Reasoning Under Uncertainty

- Introduction
- Representing uncertain knowledge: logic and probability (a reminder!)
- Probabilistic inference using the joint probability distribution
- Bayesian networks (theory and algorithms)
- Other approaches to uncertainty

The Importance of Uncertainty

Uncertainty is unavoidable in everyday reasoning and in many real-world domains.

Examples:

- Waiting for a colleague who has not shown up for a meeting.
- Deciding whether to go to school on a very snowy winter morning.
- Judgmental domains such as medicine, business, law and so on.

Sources of Uncertainty

- **Incomplete knowledge.** E.g., laboratory data can be late, medical science can have an incomplete theory for some diseases.
- **Imprecise knowledge.** E.g., the time that an event happened can be known only approximately.
- **Unreliable knowledge.** E.g., a measuring instrument can be biased or defective.

Representing Certain Knowledge: Logic

Example: Patient John has a cavity.

How can we represent this fact in logic?

- Propositional logic: $Cavity$
- First-order logic: $DentalDisease(John, Cavity)$

Ontological commitments: Facts hold or do not hold in the world.

Epistemological commitments: An agent believes a sentence to be true, false or has no opinion.

Question

How can an agent capture the fact that he is **not certain** that John has a cavity?

First Answer

Use logic:

> I have no knowledge regarding whether John has a cavity or not.

The formulas $Cavity$ and $\neg Cavity$ do not follow from my KB.

Second (Better?) Answer

Use probabilities:

The probability that patient John has a cavity is 0.8.

We might know this from statistical data or some general dental knowledge.

Representing Uncertain Knowledge: Probability

- Probabilities provide us with a way of assigning **degrees of belief** in a sentence.
- Probability is a way of **summarizing the uncertainty** regarding a situation.
- The exact probability assigned to a sentence depends on existing **evidence**: the knowledge available up to date.
- Probabilities can change when **more evidence** is acquired.

Probability

Ontological commitments: Facts hold or do not hold in the world.

Epistemological commitments: A probabilistic agent has a **degree of belief** in a particular sentence. Degrees of belief range from 0 (for sentences that are certainly false) to 1 (for sentences that are certainly true).

Uncertainty and Rational Decisions

- Agents have **preferences** over states of the world that are possible outcomes of their actions.
- Every state of the world has a degree of usefulness, or **utility**, to an agent. Agents prefer states with higher utility.
- **Decision theory**=Probability theory + Utility theory
- An agent is **rational** if and only if it chooses the action that yields the highest expected utility, averaged over all the possible outcomes of the action.

Probability Theory: The Basics (AI Style)

- Like logical assertions, probabilistic assertions are about **possible worlds.**
- Logical assertions say which possible worlds (interpretations) are ruled out (those in which the KB assertions are false).
- Probabilistic assertions talk about **how probable the various worlds are**.

The Basics (cont'd)

- The set of possible worlds is called the **sample space** (denoted by Ω). The elements of Ω (**sample points**) will be denoted by ω.
- The possible words of Ω (e.g., outcomes of throwing a dice) are **mutually exclusive** and **exhaustive**.
- In standard probability theory textbooks, instead of possible worlds we talk about **outcomes**, and instead of sets of possible worlds we talk about **events** (e.g., when two dice sum up to 11).

  We will represent events by **propositions in a logical language** which we will define formally later.

Basic Axioms of Probability Theory

Every possible world ω is assigned a number $P(\omega)$ which is called the **probability of** ω.

This number has to satisfy the following conditions:

- $0 \leq P(\omega) \leq 1$
- $\sum_{\omega \in \Omega} P(\omega) = 1$
- For any proposition ϕ, $P(\phi) = \sum_{\phi \text{ holds in } \omega \in \Omega} P(\omega)$.

Basic Axioms of Probability Theory (cont'd)

The last condition is usually given in standard probability textbooks as follows.

Let A and B be two events such that $A \cap B \neq \emptyset$. Then:

$$P(A \cup B) = P(A) + P(B)$$

Then, the following more general formula, that corresponds to the formula we gave above, can be proven by induction.

Let $A_i,\ i = 1, \ldots, n$ be events such that $A_i \cap A_j \neq \emptyset$ for all $i \neq j$. Then:

$$P(\bigcup_{i=1}^{n} A_i) = \sum_{i=1}^{n} P(A_i).$$

Example - Question

Let us consider the experiment of throwing two fair dice. What is the probability that the total of the two numbers that will appear on the dice is 11?

Example - Answer

$$P(Total = 11) = P((5,6)) + P((6,5)) = 1/36 + 1/36 = 2/36 = 1/18$$

Where do Probabilities Come From?

There has been a philosophical debate regarding the source and meaning of probabilities:

- Frequency interpretation
- Subjective interpretation (degrees of belief)
- ...

Consequences of the Basic Axioms

- $P(\neg a) = 1 - P(a)$
- $P(a \vee b) = P(a) + P(b) - P(a \wedge b)$
- ...

Unconditional vs. Conditional Probability

- **Unconditional** or **prior** probabilities refer to degrees of belief in propositions in the absence of any other information.

  Example: $P(Total = 11)$

- **Conditional** or **posterior** probabilities refer to degrees of belief in propositions given some more information which is usually called **evidence**.

  Example: $P(Doubles = true \mid Die_1 = 5)$

Conditional Probability

- Conditional probabilities are defined in terms of unconditional ones.
- For any propositions a and b such that $P(b) > 0$, we define the **(conditional) probability of a given b** as follows:

$$P(a|b) = \frac{P(a \wedge b)}{P(b)}$$

- From the above definition we have:

$$P(a \wedge b) = P(a|b)\ P(b)$$

This equation is traditionally called the **product rule**.

Example

$$P(Doubles = true \mid Dice_1 = 5) = \frac{P(Doubles = true \wedge Dice_1 = 5)}{P(Dice_1 = 5)} =$$

$$\frac{1/36}{1/6} = 1/6$$

Random Variables

- **Random variables** are variables that take values assigned to elements of a sample space (outcomes of an experiment in the traditional probability jargon).
- A random variable takes its values from a certain **domain**.

  Examples:
  - The domain of $Dice_1$ is $\{1, \ldots, 6\}$.
  - The domain of $Weather = \{sunny, rainy, cloudy, snowy\}$.
  - A Boolean random variable has the domain $\{true,\ false\}$.
- **Notation:** The names of random variables (e.g., $Total$) will start with an upper case letter.

Random Variables (cont'd)

- Random variables can be **discrete** (with finite or countably infinite domain) or **continuous** (the domain is a subset of $\mathbb{R}$).

  We will only consider discrete random variables.

Notation

- If X is a random variable, we will write $P(x_i)$ instead of $P(X = x_i)$.
- If A is a Boolean random variable, we will write a instead of $A = true$ and $\neg a$ instead of $A = false$.
- If X is random variable, we will use $\mathbf{P}(X)$ to denote the **vector** of probabilities

$$\langle P(X = x_1), \ldots, P(X = x_n) \rangle.$$

$\mathbf{P}(X \mid Y)$ is defined similarly.

Our Language: Syntax

- **Propositions:** Boolean combinations of atomic formulas of the form $X = x_i$ where X is a random variable and x_i a value in its domain.
- **Probability assertions:** $P(\phi)$ and $P(\phi \mid \psi)$ where ϕ and ψ are propositions.

Our Language: Semantics

- A **possible world** is an assignment of values to all the random variables under consideration.
- **Propositions:** If ϕ is a proposition, checking whether ϕ holds or not in a possible world can be done as in propositional logic.
- **Probability assertions:** Use the axiom

  For any proposition ϕ, $P(\phi) = \sum_{\phi \text{ holds in } \omega \in \Omega} P(\omega)$.

Probabilistic Inference

We are interested in doing **probabilistic reasoning** or **probabilistic inference**: computing posterior probabilities for query propositions given observed evidence.

We will present two methods:

- Using the full joint probability distribution of the involved random variables.
- Using graphs called Bayesian networks.

Let us start by defining the **full joint probability distribution**.

Probability Distribution of a Random Variable

- A **probability distribution** or **probability density function (p.d.f.)** of a random variable X is a function that tells us how the probability mass (i.e., total mass of 1) is allocated across the values that X can take.
- For a discrete random variable X, a probability density function is the function $f(x) = P(X = x)$.
- All the values of a probability density function for X are given by the vector $\mathbf{P}(X)$.

Example

$$P(Weather = sunny) = 0.6$$

$$P(Weather = rainy) = 0.1$$

$$P(Weather = cloudy) = 0.29$$

$$P(Weather = snowy) = 0.01$$

Equivalently as a vector:

$$\mathbf{P}(Weather) = \langle 0.6, 0.1, 0.29, 0.01 \rangle$$

Example (cont'd)

Equivalently as a table:

	$P(\cdot)$
$Weather = sunny$	0.6
$Weather = rainny$	0.1
$Weather = cloudy$	0.29
$Weather = snowy$	0.01

The Joint Probability Distribution

If we have more than one random variable and we are considering problems that involve two or more of these variables at the same time, then the **joint probability distribution** specifies degrees of belief in the values that these functions take **jointly**.

The joint probability distribution $\mathbf{P}(\mathbf{X})$, where $\mathbf{X}$ is a vector of random variables, is usually specified graphically by a n-dimensional table (where n is the dimension of $\mathbf{X}$).

Example: (two Boolean variables *Toothache* and *Cavity*)

	toothache	$\neg$*toothache*
cavity	0.04	0.06
$\neg$*cavity*	0.01	0.89

The Full Joint Probability Distribution

The **full joint probability distribution** is the joint probability distribution for all random variables.

If we have this distribution, then **we can compute the probability of any propositional sentence** using the formulas about probabilities we presented earlier.

Example I

	toothache	$\neg$*toothache*
cavity	0.04	0.06
$\neg$*cavity*	0.01	0.89

The above table gives the **full joint probability distribution** of variables *Toothache* and *Cavity*. Using this table, we can compute:

$$P(toothache) = \sum_{toothache \text{ holds in } \omega \in \Omega} P(\omega) = 0.04 + 0.01 = 0.05$$

$$P(cavity \vee toothache) = 0.04 + 0.01 + 0.06 = 0.11$$

Example I (cont'd)

	$toothache$	$\neg toothache$
$cavity$	0.04	0.06
$\neg cavity$	0.01	0.89

$$P(cavity \mid toothache) = \frac{P(cavity \wedge toothache)}{P(toothache)} =$$

$$\frac{0.04}{0.04 + 0.01} = 0.80$$

Example II

	$toothache$		$\neg toothache$	
	$catch$	$\neg catch$	$catch$	$\neg catch$
$cavity$	0.108	0.012	0.072	0.008
$\neg cavity$	0.016	0.064	0.144	0.576

The above table gives the **full joint distribution** of $Toothache, Cavity$ and $Catch$.

Computing Marginal Probabilities

We can use the full joint probability to extract the distribution over some subset of variables or a single variable.

Example:

$$P(cavity) = 0.108 + 0.012 + 0.072 + 0.008 = 0.2$$

This is called **marginalizing the joint distribution** to $Cavity$ and the probability we computing the **marginal probability** of $cavity$.

General Formula for Marginal Probabilities

The general formula for computing marginal probabilities is

$$\mathbf{P}(\mathbf{Y}) = \sum_{\mathbf{z} \in \mathbf{Z}} \mathbf{P}(\mathbf{Y}, \mathbf{z})$$

or equivalently (using the product rule)

$$\mathbf{P}(\mathbf{Y}) = \sum_{\mathbf{z} \in \mathbf{Z}} \mathbf{P}(\mathbf{Y}|\mathbf{z})\mathbf{P}(\mathbf{z})$$

The second formula is very useful in practice and it is known as the **total probability theorem**.

Example II (cont'd)

The full joint probability distribution can also be used to compute **conditional probabilities** by first using the relevant definition:

$$P(cavity \mid toothache) = \frac{P(cavity \wedge toothache)}{P(toothache)} =$$

$$\frac{0.108 + 0.012}{0.108 + 0.012 + 0.016 + 0.064} = 0.6$$

$$P(\neg cavity \mid toothache) = \frac{P(\neg cavity \wedge toothache)}{P(toothache)} =$$

$$\frac{0.016 + 0.064}{0.108 + 0.012 + 0.016 + 0.064} = 0.4$$

General Formula for $\mathbf{P}(X|\mathbf{e})$

The examples on the previous slide generalize as follows.

Let X be the query variable, $\mathbf{E}$ be the vector of evidence variables, $\mathbf{e}$ the vector of observed values and $\mathbf{Y}$ the vector of the remaining (unobserved) variables.

Then, the **conditional probability** $\mathbf{P}(X|\mathbf{e})$ can be computed as follows:

$$\mathbf{P}(X|\mathbf{e}) = \frac{\mathbf{P}(X,\mathbf{e})}{\mathbf{P}(\mathbf{e})} = \frac{\sum_{\mathbf{y}} \mathbf{P}(X,\mathbf{e},\mathbf{y})}{\mathbf{P}(\mathbf{e})}$$

Difficulties

- Using the full joint probability distribution table works fine but it needs $O(2^n)$ **space**.
- Specifying probabilities for all combinations of propositional variables might be unnatural, and might require a huge amount of statistical data.
- So this is **not a practical tool** for probabilistic inference. We will see better reasoning tools in the rest of the presentation.

Independence

The notion of independence captures the situation when the probability of a random variable taking a certain value is **not influenced** by the fact that we know the value of some other variable.

Definition. Two propositions a and b are called **independent** if $P(a|b) = P(a)$ (equivalently: $P(b|a) = P(b)$ or $P(a \wedge b) = P(a)P(b)$).

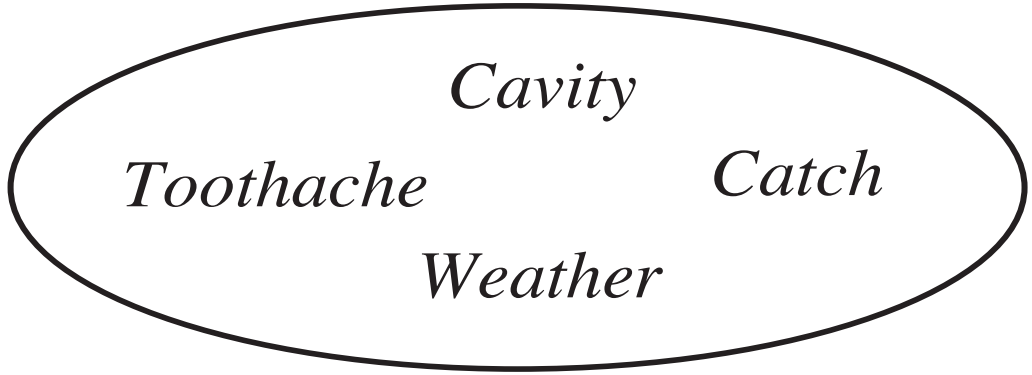
Definition. Two random variables X and Y are called **independent** if $\mathbf{P}(X \mid Y) = \mathbf{P}(X)$ (equivalently: $\mathbf{P}(Y \mid X) = \mathbf{P}(Y)$ or $\mathbf{P}(X, Y) = \mathbf{P}(X)\ \mathbf{P}(Y)$).

Example

$$\mathbf{P}(Weather \mid Toothache, Catch, Cavity) = \mathbf{P}(Weather)$$

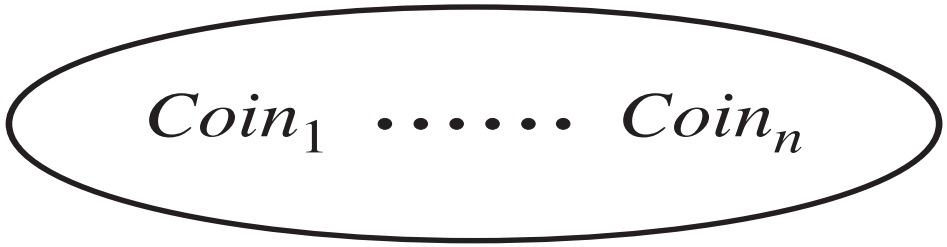
Note: Zeus might be an exception to this rule!

Examples

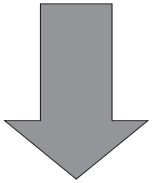


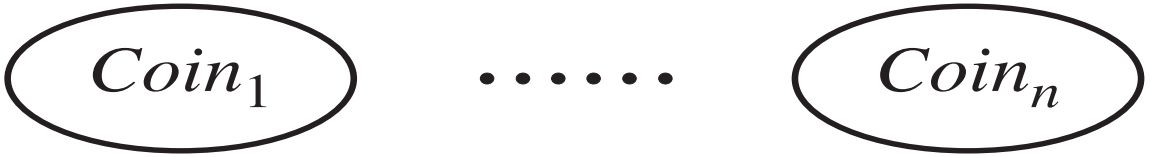
decomposes into

Examples (cont'd)



decomposes into





Difficulties

- Independence is a **useful principle** when we have it, and can be used to reduce the size of the full joint probability distribution tables and the complexity of the inference process.
- But clean separation of random variables into independent sets might be **rare in applications**.

  We will see below the concept of **conditional independence** which is more frequently found in applications.

Bayes' Rule

From the product rule, we have:

$$P(a \wedge b) = P(a|b)P(b) \quad \text{and} \quad P(a \wedge b) = P(b|a)P(a)$$

If we equate the right hand sides and divide by $P(a)$, we have **Bayes' rule**:

$$P(b|a) = \frac{P(a|b)P(b)}{P(a)}$$

Bayes' rule is the **basis of most modern probabilistic inference systems.**

More General Forms of the Bayes' Rule

For random variables:

$$\mathbf{P}(Y \mid X) = \frac{\mathbf{P}(X \mid Y)\ \mathbf{P}(Y)}{\mathbf{P}(X)}$$

Equivalently, using the total probability theorem:

$$\mathbf{P}(Y \mid X) = \frac{\mathbf{P}(X \mid Y)\ \mathbf{P}(Y)}{\sum_{z \in Z} \mathbf{P}(X \mid z)\ P(z)}$$

Conditionalized on some evidence:

$$\mathbf{P}(Y \mid X, \mathbf{e}) = \frac{\mathbf{P}(X \mid Y, \mathbf{e})\ \mathbf{P}(Y \mid \mathbf{e})}{\mathbf{P}(X \mid \mathbf{e})}$$

Applying Bayes' Rule: The Simple Case

In many applications, we perceive as evidence the effect of some unknown cause and we would like to determine that cause. In this case, Bayes' rule can help:

$$P(cause \mid effect) = \frac{P(effect \mid cause)\ P(cause)}{P(effect)}$$

Comments:

- $P(effect \mid cause)$ quantifies the relationship between $cause$ and $effect$ in a **causal** way.
- $P(cause \mid effect)$ does the same thing in a **diagnostic** way.

Application: Medical Diagnosis

A doctor might have the following knowledge:

- Meningitis causes a stiff neck in 70% of the patients.
- The probability that a patient has meningitis is 1/50000.
- The probability that a patient has a stiff neck is 1/100.

Then, we can use Bayes' rule to **compute the probability that a patient has meningitis given that he has a stiff neck.**

Medical Diagnosis (cont'd)

$$P(m \mid s) = \frac{P(s \mid m)\ P(m)}{P(s)} = \frac{0.7 * 1/50000}{0.01} = 0.0014$$

The probability is very small.

There has been much work in using Bayesian networks for medical diagnosis.

Applying Bayes' Rule: Combining Evidence

What happens when we have two or more pieces of evidence?

Example: What can a dentist conclude if her steel probe catches in the aching tooth of a patient?

How can we compute $\mathbf{P}(Cavity \mid toothache \wedge catch)$?

Combining Evidence (cont'd)

- Use the full joint distribution table (**does not scale**).
- Use Bayes' rule:

$$\mathbf{P}(Cavity \mid toothache \wedge catch) = \frac{\mathbf{P}(toothache \wedge catch \mid Cavity)\ \mathbf{P}(Cavity)}{\mathbf{P}(toothache \wedge catch)}$$

This approach **does not scale too if we have a large number of evidence variables**.

Question: Can we use independence?

Combining Evidence (cont'd)

Answer: No, *Toothache* and *Catch* are not independent in general.

If the probe catches in the tooth, then it is likely that the tooth has a cavity and that the cavity causes a toothache.

Conditional Independence

Observation: *Toothache* and *Catch* **are independent given the presence or absence of a cavity.**

Explanation: The presence or absence of a cavity directly causes toothache or the catching of the steel probe. However, **the two variables do not directly depend on each other** if we take *Cavity* into account. The existence of toothache depends on the state of the nerves of the teeth, while, whether the steel probe catches in a tooth, depends on the skill of the dentist.

Conditional Independence (cont'd)

Formally:

$$\mathbf{P}(toothache \wedge catch \mid Cavity) =$$

$$\mathbf{P}(toothache \mid Cavity)\ \mathbf{P}(catch \mid Cavity)$$

This property is called the **conditional independence** of *toothache* and *catch* given *Cavity*.

Conditional Independence (cont'd)

Definition. Let X, Y and Z be random variables. X and Y are **conditionally independent given** Z if

$$\mathbf{P}(X, Y \mid Z) = \mathbf{P}(X \mid Z)\ \mathbf{P}(Y \mid Z).$$

Equivalent conditions (proof?):

$$\mathbf{P}(X \mid Y, Z) = \mathbf{P}(X \mid Z)$$

$$\mathbf{P}(Y \mid X, Z) = \mathbf{P}(Y \mid Z)$$

Example (cont'd)

We can apply conditional independence to the computation of the **full joint probability distribution** as follows:

$$\mathbf{P}(Toothache, Catch, Cavity) =$$

$$\mathbf{P}(Toothache, Catch \mid Cavity)\ \mathbf{P}(Cavity) =$$

$$\mathbf{P}(Toothache \mid Cavity)\ \mathbf{P}(Catch \mid Cavity)\ \mathbf{P}(Cavity)$$

Result: For n symptoms that are conditionally independent given $Cavity$, the size of the representation grows as $O(n)$ instead of $O(2^n)$!

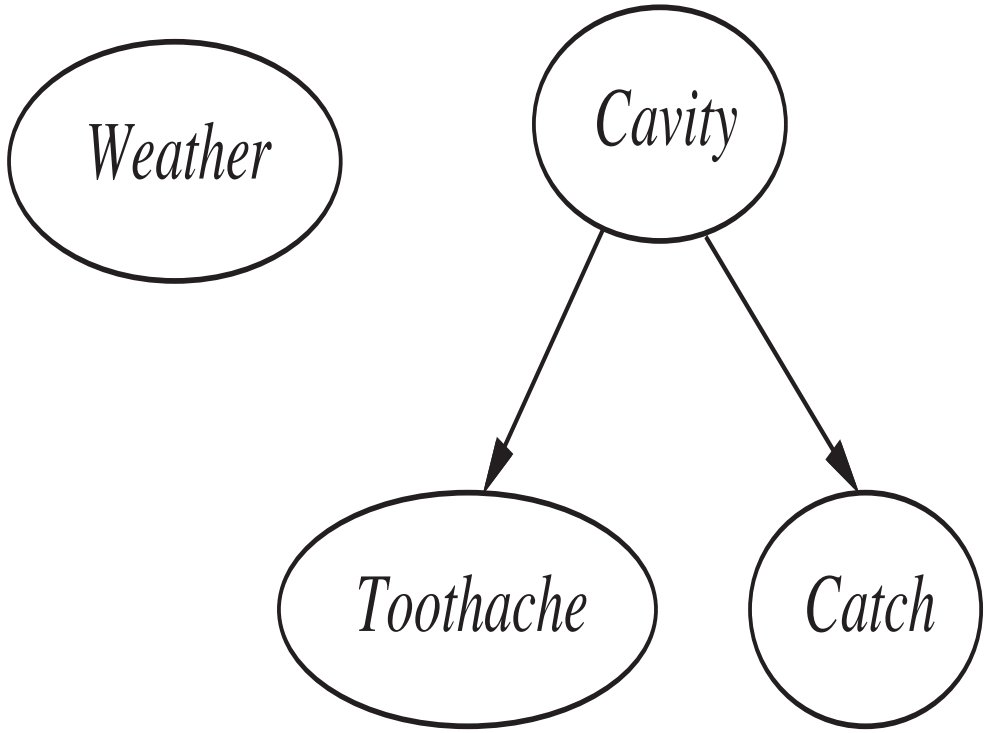
The Importance of Conditional Independence

- Conditional independence assertions allow probabilistic systems (such as Bayesian networks that we will immediately see) to **scale up**.
- They are **much more commonly available in applications** than absolute independence assertions.

Bayesian Networks

- Bayesian networks are graphical representations that allow us to represent explicitly **causal, independence** and **conditional independence** relationships and **reason with them efficiently**.
- Bayesian networks enable us to represent **qualitative** (causality, independence) and **quantitative** (probabilistic) **knowledge**.
- Other terms for the same thing: belief networks, probabilistic networks, causal networks etc.

Example



Bayesian Networks

Definition. A **Bayesian network** is a directed acyclic graph $G = (V, E)$ where:

- Each node $X \in V$ is a random variable (discrete or continuous).
- Each directed edge $(X, Y) \in E$ indicates a **causal dependency** between variables X and Y (i.e., X **directly influences** Y).
- Each node Y has a **conditional probability distribution** $P(Y \mid Parents(Y))$ that quantifies this dependency.

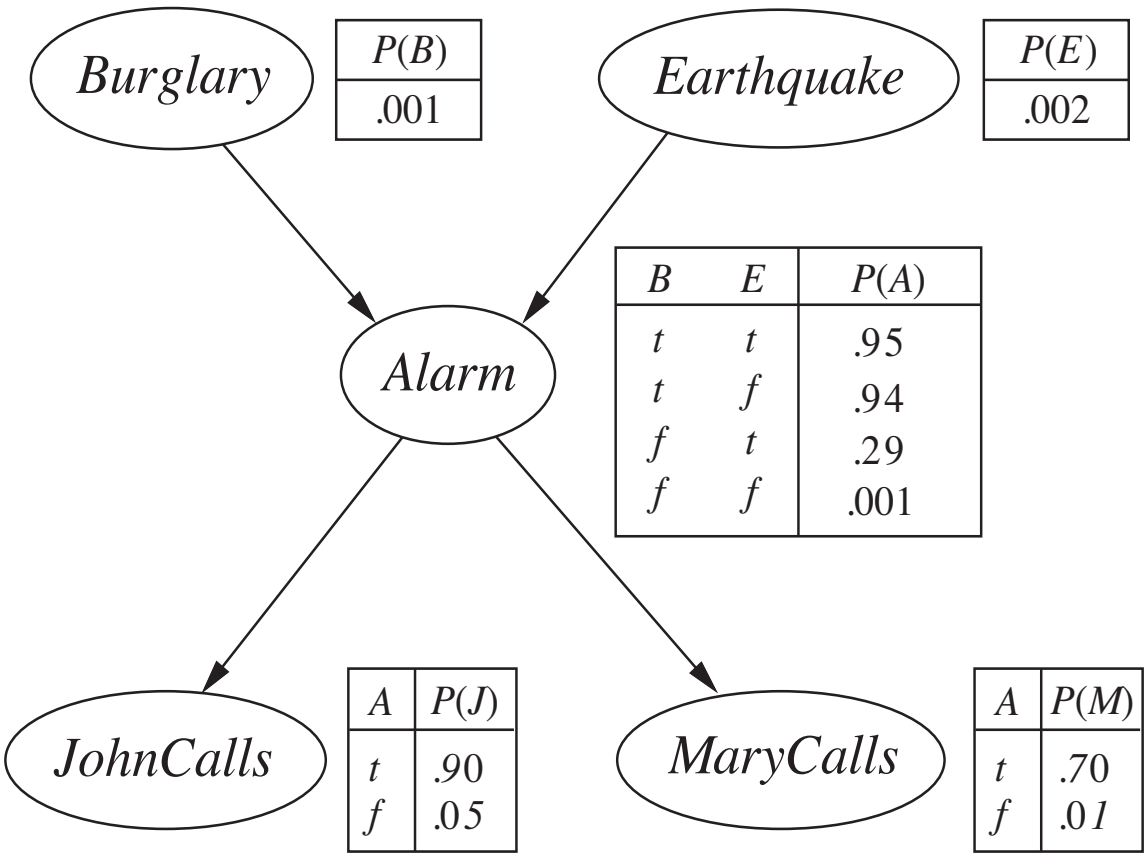
Terminology: If there is an edge $(X, Y) \in E$, X is called the **parent** of Y and Y is called the **child** of X. $Parents(X)$ will be used to denote the parents of a node X.

Bayesian Networks (cont'd)

- It is usually easy for a domain expert to decide what **direct causal influences** exist in a domain.
- Once the topology of the Bayesian network is specified, we can also specify the probabilities themselves (more difficult!).

  For discrete variables, this will be done by giving a **conditional probability table (CPT)** for each variable Y. This table specifies the conditional distribution $P(Y \mid Parents(Y))$.
- The combination of network topology and conditional distributions **specifies implicitly the full joint distribution for all variables**.

Example


Burglary
P(B)
.001
Earthquake
P(E)
.002
Alarm
B E P(A)
t t .95
t f .94
f t .29
f f .001
JohnCalls
A P(J)
t .90
f .05
MaryCalls
A P(M)
t .70
f .01

Comments

- Each row of a CPT gives the **conditional probability of each node value given a conditioning case**. The values in each row must sum to 1.
- For Boolean variables, we only give the probability p of the value *true*. The probability of the value $false$ is $1 - p$ and is omitted.
- A CPT for a Boolean variable with k Boolean parents contains 2^k independently specifiable probabilities.

  Question: What is the formula for random variables in general?
- A node with no parents has only one row, representing the **prior probabilities** of each possible value of the variable.

Semantics of Bayesian Networks

From a Bayesian network, we can compute the **full joint probability distribution** of random variables $X_1, \ldots, X_n$ using the formula

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

where $parents(X_i)$ denotes the values of $Parents(X_i)$ that appear in $x_1, \ldots, x_n$.

Example:

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

$$= 0.90 \times 0.70 \times 0.001 \times 0.999 \times 0.998 = 0.000628$$

Constructing Bayesian Networks

Using the product rule, the joint probability distribution of variables $X_1, \ldots, X_n$ can be written as:

$$P(x_1, \ldots, x_n) = P(x_n \mid x_{n-1}, \ldots, x_1)\ P(x_{n-1}, \ldots, x_1) =$$

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

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

This equation is called the **chain rule**.

Constructing Bayesian Networks (cont'd)

Now compare the formula we just computed with the one that gives us the semantics of Bayesian networks:

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

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

Constructing Bayesian Networks (cont'd)

If, for every variable X_i, we have

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

provided that

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

then the Bayesian network is indeed a **correct representation of the joint probability distribution.**

Constructing Bayesian Networks (cont'd)

Let us assume that the nodes of a Bayesian network are ordered as $X_1, \ldots, X_n$ with children following parents in the ordering.

The equation

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

says that **node X_i is conditionally independent of its other predecessors in the node ordering, given its parents.**

We can satisfy this condition with the following methodology.

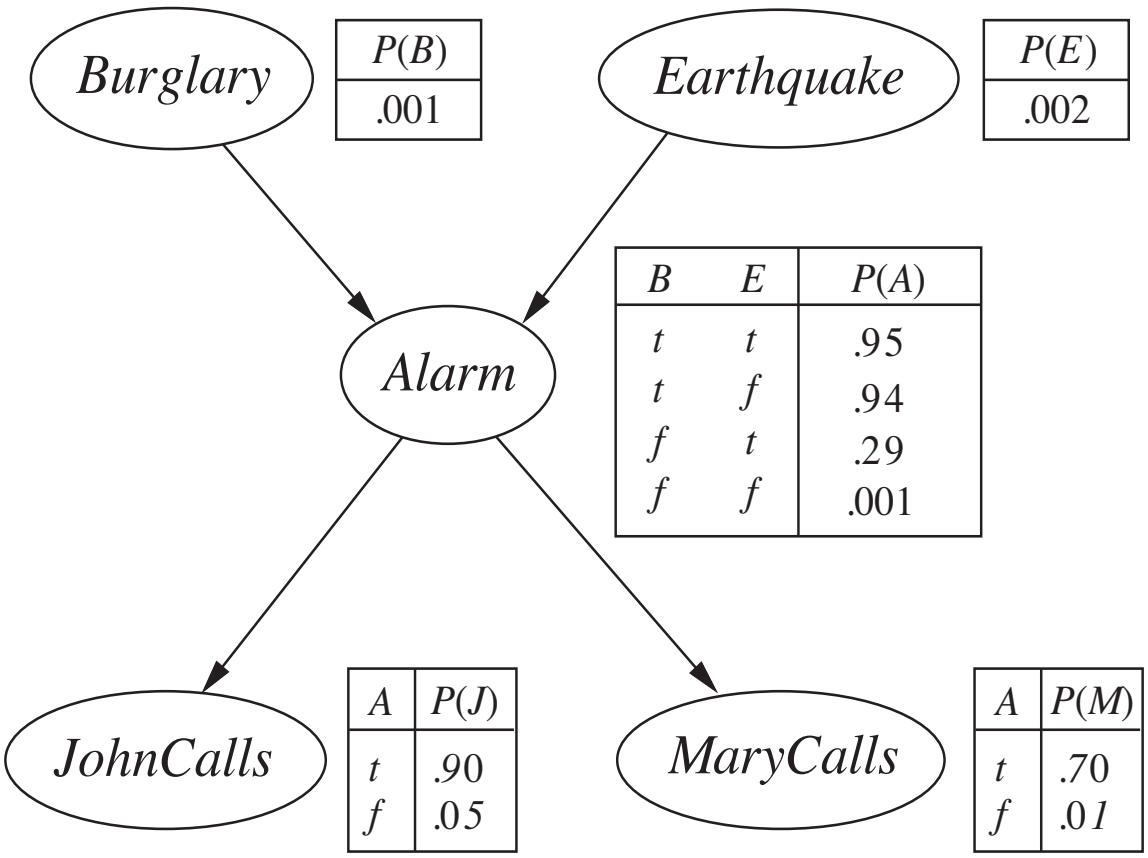
Constructing Bayesian Networks: A Simple Methodology

1. **Nodes:** Determine the set of variables that are required to model the domain. Order them as $X_1, \ldots, X_n$ such that causes precede effects.
2. **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 equation

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

     is satisfied.
   - For each parent, insert a link from the parent to X_i.
   - Write down the CPT specifying $P(X_i \mid Parents(X_i))$.

Example


Burglary
P(B)
.001
Earthquake
P(E)
.002
Alarm
B E P(A)
t t .95
t f .94
f t .29
f f .001
JohnCalls
A P(J)
t .90
f .05
MaryCalls
A P(M)
t .70
f .01

Example (cont'd)

How did we construct the previous network given our knowledge of the domain?

The difficulty is in Step 2 where we should specify the parents of a node X_i i.e., all the nodes that **directly influence** X_i.

Suppose we have completed the previous network except for the choice of parents for $MaryCalls$. The important thing to notice that $Burglary$ or $Earthquake$ influence $MaryCalls$ but **not directly**. Only $Alarm$ influences $MaryCalls$ directly. Similarly, $JohnCalls$ has no influence on $MaryCalls$.

Formally:

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

$$\mathbf{P}(MaryCalls \mid Alarm)$$

Compactness of Bayesian Networks

Bayesian networks are **compact representations** of causality and independence relationships.

Let us assume Boolean random variables for simplicity. If each variable is influenced by at most k ($k \ll n$) other variables then:

- The complete Bayesian network requires the specification of $n2^k$ probabilities.
- The joint distribution contains 2^n probabilities.

Inference in Bayesian Networks

We will now present **algorithms for probabilistic inference** in Bayesian networks:

- Exact algorithms
- Approximate algorithms (since the worst-case complexity of the exact algorithms is exponential).

Probabilistic Inference using Bayesian Networks

Let X be the query variable, $\mathbf{E}$ be the vector of evidence variables, $\mathbf{e}$ the vector of observed values and $\mathbf{Y}$ the vector of the remaining variables (called **hidden variables**).

The typical query is $\mathbf{P}(X \mid \mathbf{e})$ and can be computed by

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

where $\alpha = 1/\mathbf{P}(\mathbf{e})$.

Probabilistic Inference (cont'd)

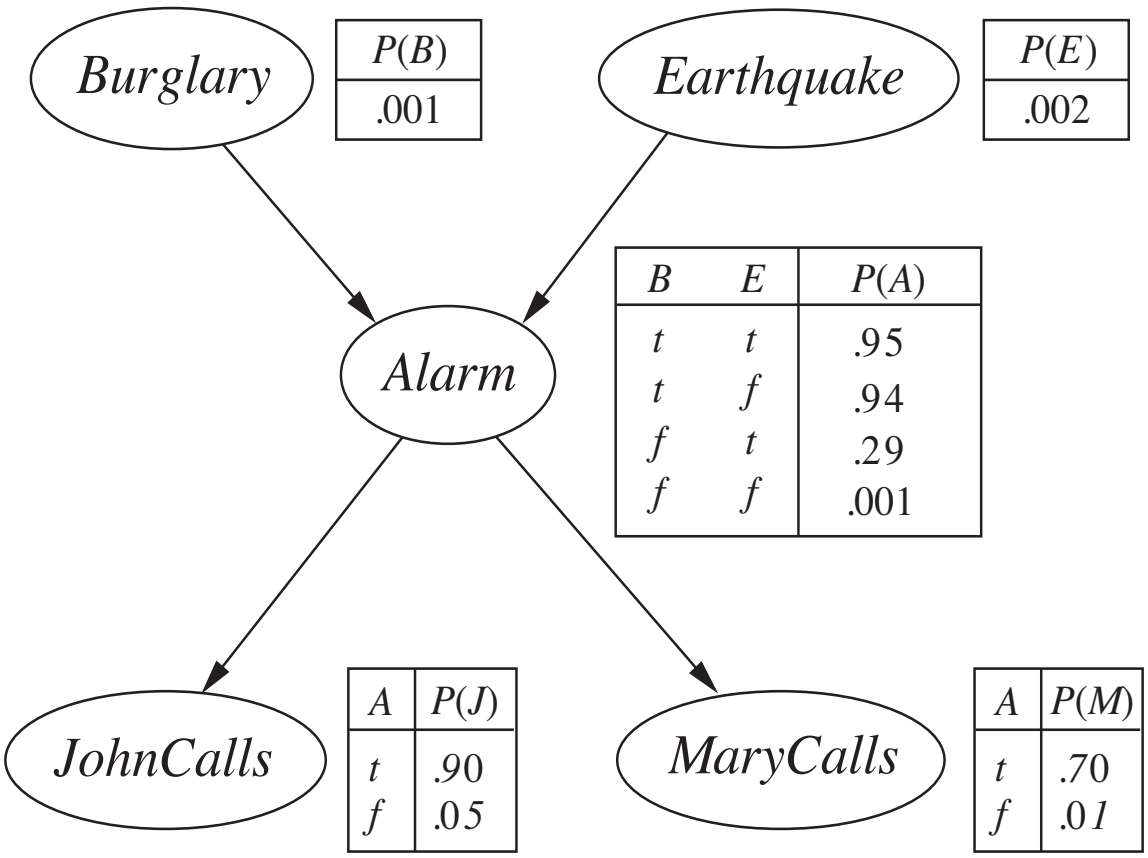
We will use the equation that gives the semantics of Bayesian networks

$$P(X_1, \ldots, X_n) = \prod_{i=1}^{n} P(X_i \mid Parents(X_i))$$

to compute the terms $\mathbf{P}(X, \mathbf{e}, \mathbf{y})$.

Thus, the computation involves **computing sums of products** of conditional probabilities from the network.

Example


Burglary
P(B)
.001
Earthquake
P(E)
.002
Alarm
B E P(A)
t t .95
t f .94
f t .29
f f .001
JohnCalls
A P(J)
t .90
f .05
MaryCalls
A P(M)
t .70
f .01

Example (cont'd)

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

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

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

Complexity: In a network with n Boolean variables, where we have to sum out almost all variables, the complexity of this computation is $O(n2^n)$.

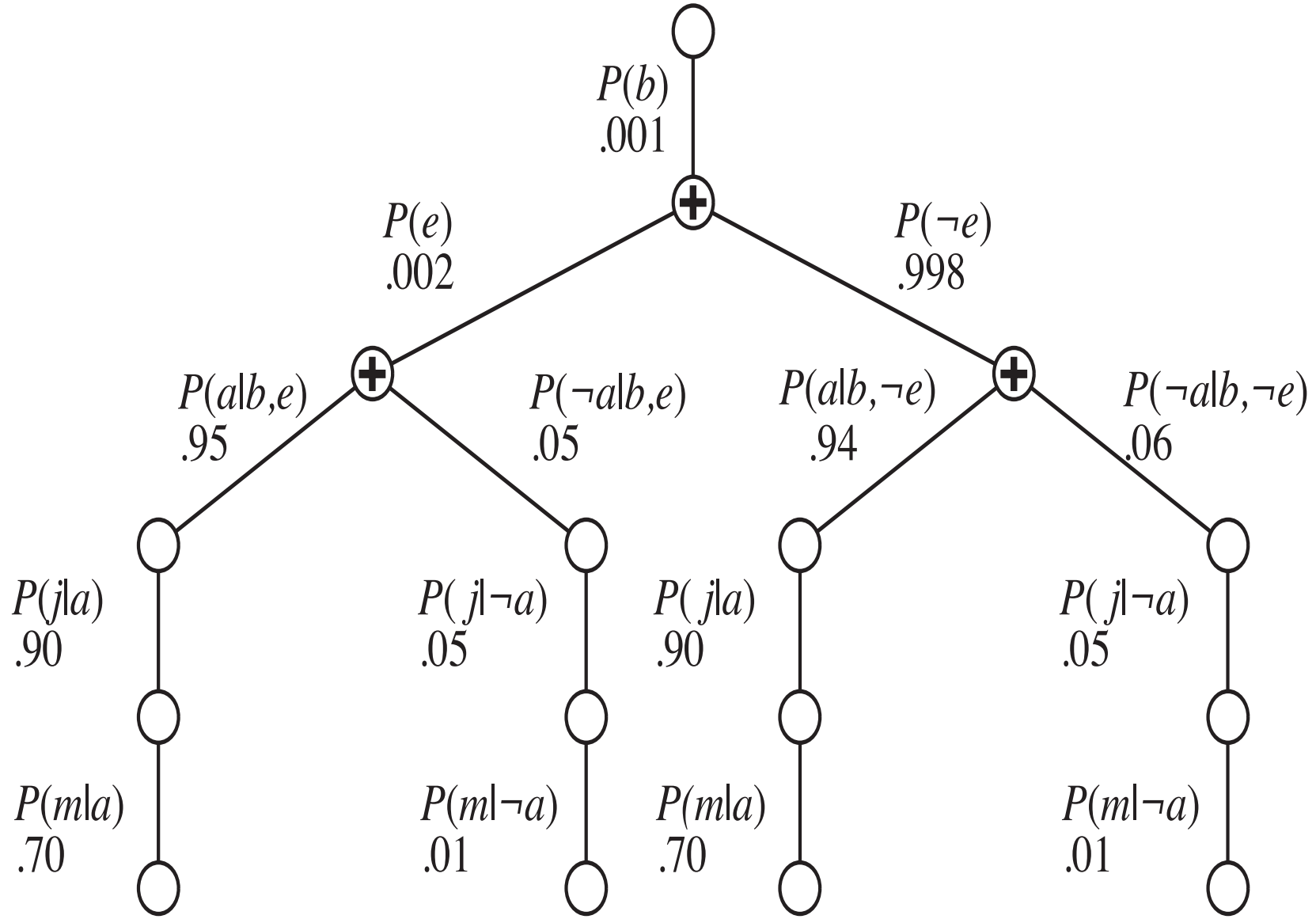
Example (cont'd)

We can do better by noticing which probabilities are constants in the summations:

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

This computation is shown graphically in the next figure.

Example (cont'd)



Example (cont'd)

Finally:

$$P(b \mid j, m) = \alpha \times 0.00059224 \approx 0.284$$

The chance of a burglary given calls from both John and Mary is about 28%.

The Enumeration Algorithm

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 Bayesian network with variables $\{X\} \cup \mathbf{E} \cup \mathbf{Y}$
(with hidden variables $\mathbf{Y}$)

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

The variables in the list bn.VARS are given in the order $\mathbf{Y}, \mathbf{E}, \{X\}$.

The Enumeration Algorithm (cont'd)

function ENUMERATION-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 **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 $Y = y$.

The Enumeration Algorithm (cont'd)

The enumeration algorithm uses **depth-first recursion** and is very similar in structure with the backtracking algorithm for CSPs.

Evaluation:

- **Time complexity:** $O(2^n)$
- **Space complexity:** $O(n)$
- **Inefficiency:** We are repeatedly evaluating the same subexpressions (in the above example, the expressions $P(j|a)P(m|a)$ and $P(j|\neg a)P(m|\neg a)$, once for each value of e). How can we avoid this?

Idea

- **Use dynamic programming:** do the calculation once and save the results for later reuse.
- There are several versions of this approach. We will present the **variable elimination** algorithm.

Variable Elimination

The variable elimination algorithm works as follows:

- It **factorizes the joint probability distribution** corresponding to a Bayesian network into a list of **factors** from which we can reconstruct the distribution.
- It operates on this list of factors repeatedly, **eliminating all the hidden variables** of the network. This operation preserves the property of the original list of factors that they can be used to construct the joint probability distribution of the involved variables.
- It uses the resulting list of factors to compute the final joint distribution of interest.

Factors

Example: Let us calculate the probability distribution $\mathbf{P}(B \mid j, m)$ using the burglary network.

$$\mathbf{P}(B \mid j, m) = \alpha \underbrace{\mathbf{P}(B)}_{\mathbf{f}_1(B)} \sum_e \underbrace{P(e)}_{\mathbf{f}_2(E)} \sum_a \underbrace{\mathbf{P}(a \mid B, e)}_{\mathbf{f}_3(A,B,E)} \underbrace{P(j \mid a)}_{\mathbf{f}_4(A)} \underbrace{P(m \mid a)}_{\mathbf{f}_5(A)}$$

The expressions $\mathbf{f}_i(\cdot)$ in the above formula are called **factors**.

Factors (cont'd)

$$\mathbf{P}(B \mid j, m) = \alpha \underbrace{\mathbf{P}(B)}_{\mathbf{f}_1(B)} \sum_e \underbrace{P(e)}_{\mathbf{f}_2(E)} \sum_a \underbrace{\mathbf{P}(a \mid B, e)}_{\mathbf{f}_3(A,B,E)} \underbrace{P(j \mid a)}_{\mathbf{f}_4(A)} \underbrace{P(m \mid a)}_{\mathbf{f}_5(A)}$$

Definition. Each factor $\mathbf{f}(X_1, \ldots, X_n)$ is a **matrix** of dimension $v_1 \times \cdots \times v_n$ with numeric values (probabilities), where v_i is the cardinality of the domain of random variable X_i.

Examples: The factor $\mathbf{f}_1(B)$ is a single column with two rows. The factor $\mathbf{f}_3(A, B, E)$ is a $2 \times 2 \times 2$ matrix.

Factors (cont'd)

Let $\mathbf{f}(X_1, \ldots, X_n)$ be a factor. **Setting** the variable (e.g., X_1) in $\mathbf{f}(X_1, \ldots, X_n)$ to a particular value (e.g., $X_1 = a$) gives us a **new factor** which is a function of variables $X_2, \ldots, X_n$.

Example: Because j is fixed by the previous query, factor $\mathbf{f}_4$ can be obtained from a factor $\mathbf{f}(J, A)$ by setting $J = j$. Factor $\mathbf{f}_4$ depends only on A:

$$\mathbf{f}_4(A) = \begin{pmatrix} P(j|a) \\ P(j|\neg a) \end{pmatrix} = \begin{pmatrix} 0.90 \\ 0.05 \end{pmatrix}$$

Factorizing the Joint Probability Distribution

Using factors, the previous expression can be written equivalently as:

$$\mathbf{P}(B \mid j, m) = \alpha\ \mathbf{f}_1(B) \times \sum_e \mathbf{f}_2(E) \times \sum_a \mathbf{f}_3(A, B, E) \times \mathbf{f}_4(A) \times \mathbf{f}_5(A)$$

where $\times$ is the **pointwise product** of matrices (i.e., the probability distribution has been factorized).

We evaluate this expression by **summing out variables (right to left)** from pointwise products of factors **to produce new factors**, eventually yielding a factor that is the solution (i.e., the posterior distribution of the query variable).

Example

$$\mathbf{P}(B \mid j, m) = \alpha\ \mathbf{f}_1(B) \times \sum_e \mathbf{f}_2(E) \times \sum_a \mathbf{f}_3(A, B, E) \times \mathbf{f}_4(A) \times \mathbf{f}_5(A)$$

First we sum out A to get a new 2×2 factor $\mathbf{f}_6(B, E)$:

$$\mathbf{f}_6(B, E) = \sum_a \mathbf{f}_3(A, B, E) \times \mathbf{f}_4(A) \times \mathbf{f}_5(A) =$$

$$= (\mathbf{f}_3(a, B, E) \times \mathbf{f}_4(a) \times \mathbf{f}_5(a)) + (\mathbf{f}_3(\neg a, B, E) \times \mathbf{f}_4(\neg a) \times \mathbf{f}_5(\neg a))$$

Thus we arrive at:

$$\mathbf{P}(B \mid j, m) = \alpha\ \mathbf{f}_1(B) \times \sum_e \mathbf{f}_2(E) \times \mathbf{f}_6(B, E)$$

Example (cont'd)

Then we sum out E to get $\mathbf{f}_7(B)$:

$$\mathbf{f}_7(B) = \sum_e \mathbf{f}_2(E) \times \mathbf{f}_6(B, E) = (\mathbf{f}_2(e) \times \mathbf{f}_6(B, e)) + (\mathbf{f}_2(\neg e) \times \mathbf{f}_6(B, \neg e))$$

Thus we arrive at:

$$\mathbf{P}(B \mid j, m) = \alpha \ \mathbf{f}_1(B) \times \mathbf{f}_7(B)$$

This formula can be evaluated by taking the pointwise product and then normalizing.

Question: How do we perform the **pointwise products** or the matrix additions above?

Pointwise Product

Definition. The **pointwise product** of two factors $\mathbf{f}_1$ and $\mathbf{f}_2$ is a new factor $\mathbf{f}$ whose variables is the union of the variables in $\mathbf{f}_1$ and $\mathbf{f}_2$ and whose elements are given by the product of the corresponding elements in the two factors. In other words:

$$\mathbf{f}(X_1, \ldots, X_j, Y_1, \ldots, Y_k, Z_1, \ldots, Z_l) =$$

$$\mathbf{f}_1(X_1, \ldots, X_j, Y_1, \ldots, Y_k)\ \mathbf{f}_2(Y_1, \ldots, Y_k, Z_1, \ldots, Z_l)$$

If all the variables are Boolean, then $\mathbf{f}_1$ and $\mathbf{f}_2$ have 2^{j+k} and 2^{k+l} entries respectively, and the pointwise product $\mathbf{f}$ has 2^{j+k+l} entries.

Example (cont'd)

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

Example (cont'd)

A	B	C	$\mathbf{f}_3(A, B, C)$
T	T	T	$.3 \times .2 = .06$
T	T	F	$.3 \times .8 = .24$
T	F	T	$.7 \times .6 = .42$
T	F	F	$.7 \times .4 = .28$
F	T	T	$.9 \times .2 = .18$
F	T	F	$.9 \times .8 = .72$
F	F	T	$.1 \times .6 = .06$
F	F	F	$.1 \times .4 = .04$

Complexity Issues

- The result of a pointwise product can contain more variables than the input factors.
- The **size** of a factor is **exponential in the number of variables**.
- This is where both **time** and **space** complexity arise in the variable elimination algorithm.

Summing Out Variables

As we have seen above, summing out a variable from a product of factors is done by **adding the submatrices** formed by fixing the variable to each of its values in turn. This operation is matrix addition as we know it from linear algebra.

Example:

$$\mathbf{f}(B,C) = \sum_a \mathbf{f}_3(A,B,C) = \mathbf{f}_3(a,B,C) + \mathbf{f}_3(\neg a,B,C) =$$

$$\begin{pmatrix} .06 & .24 \\ .42 & .28 \end{pmatrix} + \begin{pmatrix} .18 & .72 \\ .06 & .04 \end{pmatrix} = \begin{pmatrix} .24 & .96 \\ .48 & .32 \end{pmatrix}$$

Summing Out Variables (cont'd)

Any factor that does not depend on the variable to be summed out can be **moved outside the summation**. For example:

$$\sum_e \mathbf{f}_2(E) \times \mathbf{f}_3(A, B, E) \times \mathbf{f}_4(A) \times \mathbf{f}_5(A) =$$

$$\mathbf{f}_4(A) \times \mathbf{f}_5(A) \times \sum_e \mathbf{f}_2(E) \times \mathbf{f}_3(A, B, E)$$

The Variable Elimination Algorithm

function ELIMINATION-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 Bayesian network specifying the joint distribution $\mathbf{P}(X_1, \ldots, X_n)$

$\mathcal{F} \leftarrow$ a list of factors corresponding to bn with evidence variables
set to their observed values.
for each Y **in** ORDER(bn.HIDDENVARS) **do**
Delete from $\mathcal{F}$ all factors containing variable Y and add the factor
SUM-OUT($Y, \mathcal{F}$) to it.
return NORMALIZE(POINTWISE-PRODUCT($\mathcal{F}$))

The Variable Elimination Algorithm (cont'd)

function SUM-OUT($Y, \mathcal{F}$) **returns** a factor over $\mathbf{Z} \setminus \{Y\}$
inputs Y, the variable to be eliminated
$\mathcal{F}$, a list of factors containining variables $\mathbf{Z}$

return the factor $\sum_y \prod_{i=1}^{k} \mathbf{f}_i$ where $\mathbf{f}_1, \ldots, \mathbf{f}_k$ are the factors of $\mathcal{F}$ containing variable Y.

Note: The symbol $\prod$ in the above algorithm denotes the **pointwise product** operation for factors.

Correctness

Theorem. Let $\mathbf{P}(Z_1, \ldots, Z_m)$ be a joint probability distribution factorized into the pointwise product of a list $\mathcal{F}$ of factors. After performing the variable elimination operation for variable Z_1, the resulting list of factors (**for** loop in function ELIMINATION-ASK) has pointwise product the joint probability distribution $\mathbf{P}(Z_2, \ldots, Z_m)$.

Proof. Suppose $\mathcal{F}$ consists of $\mathbf{f}_1, \ldots, \mathbf{f}_k$ and suppose Z_1 appears only in factors $\mathbf{f}_1, \ldots, \mathbf{f}_l$ where $l \leq k$. Then

$$P(Z_2, \ldots, Z_m) = \sum_{z_1} P(Z_1, Z_2, \ldots, Z_m) = \sum_{z_1} \prod_{i=1}^{k} f_i =$$

$$(\sum_{z_1} \prod_{i=1}^{l} f_i)(\prod_{i=l+1}^{k} f_i)$$

Correctness (cont'd)

The first term in the above result is the factor added to list $\mathcal{F}$ in the place of the factors deleted, while the second term is the pointwise product of the factors that remained in the list.

Complexity Issues

- Function ORDER: **any ordering of the variables will work**, but different orderings cause different intermediate factors to be generated.

  Example: In the previous example, if we eliminate E and then A, the computation becomes

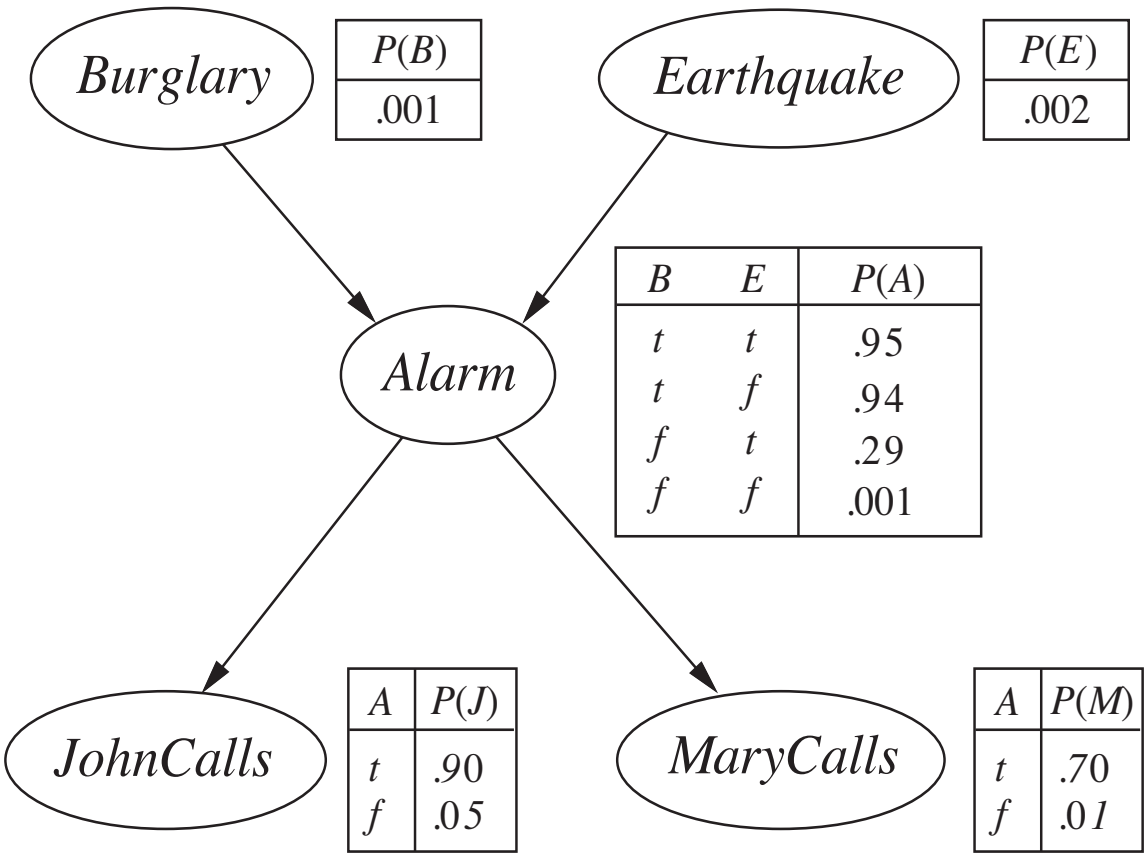
  $$\mathbf{P}(B \mid j, m) = \alpha\, \mathbf{f}_1(B) \times \sum_a \mathbf{f}_4(A) \times \mathbf{f}_5(A) \times \sum_e \mathbf{f}_2(E) \times \mathbf{f}_3(A, B, E)$$

  during which a new factor $\mathbf{f}_6(A, B)$ is generated.

Complexity Issues (cont'd)

- The **time** and **space** complexity of variable elimination is **dominated by the size of the largest factor** constructed. This in turn is determined by the order of elimination of variables and by the structure of the network.
- It is **intractable** to determine an optimal ordering. One good **heuristic** is the following: eliminate whichever variable minimizes the size of the next factor to be constructed.

Example


Burglary
P(B)
.001
Earthquake
P(E)
.002
Alarm
B E P(A)
t t .95
t f .94
f t .29
f f .001
JohnCalls
A P(J)
t .90
f .05
MaryCalls
A P(M)
t .70
f .01

Example

Let us compute $\mathbf{P}(JohnCalls \mid Burglary = true)$:

$$\mathbf{P}(J \mid b) = \alpha \; P(b) \sum_e P(e) \sum_a P(a \mid b, e) \; \mathbf{P}(J \mid a) \sum_m P(m \mid a)$$

Notice that $\sum_m P(m \mid a)$ is 1 i.e., the variable M is irrelevant to the query.

Removal of Irrelevant Variables

We can remove any **leaf node that is not a query variable or evidence variable** from the network. Then, we might have some more leaf nodes that are irrelevant so they should be removed as well.

The result of this iterative process is:

- Every variable that is not an ancestor of a query variable or an evidence variable is **irrelevant** to the query.

Thus we should **remove these variables** before evaluating the query by variable elimination.

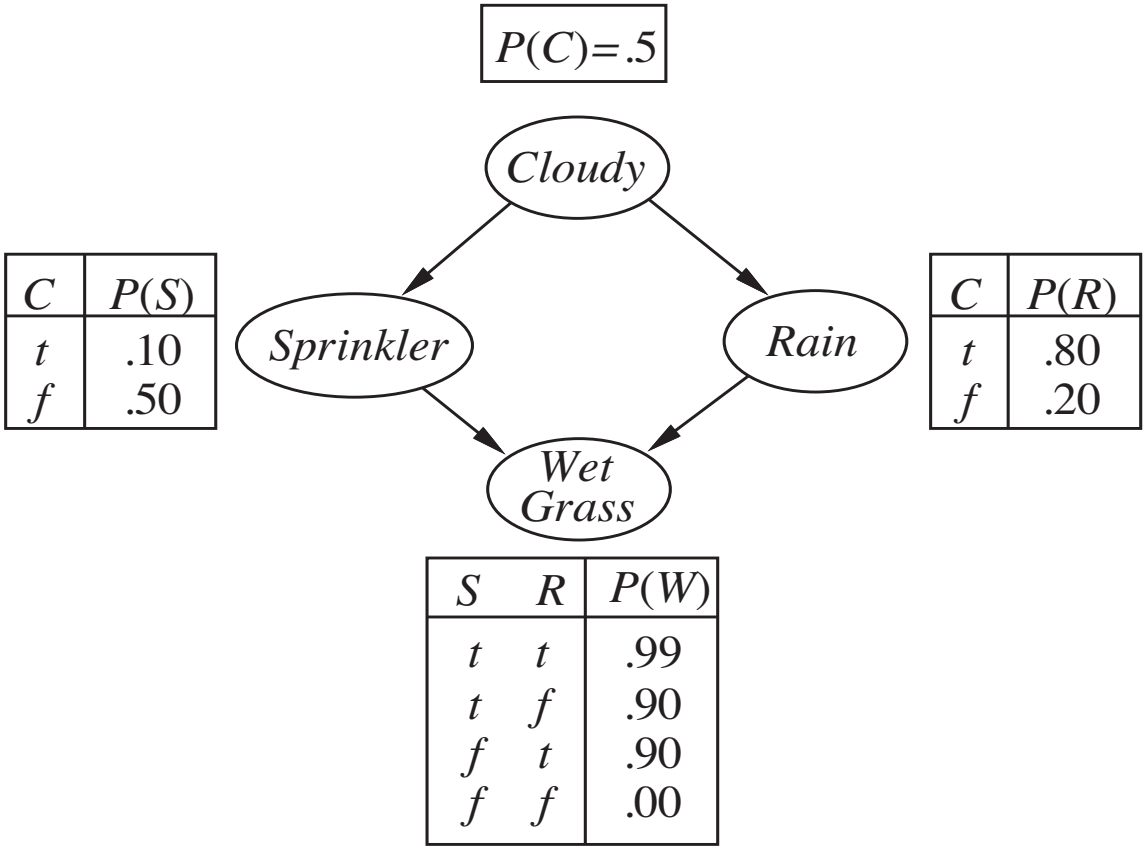
Computational Complexity of Exact Inference

- Bayesian network inference **includes propositional logic inference** as a special case (why?). Therefore, it is NP-hard.
- Bayesian network inference is an **#P-hard problem** i.e., as hard as the problem of computing the satisfying assignments of a propositional logic formula. These problems are strictly harder than NP-complete problems.
- There are interesting **tractable cases**.

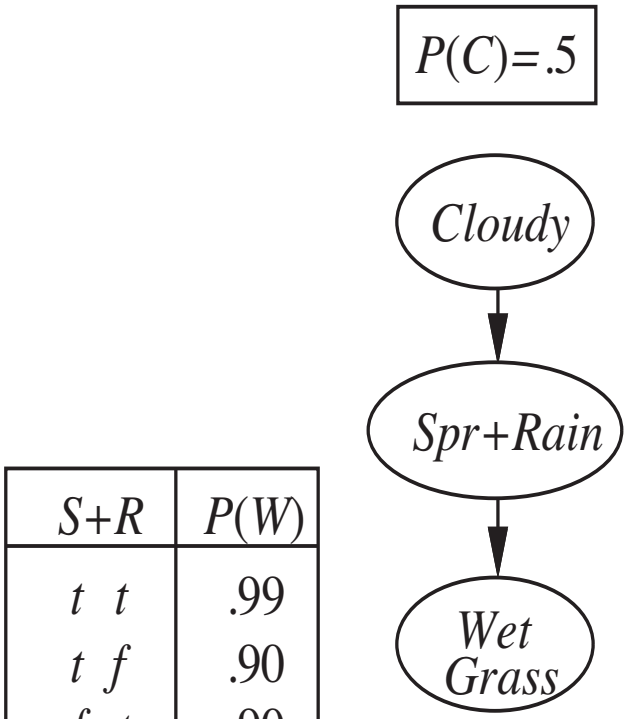
Complexity of Exact Inference (cont'd)

- A Bayesian network is **singly connected** (or **polytree**) if there is at most one undirected path between any two nodes.
  Example: the burglary network
- For singly connected networks, Bayesian network inference is **linear** in the maximum number of CPT entries. If the number of parents of each node is bounded by a constant, then the complexity is also linear in the number of nodes.
- There are **close connections among CSPs and Bayesian networks** (complexity, tractable special cases, variable elimination algorithms).
- There are efficient (but still worst-case exponential) techniques that transform **multiply connected** networks into polytrees and then run specialized inference algorithms on them.

Example of Multiply Connected Network



A Clustered Equivalent



C	P(S+R=x) t t	t f	f t	f f
t	.08	.02	.72	.18
f	.10	.40	.10	.40

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

Approximate Inference in Bayesian Networks

There are **randomized sampling (Monte Carlo)** algorithms that provide approximate answers to Bayesian inference problems:

- Direct and rejection sampling
- Likelihood weighting
- Markov chain Monte Carlo
- Gibbs sampling

See AIMA for more details.

How do we Construct Bayesian Networks

In related theory and practice, the following methods have been used:

- Using expert knowledge
- Automatic synthesis from some other formal representation of domain knowledge (e.g., in reliability analysis or diagnosis of hardware systems).
- Learning from data (e.g., in medicine).

First-Order Logic and Bayesian Networks

- Bayesian networks can be seen as a **probabilistic extension of propositional logic**.
- The combination of the expressive power of first-order logic and Bayesian networks would increase dramatically the range of problems that we can model.
- Various authors have studied the combination of Bayesian networks and first-order logic representations but much remains to be done in this area.

Applications of Bayesian Networks

- Medicine
- Engineering (e.g., monitoring power generators).
- Networking (e.g., network tomography)
- Diagnosis-and-repair tools for Microsoft software.

  Check out the tool MSBNx at `http://research.microsoft.com/en-us/um/redmond/groups/adapt/msbnx/`.
- Bioinformatics
- ...

Many references are given in Chapter 14 of AIMA.

Other Approaches to Uncertain Reasoning

- Rule-based techniques
- Dempster-Shafer theory
- Fuzzy sets and fuzzy logic

Rule-based Techniques

- Rule-based systems were used for uncertain reasoning in the 1970's. MYCIN, an **expert system** for medical diagnosis, used the **certainty factors** model.
- In the cases that worked well, the techniques of MYCIN were essentially the predecessors of Bayesian network inference on polytrees. But in many other cases, these techniques did not work well.
- In general, the properties of **locality, detachment** and **truth-functionality** of rule-based techniques are not appropriate for reasoning with probabilities.

The Dempster-Shafer Theory

- The Dempster-Shafer theory distinguishes between **uncertainty** and **ignorance**.
- In this theory, one does not compute the probability of a proposition but rather **the probability that the evidence supports the proposition.**

  This is a measure of belief denoted by $Bel(X)$.

Example

Question: A magician gives you a coin. Given that you do not know whether it is fair, what belief should you ascribe to the event that it comes up heads?

Answer: According to Dempster-Shafer theory, $Bel(Heads) = 0$ and $Bel(\neg Heads) = 0$.

This is an intuitive property of Dempster-Shafer reasoning systems.

Example (cont'd)

Question: Now suppose you have an expert who testifies that he is 90% sure that the coin is fair (i.e., $P(Heads) = 0.5$). How do your beliefs change now?

Answer: According to Dempster-Shafer theory, $Bel(Heads) = 0.9 \times 0.5 = 0.45$ and $Bel(\neg Heads) = 0.45$. There is a 10% point "gap" that is not accounted by the evidence.

This is not a good property of Dempster-Shafer theory.

Fuzzy Sets and Fuzzy Logic

- **Fuzzy set theory** is a way of specifying how well an object satisfies a **vague description**.

  Example: John is tall.

- **Fuzzy logic** is a method of reasoning with logical expressions describing membership in fuzzy sets.

- **Fuzzy control** systems have applied fuzzy logic in many commercial products (e.g., video cameras, other household appliances).

Fuzziness and uncertainty as we have studied it are **orthogonal** issues.

Readings

- Chapter 13 and 14 of AIMA.

  I used the 3rd edition of the book for these slides, but the relevant chapters are the same in the 2nd edition.

- The **variable elimination algorithm** comes from Sections 2 and 3 of the paper:

  Nevin Lianwen Zhang and David Poole. *Exploiting Causal Independence in Bayesian Network Inference.* Journal of Artificial Intelligence Research, vol. 5, pages 301-328, 1996. Available from
  `http://www.jair.org/media/305/live-305-1566-jair.pdf`

Readings (cont'd)

- The variable elimination algorithm presented here is essentially an instance of Rina Dechter's **bucket elimination algorithm** applied to Bayesian networks. See the paper

  Rina Dechter. *Bucket elimination: A unifying framework for reasoning.* Artificial Intelligence. Volume 113, pages 4185, 1999. Available from `http://www.ics.uci.edu/~csp/r76A.pdf`

  for a detailed presentation of this general algorithm which is also applied to other problem domains e.g., in CSPs, linear inequalities, propositional satisfiability etc.

Readings (cont'd)

- See also the recent survey paper on Bayesian Networks:

  Adnan Darwiche. *Bayesian Networks.* Communications of the ACM. vol. 53, no. 12, December 2010. Available from `http://reasoning.cs.ucla.edu/fetch.php?id=104&type=pdf`