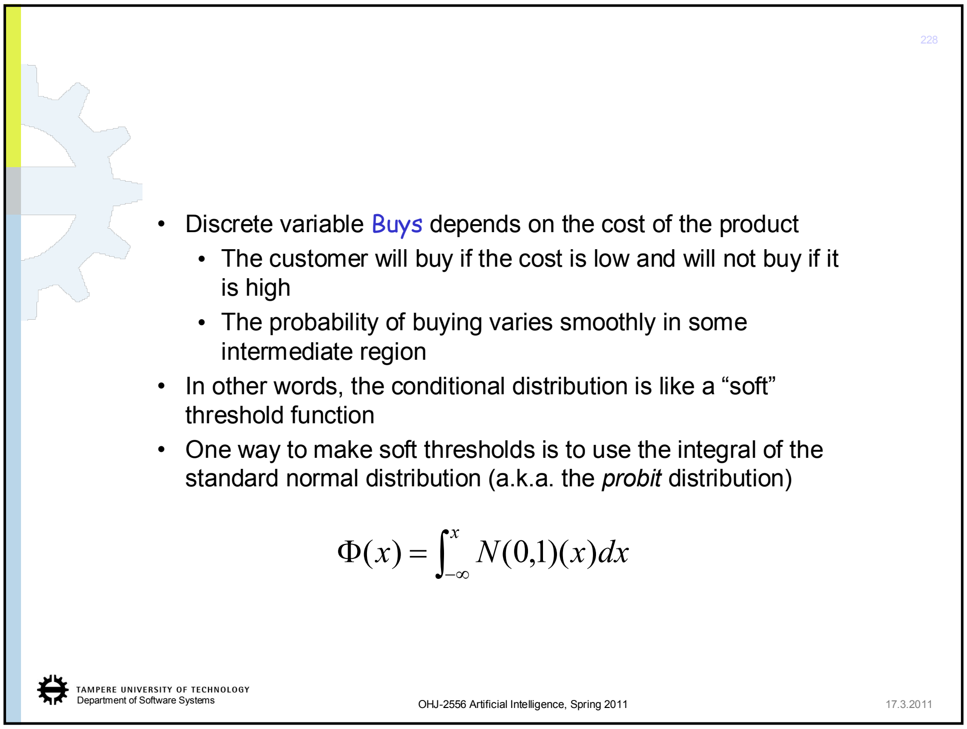
  $\underline{\mathbf{P}}(Cost \mid Harvest, subsidy)$ and $\underline{\mathbf{P}}(Cost \mid Harvest, \neg subsidy)$
- To handle Harvest we specify how the distribution over the cost c depends on the continuous value h of Harvest



- I.e., we specify the parameters of the cost distribution as a function of h
- The most common choice is the linear Gaussian distribution whose mean μ varies linearly with the value of the parent and whose standard deviation σ is fixed

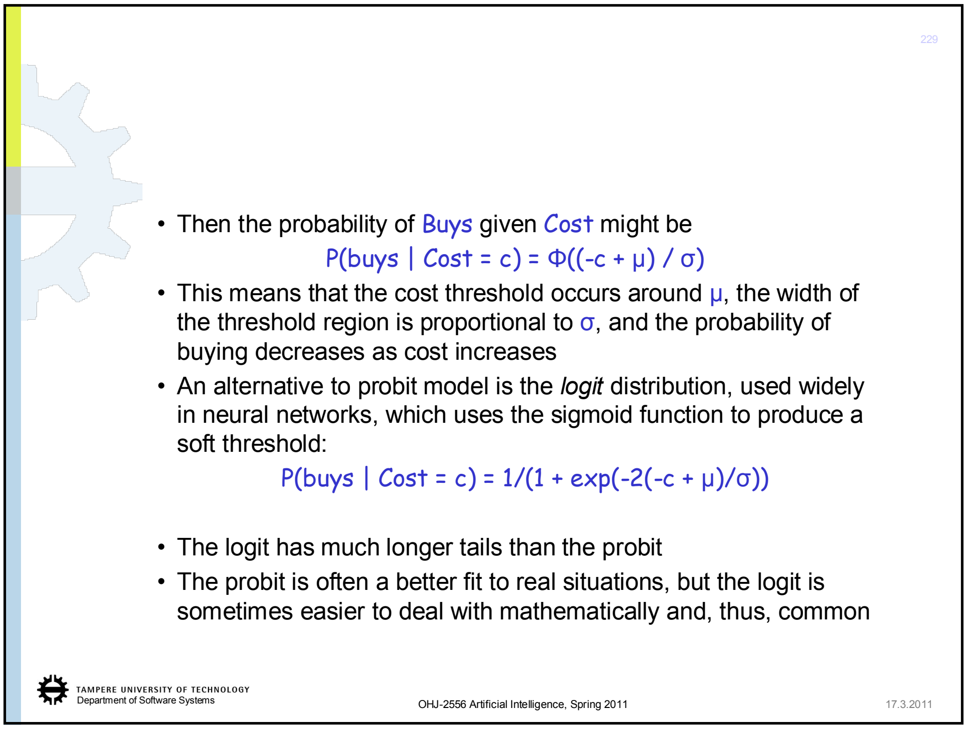
$$
\begin{aligned}
P(c \mid h, subsidy) &= N(a_t h + b_t, \sigma_t^2)(c) \\
&= \frac{1}{\sigma_t \sqrt{2\pi}} \exp\left(-\frac{1}{2} \left((c - (a_t h + b_t)) / \sigma_t \right)^2 \right) \\
P(c \mid h, \neg subsidy) &= N(a_f h + b_f, \sigma_f^2)(c)
\end{aligned}
$$

- Averaging over the two single-bump distributions eventually yields a two-bump distribution P(c | h)
- A Bayesian network employing the linear Gaussian distribution “behaves well” (has an intuitive overall distribution)



- Discrete variable Buys depends on the cost of the product
  - The customer will buy if the cost is low and will not buy if it is high
  - The probability of buying varies smoothly in some intermediate region
- In other words, the conditional distribution is like a “soft” threshold function
- One way to make soft thresholds is to use the integral of the standard normal distribution (a.k.a. the *probit* distribution)

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



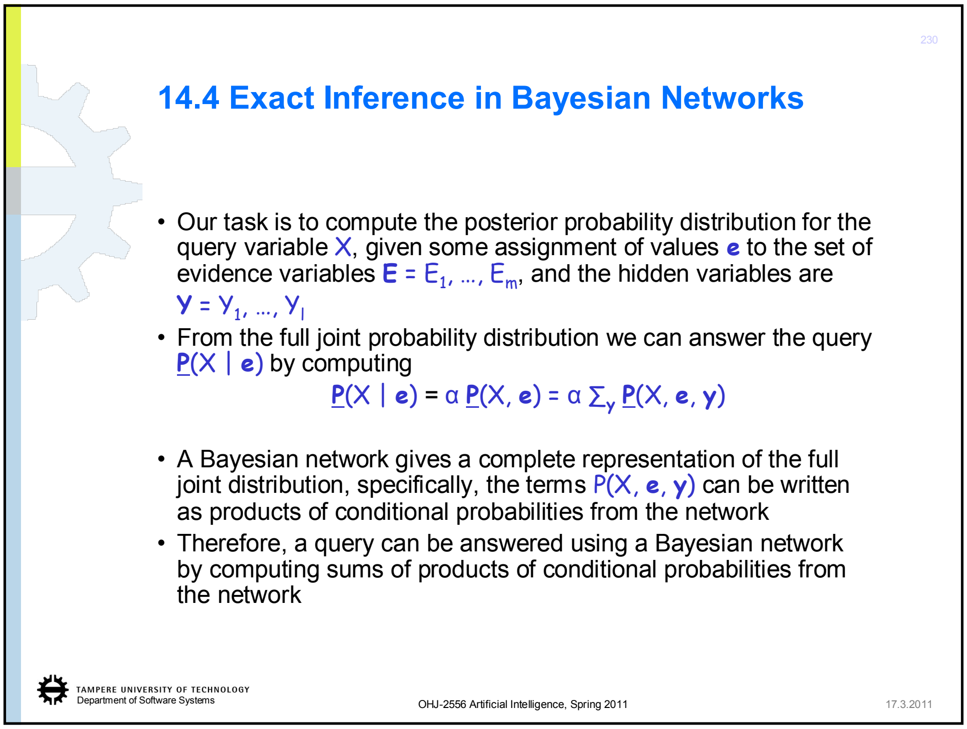
- Then the probability of Buys given Cost might be

$$P(buys \mid Cost = c) = \Phi((-c + \mu) / \sigma)$$

- This means that the cost threshold occurs around μ, the width of the threshold region is proportional to σ, and the probability of buying decreases as cost increases
- An alternative to probit model is the *logit* distribution, used widely in neural networks, which uses the sigmoid function to produce a soft threshold:

$$P(buys \mid Cost = c) = 1/(1 + \exp(-2(-c + \mu)/\sigma))$$

- The logit has much longer tails than the probit
- The probit is often a better fit to real situations, but the logit is sometimes easier to deal with mathematically and, thus, common

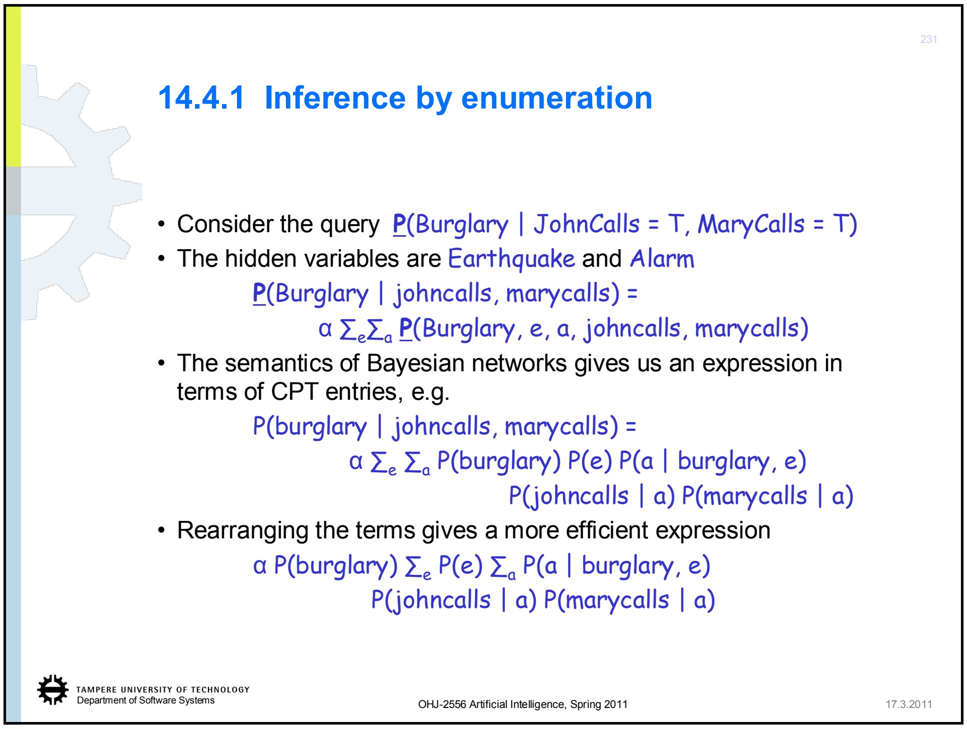


14.4 Exact Inference in Bayesian Networks

- Our task is to compute the posterior probability distribution for the query variable X, given some assignment of values $\mathbf{e}$ to the set of evidence variables $\mathbf{E} = E_1, \ldots, E_m$, and the hidden variables are $\mathbf{Y} = Y_1, \ldots, Y_l$
- From the full joint probability distribution we can answer the query $\underline{P}(X \mid \mathbf{e})$ by computing

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

- A Bayesian network gives a complete representation of the full joint distribution, specifically, the terms $P(X, \mathbf{e}, \mathbf{y})$ can be written as products of conditional probabilities from the network
- Therefore, a query can be answered using a Bayesian network by computing sums of products of conditional probabilities from the network



14.4.1 Inference by enumeration

- Consider the query $\underline{P}(Burglary \mid JohnCalls = T, MaryCalls = T)$
- The hidden variables are $Earthquake$ and $Alarm$

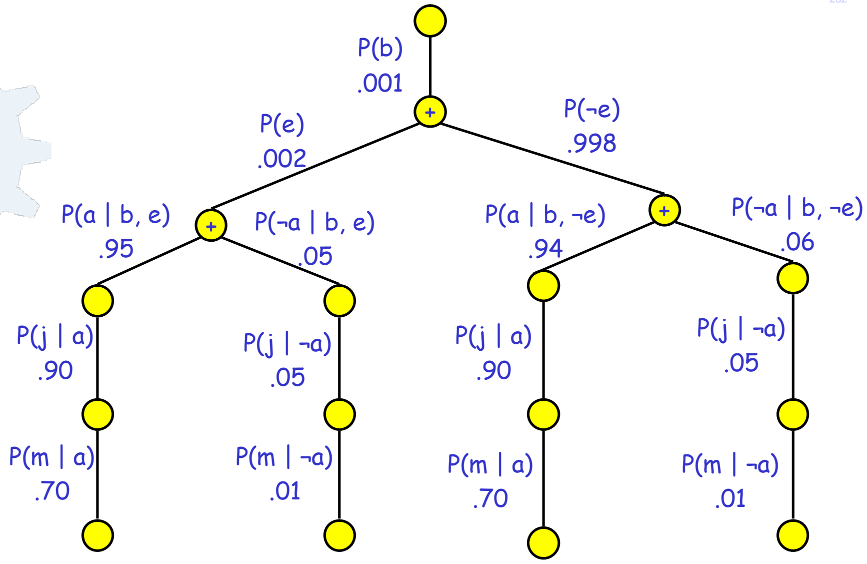
$$\underline{P}(Burglary \mid johncalls, marycalls) =$$
$$\alpha \sum_{e}\sum_{a} \underline{P}(Burglary, e, a, johncalls, marycalls)$$

- The semantics of Bayesian networks gives us an expression in terms of CPT entries, e.g.

$$P(burglary \mid johncalls, marycalls) =$$
$$\alpha \sum_{e} \sum_{a} P(burglary)\, P(e)\, P(a \mid burglary, e)$$
$$P(johncalls \mid a)\, P(marycalls \mid a)$$

- Rearranging the terms gives a more efficient expression

$$\alpha\, P(burglary) \sum_{e} P(e) \sum_{a} P(a \mid burglary, e)$$
$$P(johncalls \mid a)\, P(marycalls \mid a)$$



- Looping over possible values: P(burglary | johncalls, maycalls) and P(¬burglary | johncalls, marycalls), using the numbers used before yields

  $\underline{\mathrm{P}}$(Burglary | johncalls, marycalls) ≈ [0.284, 0.716]
- The evaluation of the DAG of a Bayesian network corresponds to the depth-first recursion of a tree, and thus the space complexity is only linear in the number of variables
- Its time complexity for a network with n variables is always $O(2^n)$
- E.g., the product P(johncalls | a) P(marycalls | a) needs to be recomputed for each value e
- By avoiding repeated subexpression re-evaluations helps to avoid wasted computations

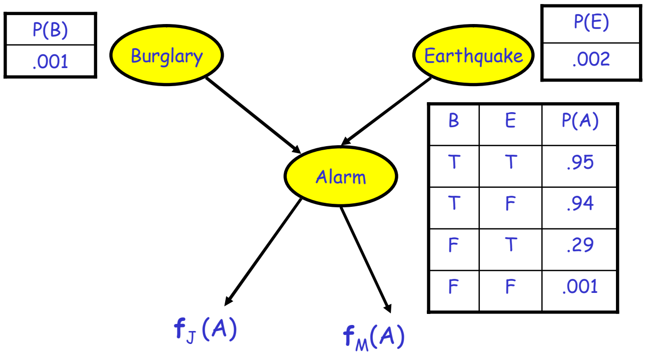
14.4.2 Variable elimination

- Repeated calculation of subexpressions can be avoided by calculating them just once and save the results for later use
- Let us illustrate the variable elimination algorithm in evaluating the expression $\underline{\mathbf{P}}(Burglary \mid johncalls, marycalls)$:

$$\alpha \underbrace{\mathbf{P}(Burglary)}_{B} \sum_e \underbrace{P(e)}_{E} \sum_a \underbrace{\mathbf{P}(a \mid Burglary, e)}_{A} \cdot \underbrace{P(johncalls \mid a)}_{J} \underbrace{P(marycalls \mid a)}_{M}$$

- The factors of the expression have been associated with names
- The factor M, $P(marycalls \mid a)$, does not require summing over $MaryCalls$, because the value $marycalls$ is already fixed

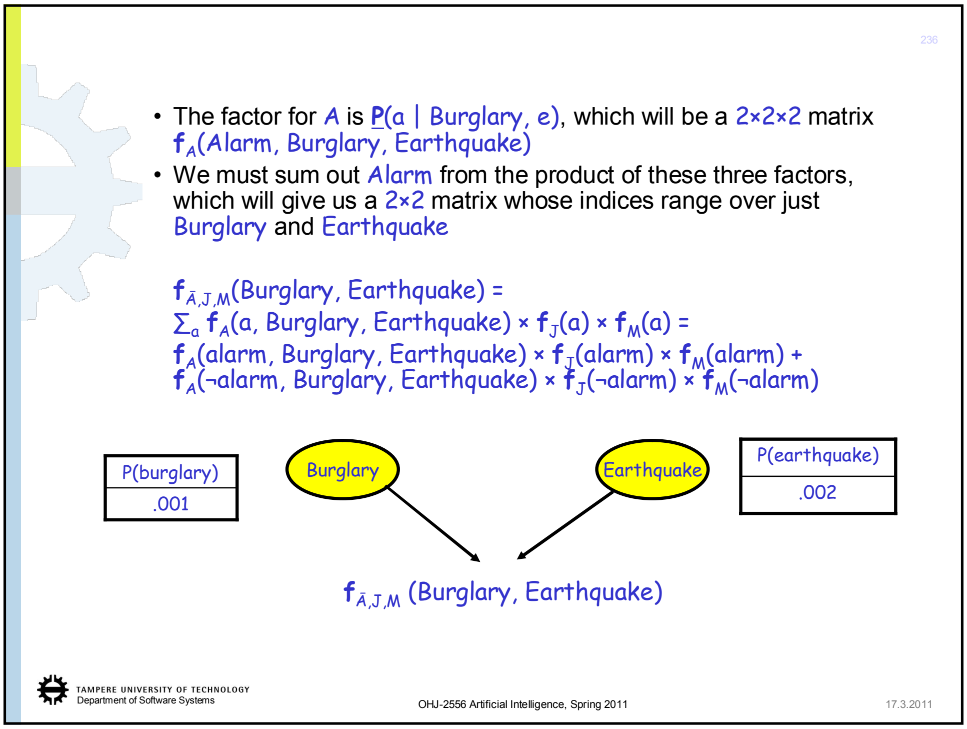
- We store the probability, given each value of a, in a two-element vector $\mathbf{f}_M(Alarm) = [P(marycalls \mid alarm), P(marycalls \mid \neg alarm)]^T$
- Similarly, we store the factor for J as the two-element vector $\mathbf{f}_J(Alarm)$



P(B)
.001

P(E)
.002

B	E	P(A)
T	T	.95
T	F	.94
F	T	.29
F	F	.001



236

- The factor for A is $\underline{P}(a \mid Burglary, e)$, which will be a 2×2×2 matrix $\mathbf{f}_A(Alarm, Burglary, Earthquake)$
- We must sum out Alarm from the product of these three factors, which will give us a 2×2 matrix whose indices range over just Burglary and Earthquake

$$\mathbf{f}_{\bar{A},J,M}(Burglary, Earthquake) =$$
$$\textstyle\sum_a \mathbf{f}_A(a, Burglary, Earthquake) \times \mathbf{f}_J(a) \times \mathbf{f}_M(a) =$$
$$\mathbf{f}_A(alarm, Burglary, Earthquake) \times \mathbf{f}_J(alarm) \times \mathbf{f}_M(alarm) +$$
$$\mathbf{f}_A(\neg alarm, Burglary, Earthquake) \times \mathbf{f}_J(\neg alarm) \times \mathbf{f}_M(\neg alarm)$$

P(burglary)
.001

P(earthquake)
.002

Burglary

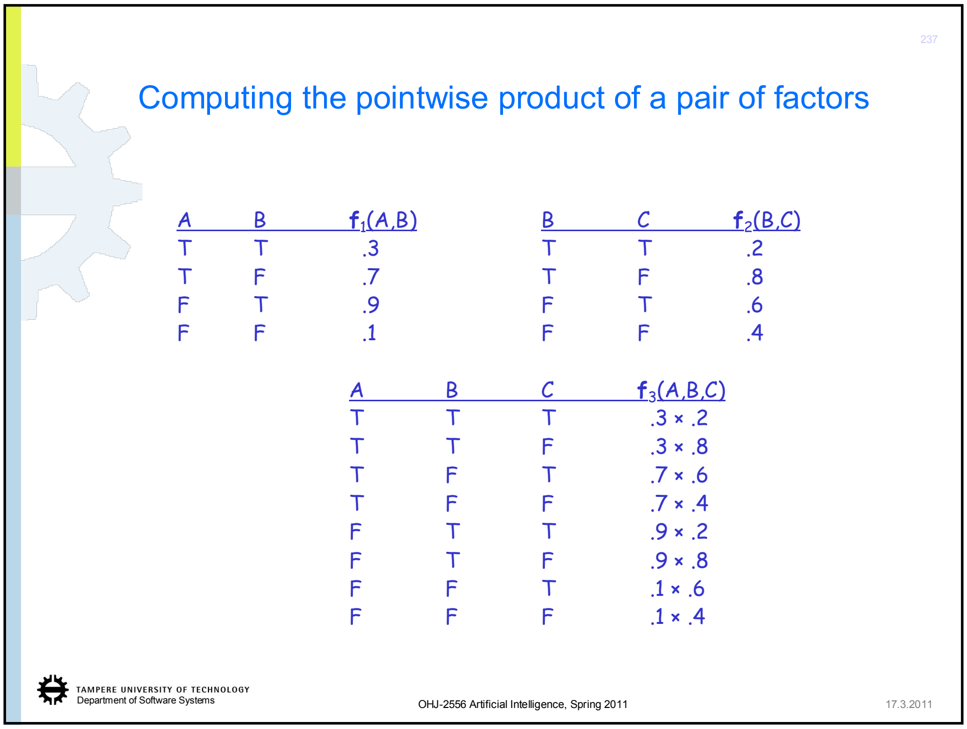
Earthquake

$\mathbf{f}_{\bar{A},J,M}$ (Burglary, Earthquake)

TAMPERE UNIVERSITY OF TECHNOLOGY
Department of Software Systems

OHJ-2556 Artificial Intelligence, Spring 2011

17.3.2011



237

Computing the pointwise product of a pair of factors

A	B	$\mathbf{f}_1(A,B)$
T	T	.3
T	F	.7
F	T	.9
F	F	.1

B	C	$\mathbf{f}_2(B,C)$
T	T	.2
T	F	.8
F	T	.6
F	F	.4

A	B	C	$\mathbf{f}_3(A,B,C)$
T	T	T	.3 × .2
T	T	F	.3 × .8
T	F	T	.7 × .6
T	F	F	.7 × .4
F	T	T	.9 × .2
F	T	F	.9 × .8
F	F	T	.1 × .6
F	F	F	.1 × .4

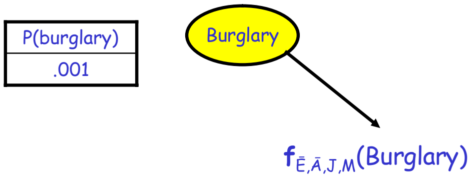
TAMPERE UNIVERSITY OF TECHNOLOGY
Department of Software Systems

OHJ-2556 Artificial Intelligence, Spring 2011

17.3.2011

- We sum out Earthquake in the same way from the product $\mathbf{f}_{E}(Earthquake) \times \mathbf{f}_{\bar{A},J,M}(Burglary, Earthquake)$, which gives the matrix

$$\mathbf{f}_{\bar{E},\bar{A},J,M}(Burglary) =$$
$$\mathbf{f}_{E}(earthquake) \times \mathbf{f}_{\bar{A},J,M}(Burglary, earthquake) +$$
$$\mathbf{f}_{E}(\neg earthquake) \times \mathbf{f}_{\bar{A},J,M}(Burglary, \neg earthquake)$$

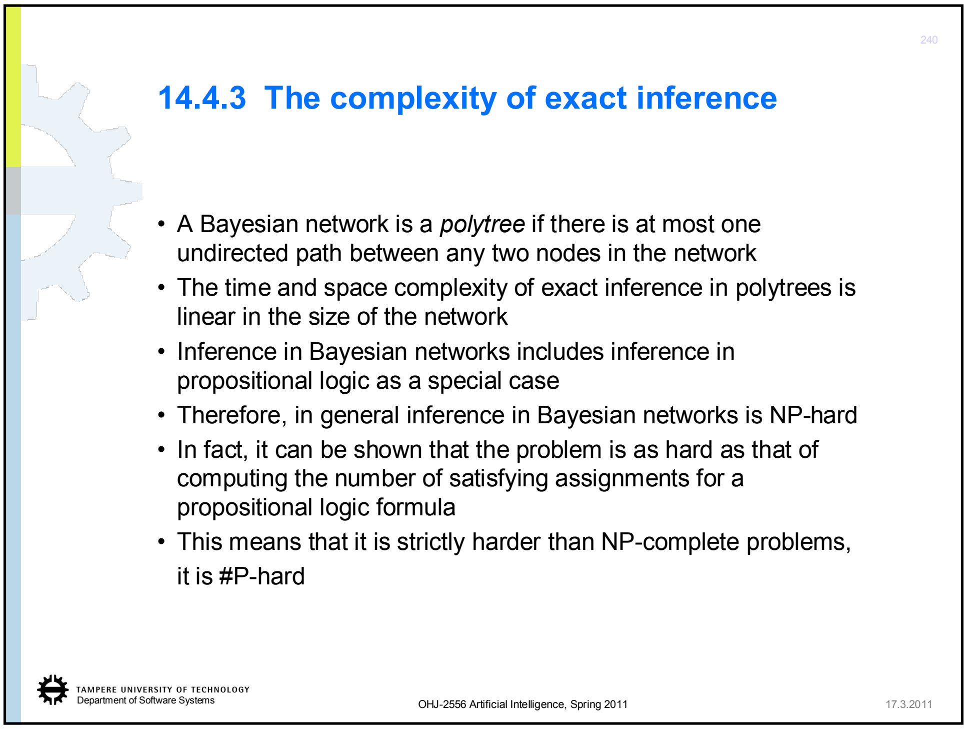


P(burglary)
.001

$\mathbf{f}_{\bar{E},\bar{A},J,M}(Burglary)$

- Now we compute the answer to the query
  $\underline{P}(Burglary \mid johncalls, marycalls)$
  by multiplying
  $\alpha\, \mathbf{f}_{B}(Burglary) \times \mathbf{f}_{\bar{E},\bar{A},J,M}(Burglary)$,
  where $\mathbf{f}_{B}(Burglary) = \underline{P}(Burglary)$
- In summing out variables from a product of a factors, any factor that does not depend on the variable can be moved outside the summation process

- Variables that are irrelevant to the query can be removed
- Query $\underline{P}(JohnCalls \mid burglary)$ yields an expression, whose last factor is $\sum_{a} \ldots \sum_{m} P(MaryCalls = m \mid Alarm = a)$, which is equal to 1 by definition
- In general, every variable that is not an ancestor of a query variable or evidence variable is irrelevant to the query



14.4.3 The complexity of exact inference

- A Bayesian network is a *polytree* if there is at most one undirected path between any two nodes in the network
- The time and space complexity of exact inference in polytrees is linear in the size of the network
- Inference in Bayesian networks includes inference in propositional logic as a special case
- Therefore, in general inference in Bayesian networks is NP-hard
- In fact, it can be shown that the problem is as hard as that of computing the number of satisfying assignments for a propositional logic formula
- This means that it is strictly harder than NP-complete problems, it is #P-hard