CAP 4630
Artificial Intelligence

Instructor: Sam Ganzfried

sganzfri@cis.fiu.edu

Schedule

- 11/14: Finish probability, discuss class project and multiagent systems (game theory)
- 11/16: Markov decision processes & reinforcement learning
- 11/21, 11/28, 11/30, 12/5: Machine learning (classification, regression, clustering, deep learning)
- 12/7: Project presentations and class project due
  - Project code due Monday 12/4 at 2PM on Moodle.
- Final exam on 12/14

Announcements

- HW3 out 10/31 due 11/14 (2:05pm in lecture or 2:00pm on Moodle)
  - https://www.cs.cmu.edu/~sganzfri/HW3_AI.pdf
  - Must be done individually (no partner)
- HW4 out this week

Class project

- For the class project students will implement an agent for 3-player Kuhn poker. This is a simple, yet interesting and nontrivial, variant of poker that has appeared in the AAAI Annual Computer Poker Competition. The grade will be partially based on performance against the other agents in a class-wide competition, as well as final reports and presentations describing the approaches used. Students can work alone or in groups of up to 3.
- Link to play against optimal strategy for one-card poker:
  - http://www.cs.cmu.edu/~ggordon/poker/
- Paper on Nash equilibrium strategies for 3-player Kuhn poker
  - http://poker.cs.ualberta.ca/publications/AAMAS13-3pkuhn.pdf
- https://moodle.cis.fiu.edu/v3.1/mod/forum/discuss.php?d=21801

GRAPHPLAN algorithm

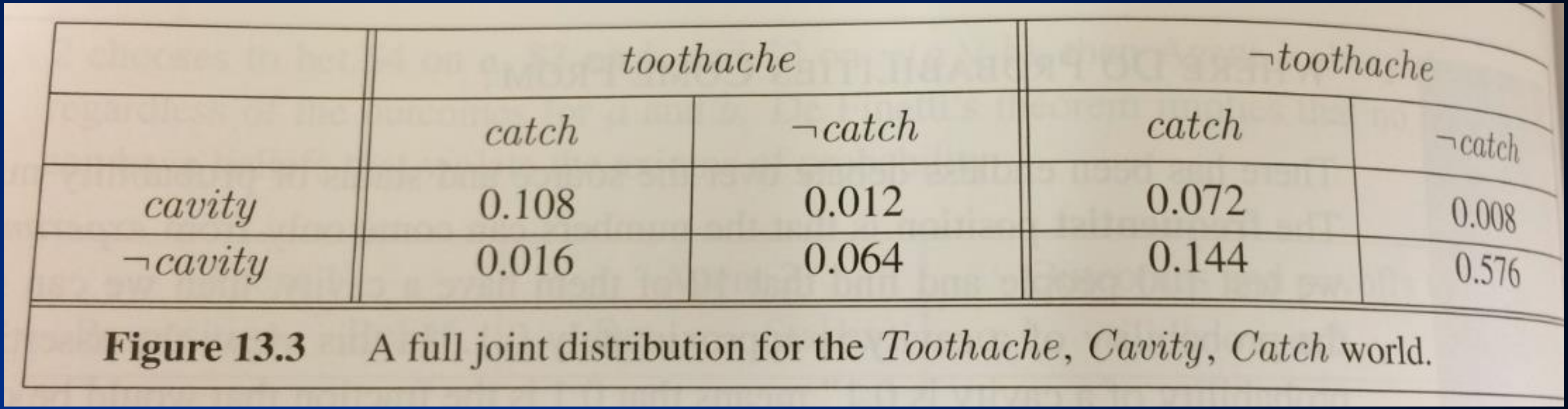
- Graphplan is an algorithm for automated planning developed by Avrim Blum and Merrick Furst in 1995. Graphplan takes as input a planning problem expressed in STRIPS and produces, if one is possible, a sequence of operations for reaching a goal state.
- The name graphplan is due to the use of a novel planning graph, to reduce the amount of search needed to find the solution from straightforward exploration of the state space graph.
- In the state space graph:
  - the nodes are possible states,
  - and the edges indicate reachability through a certain action.
- On the contrary, in Graphplan's planning graph:
  - the nodes are actions and atomic facts, arranged into alternate levels,
  - and the edges are of two kinds:
    - from an atomic fact to the actions for which it is a condition,
    - from an action to the atomic facts it makes true or false.

GRAPHPLAN

- First level contains true atomic facts identifying the initial state.
- Lists of incompatible facts that cannot be true at the same time and incompatible actions that cannot be executed together are also maintained.
- The algorithm then iteratively extends the planning graph, proving that there are no solutions of length l-1 before looking for plans of length l by backward chaining: supposing the goals are true, Graphplan looks for the actions and previous states from which the goals can be reached, pruning as many of them as possible thanks to incompatibility information.
- A closely related approach to planning is the Planning as Satisfiability (Satplan). Both reduce the automated planning problem to search for plans of different fixed horizon lengths.

Probability

- Consider a domain with three Boolean variables: *Toothache*, *Cavity*, *Catch* (the dentist's steel probe catches in my tooth).



	toothache		*¬toothache*	
	catch	*¬catch*	*catch*	*¬catch*
cavity	0.108	0.012	0.072	0.008
¬cavity	0.016	0.064	0.144	0.576

Figure 13.3 A full joint distribution for the *Toothache*, *Cavity*, *Catch* world.

Probability

- Notice that the probabilities in the **joint distribution** sum to 1, as required by the **axioms of probability**.
- Axioms of probability:
  1. 0 <= P(w) <= 1 for every possible world w
  2. Sum over all worlds w of P(w) = 1
- For example, if we roll two dice, there are 36 possible worlds: (1,1), (1,2), …, (6,6).
  - If each die is fair and rolls don't interfere with each other, then each world has probability 1/36.
  - On the other hand, if the dice conspire to produce the same number, then the worlds (1,1), (2,2), (3,3), etc. might have higher probabilities, leaving the others with lower probabilities.

Probability

- Technique to calculate the probability of any proposition, simple or complex: identify those possible worlds in which the proposition is true and add up their probabilities. For example, there are six possible worlds in which *cavity OR toothache* holds:
  - P(*cavity OR toothache*) = 0.108 + 0.012 + 0.072 + 0.008 + 0.016 + 0.064 = 0.28.
- One particularly common task is to extract the distribution over some subset of variables or a single variable. For example, adding the entries in the first row gives the **marginal probability** of *cavity*:
  - P(cavity) = 0.108 + 0.102 + 0.072 + 0.008 = 0.2.

Probability

- In general, for any sets of variables Y and Z, $P(Y) = \Sigma_{z \text{ in } Z} P(Y,z)$
- $P(\mathit{Cavity}) = \Sigma_{z \text{ in } \{\text{Catch, Toothache}\}} P(\mathit{Cavity},z)$
- Conditional probability:
  - P(a | b) = P(a AND b) / P(b) whenever P(b) > 0
  - P(*doubles* | Die1 = 5) = P(*doubles* AND Die1 = 5)/P(Die1 = 5)
- P(*cavity* | *toothache*) = P(*cavity AND toothache) / P(toothache)*

  = (0.108 + 0.012) / (.108 + 0.012 + 0.016 + 0.064) = 0.6.
- P(!*cavity* | *toothache*) = P(!*cavity AND toothache) / P(toothache)*

  = (0.016 + 0.064) / (.108 + 0.012 + 0.016 + 0.064) = 0.4.
- These two values sum to 1 as they should. This can be viewed as **normalization**.

Independence

- Let us expand the full joint distribution by adding a fourth variable, *Weather*. The full joint distribution then becomes P(Toothache, Catch, Cavity, Weather), which has 2 x 2 x 2 x 4 = 32 entries. It contains four “editions” of the table shown, one for each kind of weather.
- How do these editions relate to each other and to the original three-variable table? For example, P(toothache, catch, cavity, cloudy) vs. P(toothache, catch, cavity)?
- We can use the **product rule**:

  P(toothache, catch, cavity, cloudy)

  = P(cloudy | toothache, catch, cavity) * P(toothache, catch, cavity).

Probability

- Now, unless one is in the deity business, one should not imagine that one's dental problems influence the weather. And for indoor dentistry, at least, it seems safe to say that the weather does not influence the dental variables.
- Therefore, the following assertion seems reasonable:

  P(cloudy | toothache, catch, cavity) = P(cloudy).
- From this, we can deduce

  P(toothache, catch, cavity, cloudy) = P(cloudy)P(toothache, catch, cavity).
- A similar equation exists for every entry in P(toothache, catch, cavity, weather). In fact, we can write the general equation:

  P(toothache, catch, cavity, weather) = P(toothache, catch, cavity) P(weather)

Probability

- Thus, the 32-element table for four variables can be constructed from one 8-element table and one 4-element table. This decomposition is illustrated schematically in next slide. The property we used is called **independence** (also **marginal independence** and **absolute independence**). In particular, the weather is independent of one's dental problems. Independence between propositions a and b can be written as:
  - $P(a|b) = P(a)$ or
  - $P(b|a) = P(b)$ or
  - $P(a \text{ AND } b) = P(a)P(b)$

Independence

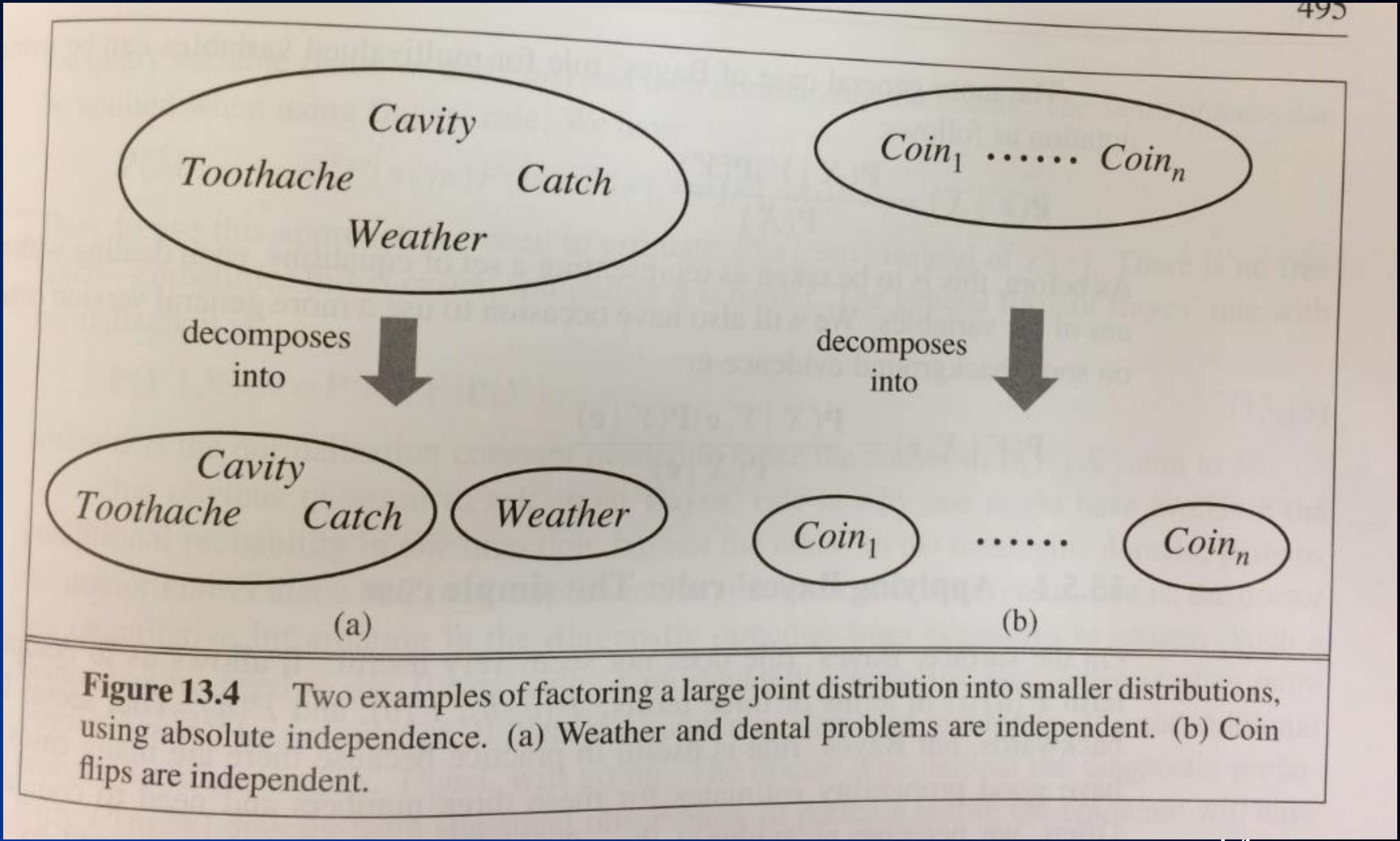


Figure 13.4 Two examples of factoring a large joint distribution into smaller distributions, using absolute independence. (a) Weather and dental problems are independent. (b) Coin flips are independent.

Independence

- Independence assertions are usually based on knowledge of the domain. As the toothache-weather example illustrates, they can dramatically reduce the amount of information necessary to specify the full joint distribution. If the complete set of variables can be divided into independent subsets, then the full joint distribution can be *factored* into separate joint distributions on those subsets. For example, the full joint distribution on the outcome of n independent coin flips, $P(C_1,\ldots, C_n)$ has 2^n entries, but it can be represented as the product of n single-variable distributions $P(C_i)$. In a more practical vein, the independence of dentistry and meteorology is a good thing, because otherwise the practice of dentistry might require intimate knowledge of meteorology, and vice versa.

Independence

- When they are available, then, independence assertions can help in reducing the size of the domain representation and the complexity of the inference problem. Unfortunately, clean separation of entire sets of variables by independence is quite rare. Whenever a connection, however indirect, exists between two variables, independence will fail to hold. Moreover, even independent subsets can be quite large—for example, dentistry might involve dozens of diseases and hundreds of symptoms, all of which are interrelated. To handle such problems, we need more subtle methods than the straightforward concept of independence.

Bayes’ Rule

- Recall the **product rule**: $P(a \text{ AND } b) = P(a \mid b)P(b)$, or equivalently, $P(a \text{ AND } b) = P(b|a)P(a)$
- Equating the two right-hand sides and dividing by P(a), we get
  - $P(b|a) = P(a|b)P(b)/P(a)$
- This equation is known as **Bayes’ rule** (also Bayes’ law or Bayes’ theorem). This simple equation underlies most modern AI systems for probabilistic inference.

Bayes' rule

- On the surface, Bayes' rule does not seem very useful. It allows us to compute the single term P(b|a) in terms of three terms: P(a|b), P(b), and P(a). That seems like two steps backwards, but Bayes' rule is useful in practice because there are many cases where we do have good probability estimates for these three numbers and need to compute the fourth. Often, we perceive as evidence the *effect* of some unknown *cause* and we would like to determine that cause. In that case, Bayes' rule becomes
  - P(cause | effect) = P(effect | cause) P(cause) / P(effect)

Bayes' rule

- The conditional probability P(effect | cause) quantifies the relationship in the **causal** direction, whereas P(cause|effect) describes the **diagnostic** direction. In a task such as medical diagnosis, we often have conditional probabilities on causal relationships (that is, the doctor knows P(symptoms| disease) and want to derive a diagnosis, P(disease | symptoms). For example, a doctor knows that the disease meningitis causes the patient to have a stiff neck, say, 70% of the time. The doctor also knows some unconditional facts: the prior probability that a patient has meningitis is 1/50,000, and the prior probability that any patient has a stiff neck is 1%. Letting s be the proposition that the patient has a stiff neck and m be the proposition that the patient has meningitis, we have:

- Meningitis is an acute inflammation of the protective membranes covering the brain and spinal cord, known collectively as the meninges. The most common symptoms are fever, headache, and neck stiffness. Other symptoms include confusion or altered consciousness, vomiting, and an inability to tolerate light or loud noises. Young children often exhibit only nonspecific symptoms, such as irritability, drowsiness, or poor feeding. If a rash is present, it may indicate a particular cause of meningitis; for instance, meningitis caused by meningococcal bacteria may be accompanied by a characteristic rash.
- In 2015 meningitis occurred in about 8.7 million people worldwide. This resulted in 379,000 deaths – down from 464,000 deaths in 1990. With appropriate treatment the risk of death in bacterial meningitis is less than 15%. Outbreaks of bacterial meningitis occur between December and June each year in an area of sub-Saharan Africa known as the meningitis belt. Smaller outbreaks may also occur in other areas of the world. The word meningitis is from Greek μῆνιγξ meninx, "membrane" and the medical suffix -itis, "inflammation".

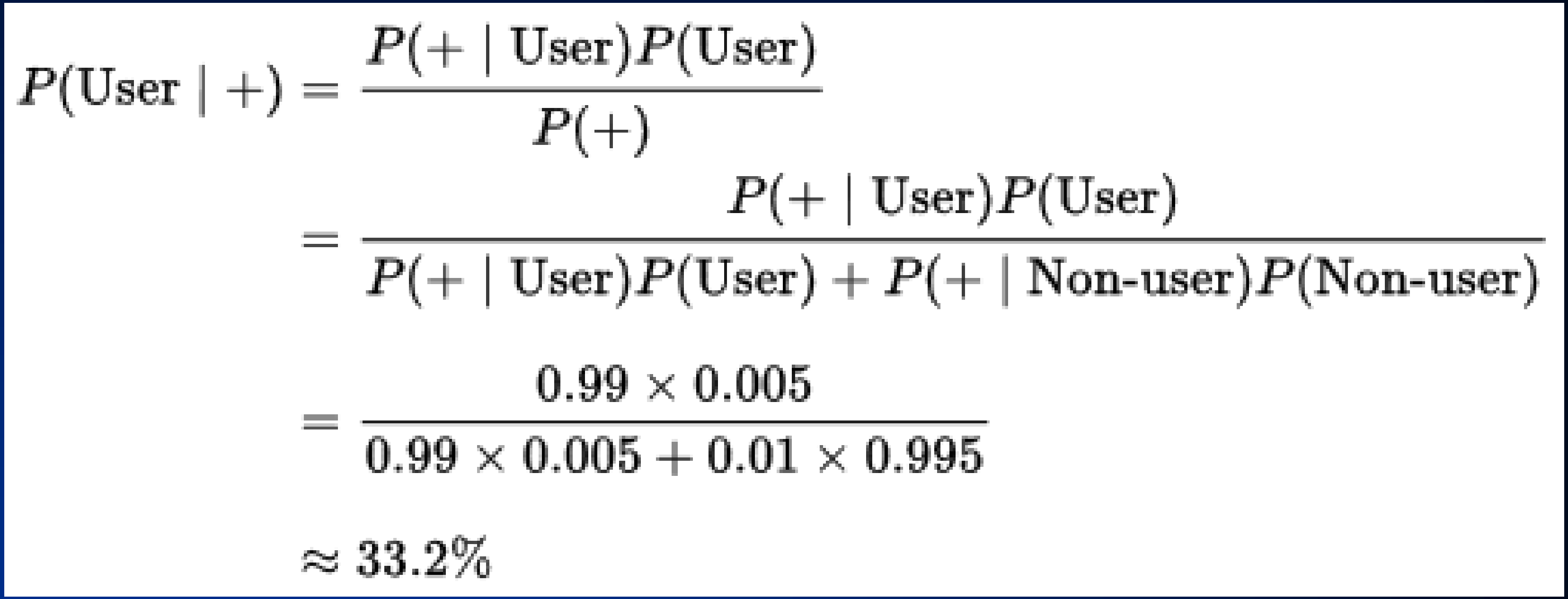
Bayes' rule

- $P(s|m) = 0.7$
- $P(m) = 1/50000$
- $P(s) = 0.01$
- $P(m \mid s) = P(s|m)P(m)/P(s) = (0.7 * 1/5000)/0.01 = 0.0014$
- Thus, we expect less than 1 in 700 patients with a stiff neck to have meningitis. Notice that even though a stiff neck is quite strongly indicated by meningitis (with probability 0.7), the probability of meningitis in the patient remains small. This is because the prior probability of stiff necks is much higher than that of meningitis.

Bayes' rule for drug testing

- Suppose that a test for using a particular drug is 99% sensitive and 99% specific. That is, the test will produce 99% true positive results for drug users and 99% true negative results for non-drug users. Suppose that 0.5% of people are users of the drug. What is the probability that a randomly selected individual with a positive test is a user?

Bayes' rule for drug testing



$$\begin{aligned} P(\text{User} \mid +) &= \frac{P(+ \mid \text{User})P(\text{User})}{P(+)} \\ &= \frac{P(+ \mid \text{User})P(\text{User})}{P(+ \mid \text{User})P(\text{User}) + P(+ \mid \text{Non-user})P(\text{Non-user})} \\ &= \frac{0.99 \times 0.005}{0.99 \times 0.005 + 0.01 \times 0.995} \\ &\approx 33.2\% \end{aligned}$$

Bayes’ rule for drug testing

- Even if an individual tests positive, it is more likely that they do not use the drug than that they do. Why? Even though the test appears to be highly accurate, the number of non-users is large compared to the number of users. The number of false positives outweighs the number of true positives.
- To use concrete numbers, if 1000 individuals are tested, there are expected to be 995 non-users and 5 users. From the 995 non-users, $0.01 \times 995 \simeq 10$ false positives are expected. From the 5 users, $0.99 \times 5 \approx 5$ true positives are expected. Out of 15 positive results, only 5, about 33%, are genuine.
- This illustrates the importance of base rates. Daniel Kahneman has argued that the formation of policy can be egregiously misguided if base rates are neglected when using statistics as a basis for guiding public policy.
- The importance of specificity in this example can be seen by calculating that even if sensitivity is raised to 100% and specificity remains at 99% then the probability of the person being a drug user only rises from 33.2% to 33.4%, but if the sensitivity is held at 99% and the specificity is increased to 99.5% then the probability of the person being a drug user rises to about 49.9%.

Bayesian networks

- A Bayesian network is a directed graph in which each node is annotated with quantitative probability information. The full specification is:
  1. Each node corresponds to a random variable, which may be discrete or continuous
  2. A set of directed links or arrows connects pairs of nodes. If there is an arrow from node X to node Y, X is said to be a *parent* of Y. The graph has no directed cycles (and hence is a directed acyclic graph), or DAG.
  3. Each node X_i has a conditional probability distribution $P(X_i | Parents(X_i))$ that quantifies the effect of the parents on the node.

Bayesian networks

- The topology of the network—the set of nodes and links–specifies the conditional independence relationships that hold in the domain, in a way that will be made precise shortly. The *intuitive* meaning of an arrow is typically that X has a *direct influence* on *Y*, which suggests that causes should be parents of effects. It is usually easy for a domain expert to decide what direct influences exist in the domain—much easier, in fact, than actually specifying the probabilities themselves. Once the topology of the Bayesian network is laid out, we need only specify a conditional probability distribution for each variable, given its parents. We will see that the combination of the topology and the conditional distributions suffices to specify (implicitly) the full joint distribution for all the variables.

Bayesian networks

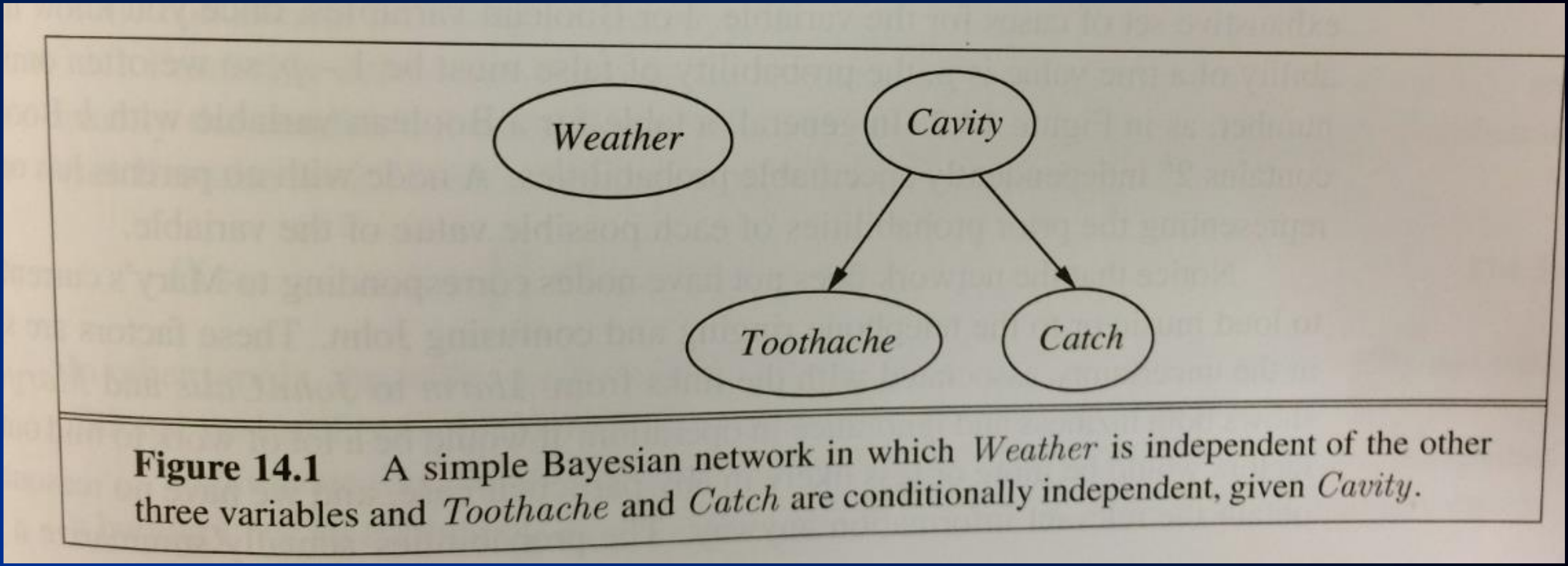


Figure 14.1 A simple Bayesian network in which *Weather* is independent of the other three variables and *Toothache* and *Catch* are conditionally independent, given *Cavity*.

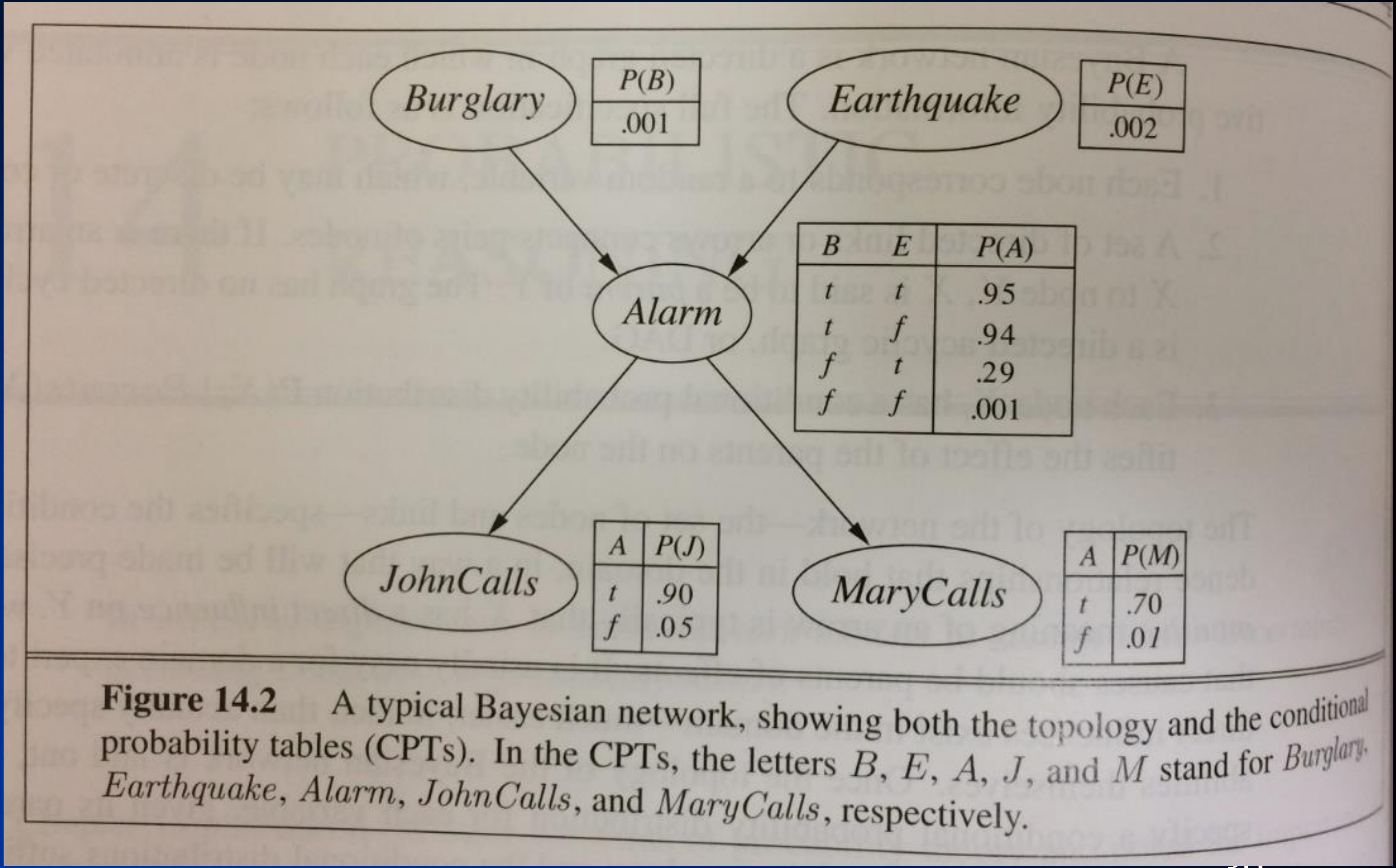
Bayesian networks

- Recall the simple world consisting of the variables *Toothache*, *Cavity*, *Catch*, and *Weather*. We argued that Weather is independent of the other variables; furthermore, we argued that Toothache and Catch are conditionally independent, given Cavity. These relationships are represented by the Bayesian network structure shown above. Formally, the conditional independence of Toothache and Catch, given Cavity, is indicated by the *absence* of a link between Toothache and Catch, whereas no direct causal relationship exists between Toothache and Catch.

Bayesian network

- Now consider the following example. You have a new burglar alarm installed at home. It is fairly reliable at detecting a burglary, but also responds on occasion to minor earthquakes. You also have two neighbors, John and Mary, who have promised to call you at work when they hear the alarm. John nearly 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 rather loud music and often misses the alarm altogether. Given the evidence of who has or has not called, we would like to estimate the probability of a burglary.

Bayesian network



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

Figure 14.2 A typical Bayesian network, showing both the topology and the conditional probability tables (CPTs). In the CPTs, the letters B, E, A, J, and M stand for *Burglary*, *Earthquake*, *Alarm*, *JohnCalls*, and *MaryCalls*, respectively.

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 represents our assumptions that they do not perceive burglaries directly, they do not notice minor earthquakes, and they do not confer before calling.

Bayesian network

- The conditional distributions are shown as a **conditional probability table**, or CPT. Each row in a CPT contains the conditional probability of each node value for a **conditioning case**. A conditioning case is just a possible combination of values for the parent nodes—a miniature possible world. Each row must sum to 1, because the entries represent an exhaustive set of cases for the variable. For Boolean variables, once you know that the probability of a true value is p, the probability of false must be 1-p, so we often omit the second number. In general, a table for a Boolean variable with k Boolean parents contains 2^k independently specifiable probabilities. A node with no parents ha sonly one row, representing the prior probabilities of each possible value of the variable.

Bayesian network

- Notice that the network does not have nodes corresponding to Mary’s currently listening to loud music or to the telephone ringing and confusing John. These factors are summarized in the uncertainty associated with the links from Alarm to JohnCalls and MaryCalls. This shows both laziness and ignorance in operation: it would be a lot of work to find out why those factors would be more or less likely in any particularly case, and we have no reasonable way to obtain the relevant information anyway. The probabilities actually summarize a *potentially infinite* set of circumstances in which the alarm might fail to go off (high humidity, power failure, dead battery, cut wires, a dead mouse stuck inside the bell, etc.) or John or Mary might fail to call and report it (out to lunch, on vacation, temporarily deaf, 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 if introduce additional relevant information.

Bayesian network

- There are two ways in which one can understand the semantics of Bayesian networks. The first is to see the network as a representation of the joint probability distribution. The second is to view it as an encoding of a collection of conditional independence statements. The two views are equivalent, but the first turns out to be helpful in understanding how to *construct* networks, whereas the second is helpful in designing inference procedures.

Bayesian network

- Viewed as a piece of "syntax," a Bayesian network is a directed acyclic graph with some numeric parameters attached to each node. One way to define what the network means—its semantics—is to define the way in which it represents a specific joint distribution over all the variables. To do this, we first need to retract (temporarily) what we said earlier about the parameters associated with each node. We said that those parameters correspond to conditional probabilities $P(X_i|Parents(X_i))$; this is a true statement, but until we assign semantics to the network as a whole, we should think of them just as numbers $\theta(X_i|Parents(X_i))$.

Bayesian networks

- A generic entry in the joint distribution is the probability of a conjunction of particular assignments to each variable, such as $P(X_1 = x_1 \text{ AND } \ldots \text{ AND } X_n = x_n)$. We use the notation $P(x_1,\ldots, x_n)$ as an abbreviation for this. The value of this entry is given by the formula:

$$P(x_1,\ldots, x_n) = \prod^{n}_{i=1} \theta(x_i|\mathrm{Parents}(x_i)),$$

- Where $\mathrm{parents}(X_i)$ denotes the values of $\mathrm{Parents}(x_i)$ that appear in $x_1,\ldots, x_n$. Thus, each entry in the joint distribution is represented by the product of the appropriate elements of the conditional probability tables (CPTs) in the Bayesian network.

Bayesian networks

- From this definition, it is easy to prove that the parameters $\theta(x_i|\text{Parents}(x_i))$, are exactly the conditional probabilities $P(x_i|\text{Parents}(x_i))$, implied by the joint distribution (homework exercise). Hence, we can rewrite the equation as

$$P(x_1,\ldots, x_n) = \prod^{n}_{i=1} P(x_i|\text{Parents}(x_i)).$$

- In other words, the tables we have been calling conditional probability tables really *are* conditional probability tables according to the semantics defined in the equation.

Bayesian network

- To illustrate this, we can calculate the probability that the alarm has sounded, but neither a burglary nor an earthquake has occurred, and both John and Mary call. We multiply entries from the joint distribution (using single-letter names for the variables):
- P(j, m, a, !b,!e) = P(j|a)P(m|a)P(a|!b AND !e)P(!b)P(!e) = 0.90 * 0.70 * 0.001 * 0.999 * 0.998 = 0.000628.
- We explained earlier that the full joint distribution can be used to answer any query about the domain. If a Bayesian network is a representation of the joint distribution, then it too can be used to answer any query, by summing all the relevant joint entries.

Bayesian networks

- Recall the equation:

  $P(x_1,\ldots, x_n) = \prod^{n}_{i=1} P(x_i|\text{Parents}(x_i))$.

- The next step is to explain how to *construct* a Bayesian network in such a way that the resulting joint distribution is a good representation of a given domain. We will now show that the equation implies certain conditional independence relationships that can be used to guide the knowledge engineer in constructing the topology of the network. First, we rewrite the entries in the joint distribution in terms of conditional probability, using product rule:
- $P(x_1,\ldots, x_n) = P(x_n \mid x_{n-1},\ldots,x_1)P(x_{n-1},\ldots, x_1)$.
- Then we repeat the process, reducing each conjunctive probability to a conditional probability and a smaller conjunction.

Bayesian networks

- We end up with one big product:
- $P(x_1,..., x_n) = P(x_n \mid x_{n-1},\ldots,x_1)\ P(x_{n-1} \mid x_{n-2},\ldots,x_1),*\ldots*$ $P(x_2|x_1)P(x_1) = \prod^n_{i=1} P(x_i|x_{i-1},\ldots,x_1)$.
- This identity is called the **chain rule.** It holds for any set of random variables. Comparing it with the previous equation, we see that the specification of the joint distribution is equivalent to the general assertion that, for every variable X_i in the network,
- $P(X_i|X_{i-1},\ldots, X_1) = P(X_i|\text{Parents}(X_i))$,
- Provided that Parents(X_i) is a subset of $\{X_{i-1},\ldots, X_1\}$. This last condition is satisfied by numbering the nodes in a way that is consistent with the partial order implicit in the graph structure.

Bayesian networks

- This new equation says that the Bayesian network is a correct representation of the domain only if each node is conditionally independent of its other predecessors in the node ordering, given its parents. We can satisfy this condition with this methodology:
  - Nodes: First determine the set of variables that are required to model the domain. Now order them, $\{X_1,\ldots, X_n\}$. 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 $X_1,\ldots, X_{i-1}$, a minimal set of parents for X_i, such that the equation is satisfied.
    - For each parent insert a link from the parent to X_i.
    - CPTs: Write down the conditional probability table, $P(X_i|\text{Parents}(X_i))$.

Bayesian networks

- Intuitively, the parents of node X_i should contain all those nodes in X_1,.., X_{i-1} that *directly influence* X_i. For example, suppose we have completed the network in the figure except for the choice of parents for MaryCalls. MaryCalls is certainly influenced by whether there is a Burglary or an Earthquake, but not *directly* influenced. Intuitively, our knowledge of the domain tells us that these events influence Mary's calling. Formally speaking, we believe that the following conditional independence statement holds:
  - P(MaryCalls|JohnCalls, Alarm, Earthquake, Burglary) = P(MaryCalls|Alarm).
- Thus, Alarm will be the only parent node for MaryCalls.

Bayesian network construction

- Because each node is connected only to earlier nodes, this construction method guarantees that the network is acyclic. Another important property of Bayesian networks is that they contain no redundant probability values. If there is no redundancy, then there is no chance for inconsistency: *it is impossible for the knowledge engineer or domain expert to create a Bayesian network that violates the axioms of probability*.

Exact inference in Bayesian networks

- The basic task for any probabilistic inference system is to compute the posterior probability distribution for a set of **query** variables, given some observed **event**—that is, some assignment of values to a set of **evidence variables**. To simplify the presentation, we will consider only one query variable at a time; the algorithms can easily be extended to queries with multiple variables. We will use the notation: X denotes the query variable, E denotes the set of evidence variables $E_1,\ldots, E_m$, and e is a particular observed event; Y will denote the nonevidence, nonquery variables $Y_1,\ldots,Y_m$ (called the **hidden variables**). Thus, a complete set of variables is **X** = {X} Union E Union Y. A typical query asks for the posterior probability distribution P(X|e).

Inference in Bayesian networks

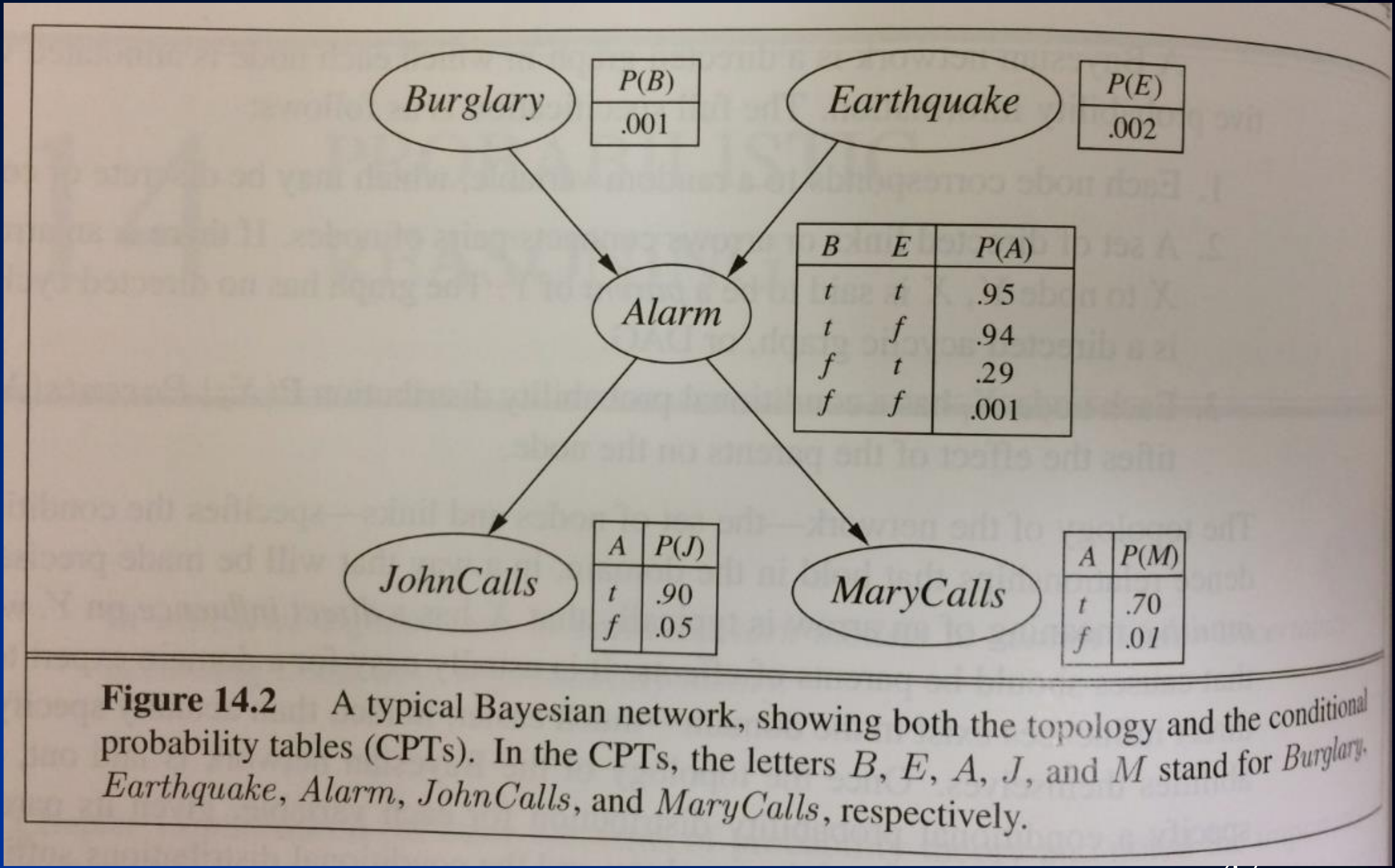
- In the burglary network, we might observe the event in which JohnCalls = true, and MaryCalls = true. We could then ask for, say, the probability that a burglary has occurred:
  - P(Burglary | JohnCalls = true, MaryCalls = true)

    = <0.284, 0.716> (for <true,false>).
- Now we will see exact algorithms for computing posterior probabilities and will consider the complexity of this task. It turns out that the general case is intractable.

Bayesian network

- We saw that any conditional probability can be computed by summing terms from the full joint distribution. More specifically, a query P(X|e) can be answered using the equation, which we repeat here:
- $P(X|e) = \alpha\, P(x|e) = \alpha \sum_y P(x,e,y)$.
- Now a Bayesian network gives a complete representation of the full joint distribution. More specifically, we showed that the terms P(x,e,y) in the joint distribution 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.*

Bayesian network



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

Figure 14.2 A typical Bayesian network, showing both the topology and the conditional probability tables (CPTs). In the CPTs, the letters B, E, A, J, and M stand for *Burglary*, *Earthquake*, *Alarm*, *JohnCalls*, and *MaryCalls*, respectively.

Bayesian network

- Consider the query P(Burglary | JohnCalls = true, MaryCalls = true). The hidden variables for this query are Earthquake and Alarm. We now see that:
- $P(B|j,m) = \alpha\, P(B,j,m) = \alpha \sum_e \sum_a P(B,j,m,e,a)$.
- The semantics of Bayesian networks then gives us an expression in terms of CPT entries. For simplicity, we do this just for Burglary = true:
- $P(b|j,m) = \alpha \sum_e \sum_a P(b)P(e)P(a|b,e)P(j|a)P(m|a)$.
- To compute this expression, we have to add four terms, each computed by multiplying five numbers. In the worst case, where we have to sum out almost all the variables, the complexity of the algorithm for a network with n Boolean variables is $O(n\,2^n)$.

Bayesian network

- An improvement can be made from the following simple observation: the P(b) term is a constant and can be moved outside the summations over a and e, and the P(e) term can be moved outside the summation over a. Hence, we have
- $P(b|j,m) = \alpha\ P(b)\sum_e P(e) \sum_a P(a|b,e)P(j|a)P(m|a)$.
- This expression can be evaluated by looping through the variables in order, multiplying CPT entries as we go. For each summation, we also need to loop over the variable's possible values. The structure of this computation is shown in the figure. Using the numbers, we obtain P(b|j,m) = α <0.00059224, 0.0014919> ~= <0.284,0.716>.
- That is, the chance of a burglary, given calls from both neighbors, is about 28%.

Bayesian network

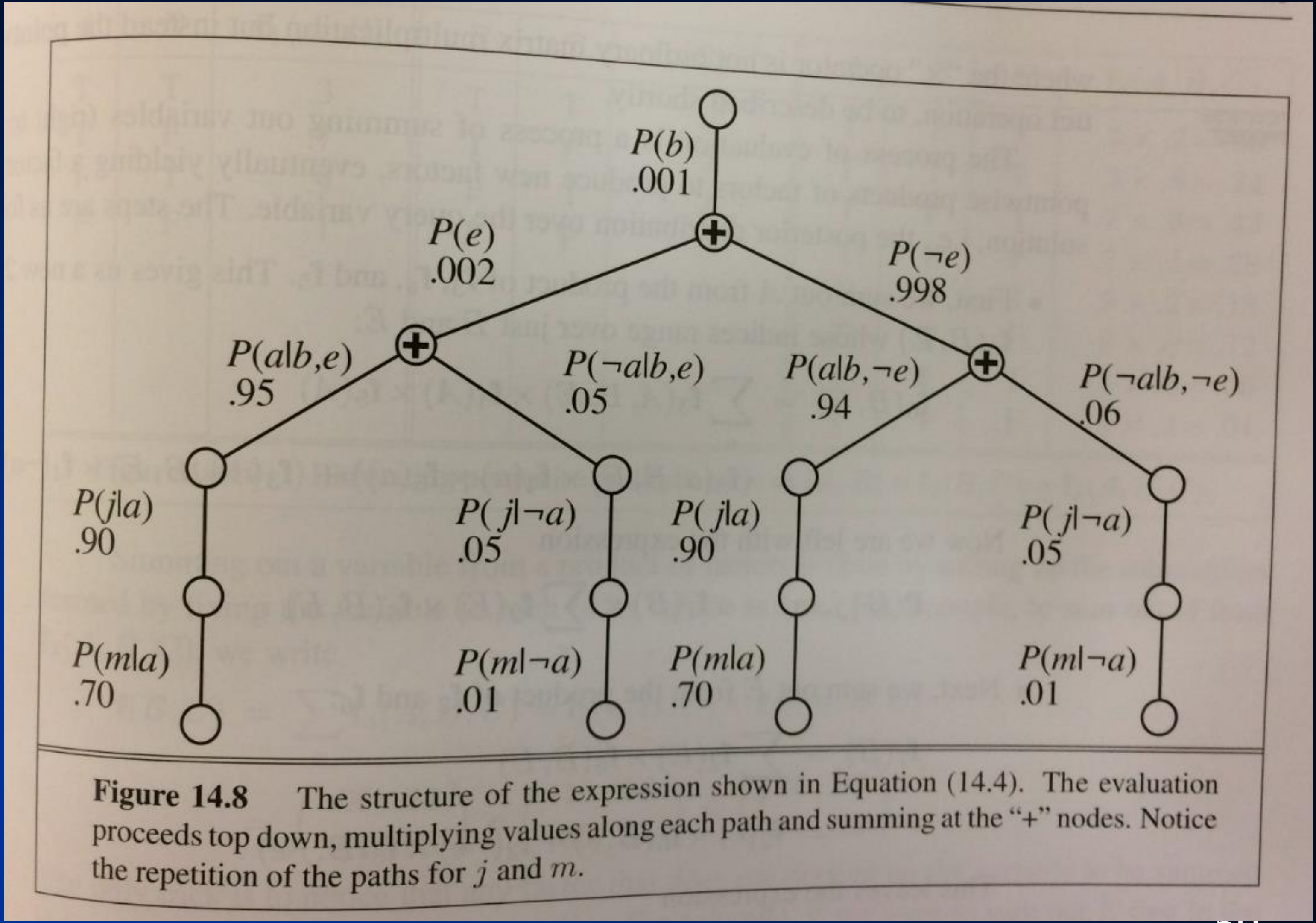
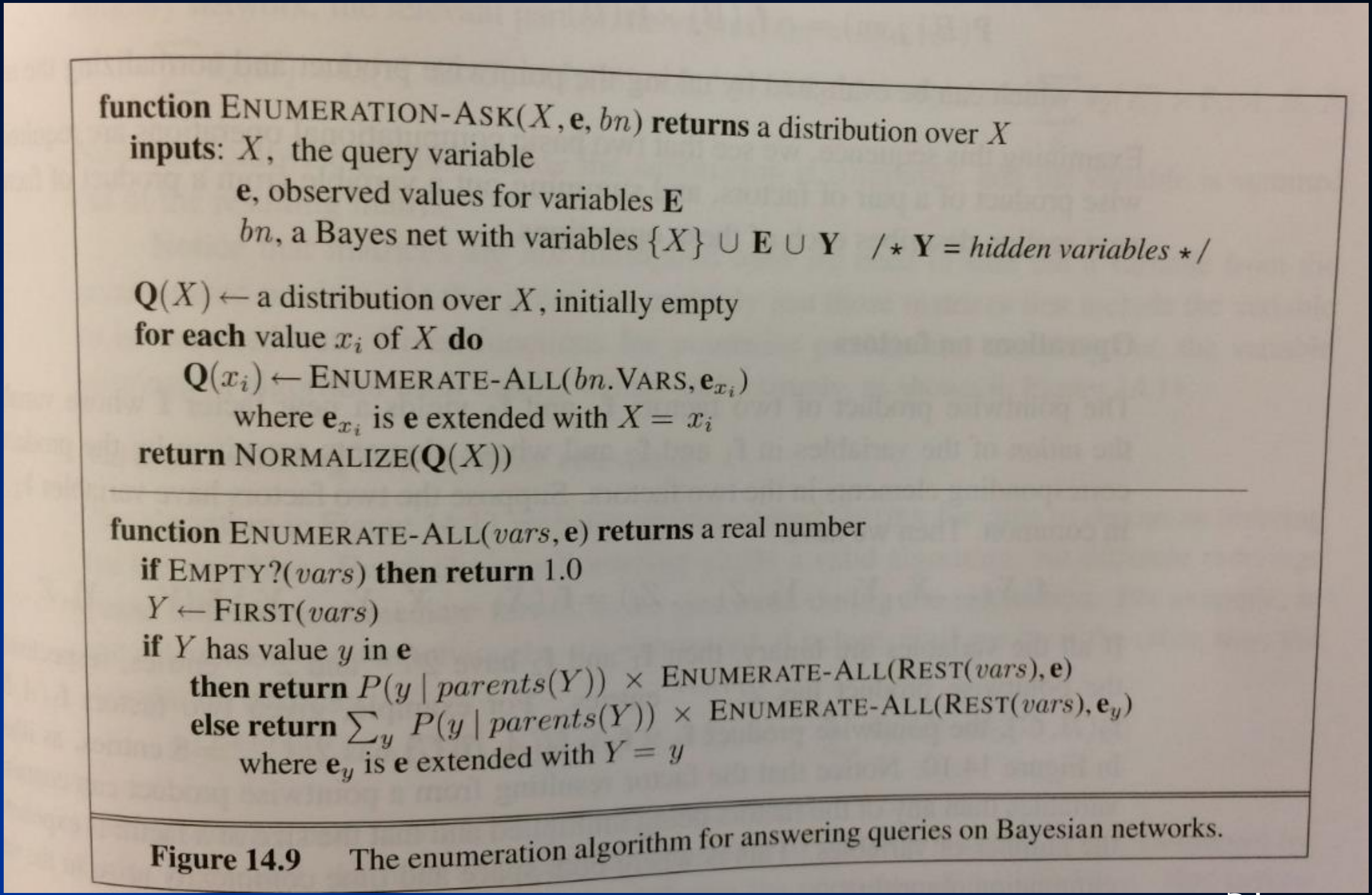


Figure 14.8 The structure of the expression shown in Equation (14.4). The evaluation proceeds top down, multiplying values along each path and summing at the "+" nodes. Notice the repetition of the paths for j and m.

Bayesian network



```
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 $Y = y$
```

Figure 14.9 The enumeration algorithm for answering queries on Bayesian networks.

Homework for next class

- Chapter 17 from Russel/Norvig
- HW3 due today
- HW4 out this week
- Next lecture: Markov decision processes and reinforcement learning