

Lecture 9

Probabilistic Reasoning

Probability is important in many branches of mathematics and science, and AI is not an exception. Many tasks require us to evaluate probabilities.

This lecture explains the basic probability framework, discuss a representation which succinctly describe probabilistic knowledge, and show how probabilistic reasoning can take place.

Reference:

- Textbook Chapter 13 and 14

AI(0270)

Suppose we want to have a system to diagnose dental problems.

- We would like to have a rule like this:
 $Symptom(p, Toothache) \Rightarrow Disease(p, Cavity)$
- But we **can't**. There are **many other reasons to have toothache**, and just having toothache is not enough for deducing cavity.
- So we are forced to write something like...
 $Symptom(p, Toothache) \Rightarrow Disease(p, Cavity) \vee Disease(p, GumDisease) \vee Disease(p, ImpactedWisdom) \vee \dots$
- But... it is difficult, since there are really so many reasons.
- How about 'causal rules', e.g., $Disease(p, Cavity) \Rightarrow Symptom(p, Toothache)$?
- Still incorrect (although much better), and difficult to use.

AI(0270)-9.1

Sources of uncertainty

So instead of saying the very long sentence, we would like to write
 $Symptom(p, Toothache) \Rightarrow Disease(p, Cavity) \vee \dots \vee RareEvent(p)$

But now we have a difficulty: **we never know that a rare event occurs!**

We encounter uncertainty in such systems because...

- we are **lazy** to write a rule that is completely correct, and even if we do, the result will be so **large** that it becomes unusable.
- we are **ignorant** about some of the causes and consequence relationship, so we can only **approximate** the physical world.
- we cannot administer all the **tests** required to obtain all information to give a completely correct inference. Doing so would be either impossible or too expensive.

AI(0270)-9.2

Using probability to model uncertainty

- Probability solves the problem by using a number to **summarizes** all the uncertainty we meet.
- So we might say that there is a 80% chance that a patient has cavity if he has a toothache. The 20% summarize all the other possibilities.
- In Propositional (and First-Order) Logic, everything is either true or false. It is still true in our probabilistic world.
- In Propositional (and First-Order) Logic, if we don't think something true or false, we have to say we "don't know anything about it".
- When doing probabilistic reasoning, we want to be able to believe something "is likely to be true" or "is likely to be false". So it **extends** FOL.
We use only proposition—using First-Order statements with probability is quite complicated.

AI(0270)-9.3

Why probability is important in AI?

Probability is needed in nearly all realistic AI (or scientific) applications. We have seen that some games are probabilistic. Some other examples...

- If you'd just have a sensor to detect the robot, it is possible that sometimes the sensor misbehave, and you never know when.
- If you'd like to understand a natural language sentence, there are a lot of ambiguity and you must use contextual and common-sense to choose the most probable one.
- For a speech recognition system (turning sounds to words), usually the sound of the speaker has a little distortion, and your only hope to acquire human-like accuracy is to use sounds around to guess what is the most likely word spoken.
- Even in domains as simple as the Wumpus world, only probability can guide us about what to do when there is no "safe" way out.

AI(0270)-9.4

What the probabilities means?

Why we say that 80%, rather than 75% or 5%?

- By the 80%, we mean that **if similar things happen many times**, 80% of the time the patient has cavity.
- Such knowledge can be collected **by using a large database of past experience**, or **combined** from different sources of information.
- In cases that past experience is not available or is too small, we might need to ask a domain expert to **supply a subjective estimate**.

But be careful...

- A probability makes sense **only if we know what are the evidences!**
- E.g., we might say that the probability of a random people to have cavity to be 1/2, but if we add the evidence that he go to the dentist, it is no longer 1/2, or that he complains about toothache!

AI(0270)-9.5

Notation: Conditional Probability

This poses a difficulty: we **cannot** simply say that “the probability to have cavity is 1/4”. We have to be more specific about the evidences. We use the following notation:

- $P(\text{Cavity}=\text{true} \mid \text{Toothache}=\text{true})$: the **conditional probability** that the propositional sentence *Cavity* is true, given that *Toothache* is true.
- We will abbreviate it as $P(\text{cavity} \mid \text{toothache})$.
- In probability theory, *Cavity* is said to be a **random variable**, and that $\text{Cavity} = \text{true}$ (or *false*) is said to be an **event**.
- If we have no evidence, then the evidence part must be trivially true, e.g., $P(\text{cavity} \mid \text{true})$.
- In this case, we write it as $P(\text{cavity})$ —the **prior probability** that *Cavity* = *true* holds (i.e., when no evidence is known).
In contrast, $P(\text{cavity} \mid \text{toothache})$ is said to be posterior probability.

AI(0270)-9.6

How probabilities work

In principle, to work with probability theory, we only need three axioms and one definition:

- For any proposition A , $0 \leq P(A) \leq 1$
- $P(\text{True}) = 1$, $P(\text{False}) = 0$
- Probability of disjunction is given by
 $P(A \vee B) = P(A) + P(B) - P(A \wedge B)$
This is just the Venn diagram.
- $P(A \mid B) = P(A \wedge B) / P(B)$
E.g., if $P(A \wedge B) = 0.4$ and $P(B) = 0.5$, then $P(A \mid B) = 0.8$. It **does not** depend on $P(A)$ in any way. Intuitively, if there are 100 samples, $A \wedge B$ holds in 40 of them, and B holds in 50 of them, and we already know B occurs, then we expect to find A in 80% of these samples.

AI(0270)-9.7

Probability distribution

- For a random variable with multiple values, we usually have to know the probability of all the possible values.
- E.g., $P(\text{Weather}=\text{sunny}) = 0.3$, $P(\text{Weather}=\text{cloudy}) = 0.5$, $P(\text{Weather}=\text{raining}) = 0.2$.
They must sum to 1, due to the probability axioms.
- We say such a collection to be a **probability distribution**, written as $\mathbf{P}(\text{Weather})$. (Note the bold P.)
- It is usually written as a vector: $\mathbf{P}(\text{Weather}) = (0.3, 0.5, 0.2)$.
- We can then say something like $\mathbf{P}(\text{Weather} \mid \text{Prediction}) = \mathbf{P}(\text{Weather} \wedge \text{Prediction}) / \mathbf{P}(\text{Prediction})$.
This represents a lot of sentences, not just one. Things like $P(\text{Weather}=\text{cloudy} \mid \text{Prediction}=\text{raining}) = P(\text{Weather}=\text{cloudy} \wedge \text{Prediction}=\text{raining}) / P(\text{Prediction}=\text{raining})$.

AI(0270)-9.8

Joint probability distributions

- Random variables can be more complicated. It can, for example, be a tuple containing many values (e.g., *Weather* and *Prediction*).
- A probability distribution of the tuple containing *Weather* and *Prediction* is said to be the **joint distribution** of the two variables.
- The joint will have values like $P(V=\{\text{sunny}, \text{sunny}\}) = 0.2$, $P(V=\{\text{sunny}, \text{cloudy}\}) = 0.05$, etc.
It is usually expressed like a table. See the next page.
- In a special case, a tuple can contain **all** the simple random variables in our domain. Its distribution is said to be a **full joint probability distribution**.
- A full joint probability contains the probability that **every possible combination of events** happening.

AI(0270)-9.9

A full joint is all we need

If we have the full joint, we can find any conditional probability. E.g., if

	<i>toothache</i>	\neg <i>toothache</i>
<i>cavity</i>	0.04	0.06
\neg <i>cavity</i>	0.01	0.89

Then $P(\text{cavity}) = 0.04 + 0.06 = 0.1$.

The axiom says that we should subtract $P(\text{cavity} \wedge \neg \text{cavity})$, but that value is 0 because $\text{cavity} \wedge \neg \text{cavity} = \text{false}$ and $P(\text{false}) = 0$.

Another example: $P(\neg \text{toothache} \mid \text{cavity}) = 0.06 / 0.1 = 0.6$.

In general, to find a conditional distribution, we just **sum up all distributions** for (1) both the evidence and the variable being true, and (2) the evidence being true. The quotient is the conditional probability.

The sums have a strange name: “marginal probability”.

AI(0270)-9.10

So a big joint is enough for anything we need...

Unluckily, it is infeasible to ask for the full joint...

- Assume we have n boolean variables. Then we have 2^n numbers to specify!
So for boolean random variables, we can handle no more than 30 of them!
- And... **we have no way to estimate their values!**
The smaller the probability, the more difficult to estimate accurately. When we talk about 30 boolean variables, some of them will be as small as $1 / 2^{30}$, which is something like 10^{-9} !

A fully specified joint is **too big a requirement** for our reasoning system. We have to do with a much simpler system that can be obtained and stored more easily.

AI(0270)-9.11

Independence

The full joint contains a lot of redundant information. The key idea to cut down these redundancy is independence and conditional independence.

- We say that a proposition A is **independent** of another proposition B if we have $P(A | B) = P(A)$
- Intuitively, this means that **the probability of A does not change when we acquire the knowledge about B .**
- With the definition of $P(A | B)$, we can write it as $P(A \wedge B) = P(A)P(B)$, which is usually called the **product rule**.
- It is clear that if A is independent of B , then B is independent of A .
- Similarly, we say that two random variables X and Y are independent if $P(X | Y) = P(X)$.
I.e., $P(X=x | Y=y) = P(X=x)$ for all x and y .

AI(0270)-9.12

Conditional independence

Sometimes two variables or events are **not independent** by themselves, but if we **have the knowledge of another variable** or event, then they **become** independent.

- We say two events A and B are independent conditioned on C if $P(A | B \wedge C) = P(A | C)$.
Once we know C , then the knowledge of B does not affect the probability of A .
- E.g., suppose the dentist uses a probe to try to check whether there is a cavity. Then we have an additional event *catch*, whether the probe is caught by a glitch on a tooth during the examination.
- The events *catch* and *toothache* are of course **not** independent—it is more likely for the tooth of somebody with toothache to catch the probe, than for somebody without toothache.
- But given *cavity*, they might become independent!
 $P(\text{toothache} | \text{cavity} \wedge \text{catch}) = P(\text{toothache} | \text{cavity})$.

AI(0270)-9.13

How conditional independence helps?

Suppose we know in advance (or assume) that *toothache* and *catch* are independent on each other given *cavity*...

- We no longer need to find or store $P(\text{Cavity} \wedge \text{Toothache} \wedge \text{Catch})$.
- Instead, we only need to keep $P(\text{Cavity} \wedge \text{Toothache})$, $P(\text{Cavity} \wedge \text{Catch})$ and $P(\text{Cavity})$. The full joint can be **computed** from the “partial” joints:

$$\begin{aligned} P(\text{Cavity} \wedge \text{Toothache} \wedge \text{Catch}) &= P(\text{Toothache} | \text{Cavity} \wedge \text{Catch}) \times P(\text{Cavity} \wedge \text{Catch}) \\ &= P(\text{Toothache} | \text{Cavity}) \times P(\text{Cavity} \wedge \text{Catch}) \\ &= P(\text{Toothache} \wedge \text{Cavity}) \times P(\text{Cavity} \wedge \text{Catch}) / P(\text{Cavity}) \end{aligned}$$

- Is it an improvement? Before we need to specify 7 values, now we still need 7...
 $P(A \wedge B)$ needs to store 3 values, the 4-th can be derived from it. The 7 here comes from 2 tables with 2 values, and 1 table with 1.

AI(0270)-9.14

How it is an improvement

Let's write the equation slightly differently:

$$\begin{aligned} P(\text{Cavity} \wedge \text{Toothache} \wedge \text{Catch}) &= P(\text{Toothache} | \text{Cavity} \wedge \text{Catch}) \times P(\text{Cavity} \wedge \text{Catch}) \\ &= P(\text{Toothache} | \text{Cavity}) \times P(\text{Cavity} \wedge \text{Catch}) \\ &= P(\text{Toothache} | \text{Cavity}) \times P(\text{Catch} | \text{Cavity}) \times P(\text{Cavity}) \end{aligned}$$

To specify $P(A | B)$, we need to specify only 2 values instead of 3 (for $P(A | B)$ and for $P(A | \neg B)$, the other two derive from them).

Using the joint representation do not reveal the relation with $P(B)$ and thus waste 1 of them.

As a result, we only have to estimate and store 5 probabilities, not 7. The savings is much larger when we talk about larger domains.

There is one additional benefit: **the probabilities we need are causal probability**, i.e., probability of a consequence given the cause.

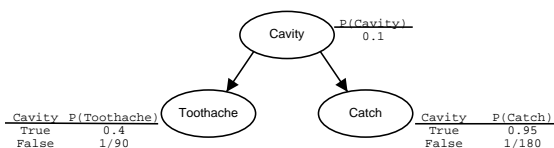
As we discussed earlier, they are easier to obtain and less likely to change.

AI(0270)-9.15

Bayesian networks

But there is one more thing to store: we need to know the conditional independence relation before we can calculate the joint that way.

We need a way to specify it—**Bayesian network**. E.g.,



The numbers are those that are required to represent $P(\text{Cavity})$, $P(\text{Toothache} | \text{Cavity})$ and $P(\text{Catch} | \text{Cavity})$. We call them **Conditional Probability Tables (CPTs)**.

What are the edges? Intuitively, *Cavity* is the primary cause of *Toothache*, and *Cavity* is the primary cause of *Catch*.

AI(0270)-9.16

The full joint view of Bayesian network

What does the graph mean exactly?

- We can view the network as a specification the **full joint**.
- The value of a full joint (i.e., given the values of all variables) is given by **multiplying** all the corresponding probabilities in the graph. E.g.,

$$\begin{aligned} P(\text{cavity} \wedge \text{toothache} \wedge \neg \text{catch}) &= 0.1 \times 0.4 \times (1 - 0.95) \\ &= 0.002 \end{aligned}$$

It can easily be seen that if we adds up all the values obtained by considering different combinations, we get 1.

- This “full-joint view” allows us to **use** the Bayesian network, we now have the full joint and can compute all conditional probabilities we ever need.
- But it doesn't directly answer us “what are the conditional independence it represents”.

AI(0270)-9.17

Reconciliation with the definition of conditional probabilities

- Suppose we list out the random variables in a way that **the parents are listed before the children**.
Usually, the causes are before the consequences.
- E.g., in our network, Cavity, Toothache, Catch.
- By repeatedly applying the “product rule” (i.e., definition of conditional probability), we know

$$P(\text{Cavity} \wedge \text{Toothache} \wedge \text{Catch}) = P(\text{Cavity})P(\text{Toothache} | \text{Cavity})P(\text{Catch} | \text{Toothache} \wedge \text{Cavity})$$
- The full joint view tells us that

$$P(\text{Cavity} \wedge \text{Toothache} \wedge \text{Catch}) = P(\text{Cavity})P(\text{Toothache} | \text{Cavity})P(\text{Catch} | \text{Cavity})$$
- So $P(\text{Catch} | \text{Toothache} \wedge \text{Cavity}) = P(\text{Catch} | \text{Cavity})$.
Toothache is independent on Catch given Cavity, as before.

AI(0270)-9.18

Conditional independence view

In general...

- If V_1, V_2, \dots, V_n are the variables in the network, in an order in which an arrow never points from V_i to V_j when $i > j$,
- then V_i is independent on V_j ($i > j$) conditioned on all parents of V_i .

In words, given all parents of a variable V_i , then V_i is independent on all its (other) predecessors.

In some sense, the parents of V_i separate all variables other than the descendants of V_i from V_i , so they become independent.

We call this the **conditional independence view** of the Bayesian network. It is equivalent to the full joint view.

AI(0270)-9.19

Building a Bayesian network

Now that we understand what Bayesian network really means, we can think about how to **build** a Bayesian network. The steps:

1. Choose the set of relevant random **variables** to describe the domain.
2. Choose an **ordering** X_1, X_2, \dots, X_n of the variables.
3. for $i = 1$ to n , do the following:
 - a. Add a node X_i to the network.
 - b. Find a minimal subset S of X_1, X_2, \dots, X_{i-1} such that $P(X_i | X_1, X_2, \dots, X_{i-1}) = P(X_i | S)$.
 - c. Add edges from X_i to each element in S .
 - d. Define the CPT for variable X_i .

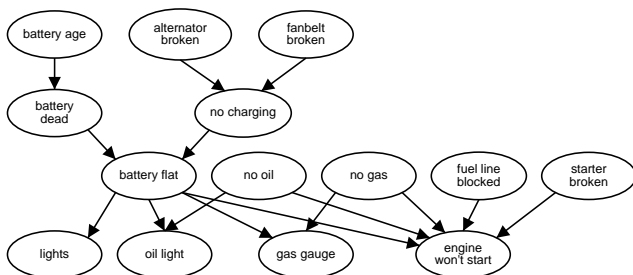
AI(0270)-9.20

Choosing an ordering

- There is something quite arbitrary: **how to choose an ordering?**
- In general, we want to choose an ordering in which **causes runs before consequences**, so that step 3b results in the smallest S .
- But the procedure **produce a correct network even if we choose a bad ordering**. The only thing that happens is that we have a larger network (i.e., more edges).
- **Large network is bad** because the size of CPT **increases exponentially** with the number of incoming edges and the **time cost** of asking for a posterior probability **increases likewise**.

AI(0270)-9.21

A typical network: car problem



Note that there are usually some nodes, like no charging and battery flat, which is solely to **reduce the network size**.

Otherwise, every consequence has to depend on battery dead, alternator broken and fanbelt broken.

AI(0270)-9.22

Computing conditional probabilities

How to use a Bayesian network to compute, e.g., $P(A | B)$?

- For each combinations of variables such that both A and B are true, multiply all the values of the CPT to get the full joint probability.
- Sum up all the products.
- Do the same thing for all combinations that B is true.
- Divide the two sums, and we get the conditional probability we want.

But... there is a **huge** number of combinations of variables! If we have 30 boolean variables, we have $2^{30} = 10^9$ of them...

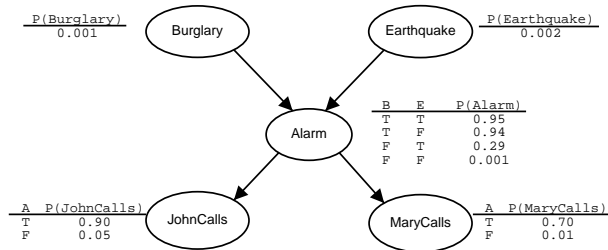
Bayesian network helps us to reduce the storage requirement. But can it also help us reduce the computational time?

The Bayesian network we've seen is a bit too simple to illustrate the ideas. We'll use a slightly more complex example.

AI(0270)-9.23

Example

- One author of the book has installed a burglary alarm, and have two neighbours to phone him if they hear the alarm.
- But he lives at a place where there are occasional earthquake, which can also trigger the alarm.



Note that the Alarm CPT has 4 entries to make up all combinations.

AI(0270)-9.24

AI(0270)-9.25

Normalization

Given both calls, what is the chance that burglary happened?

- So we want the probability distribution $\mathbf{P}(\text{Burglary} \mid \text{JohnCall}=\text{true} \wedge \text{MaryCall}=\text{true})$. (We will abbreviate it as $\mathbf{P}(B \mid j, m)$).
- To do this, we can find the probability distributions $\mathbf{P}(B \wedge j \wedge m)$, and also the probability of $P(j \wedge m)$, so that we can find $\mathbf{P}(B \wedge j \wedge m) / P(j \wedge m)$.
- But in fact, $P(j \wedge m)$ is not needed, because it is just the sum of all $\mathbf{P}(B \wedge j \wedge m)$.
- Alternatively, we can say that we will just use $\mathbf{P}(B \wedge j \wedge m)$ as the ratios between the conditional probabilities. We will **normalize** it to 1 later.
- E.g., if we find that $P(b \wedge j \wedge m)$ and $P(\neg b \wedge j \wedge m)$ are 0.2 and 0.1 respectively, then we will normalize it and conclude that $P(b \mid j \wedge m)$ and $P(\neg b \mid j \wedge m)$ are 0.333 and 0.667 respectively.

How to calculate?

- To know the fast way to do it, let's first try to do it the slow way, and see what we can improve.
- Let's find $P(b \wedge j \wedge m)$ ($P(\neg b \wedge j \wedge m)$ is similar).
- Apart from b, j and m , there are two more variables, E and A . We have to sum up all the possibilities of E and A .
- $P(b, e, a, j, m) = P(b)P(e)P(a \mid b, e)P(j \mid a)P(m \mid a)$
 $P(b, e, \neg a, j, m) = P(b)P(e)P(\neg a \mid b, e)P(j \mid \neg a)P(m \mid \neg a)$
 $P(b, \neg e, a, j, m) = P(b)P(\neg e)P(a \mid b, \neg e)P(j \mid a)P(m \mid a)$
 $P(b, \neg e, \neg a, j, m) = P(b)P(\neg e)P(\neg a \mid b, \neg e)P(j \mid \neg a)P(m \mid \neg a)$
 We use commas instead of \wedge to save space. All values on the right are entries in one of the CPTs of the bayesian network.
- They have values 0.00000119700, 0.000000000005, 0.00059101560 and .0000002994 respectively. Summing them up we get $P(b, j, m) = 0.00059224259$. (Similarly, $P(\neg b, j, m) = 0.001491857649$.)

AI(0270)-9.26

How to speed it up?

Now we know two facts...

- $P(b \mid j, m) = 0.284$ (so small!!). This number comes from normalizing the two values.
- A lot of lookups and computations are **repeated** if we are not careful. Why we need to lookup twice for each of $P(j \mid a), P(m \mid a), P(\neg j \mid a), P(\neg m \mid a), P(b)$ and $P(e)$? Even worse, if more complex networks they might not be a simple lookup (but instead a long computation)!

We can group terms out to achieve some of the savings. E.g.,

$$\begin{aligned}
 P(b) (P(e) & (P(a \mid b, e)P(j \mid a)P(m \mid a) \\
 & + P(\neg a \mid b, e)P(j \mid \neg a)P(m \mid \neg a)) \\
 + P(\neg e) & (P(a \mid b, \neg e)P(j \mid a)P(m \mid a) \\
 & + P(\neg a \mid b, \neg e)P(j \mid \neg a)P(m \mid \neg a)))
 \end{aligned}$$

Still, $P(j \mid a), P(m \mid a), P(j \mid \neg a)$ and $P(m \mid \neg a)$ are repeated. Some of you might yell out, "dynamic programming!"

AI(0270)-9.27

Let's formalize it...

- What we want to compute:

$$\sum_{E,A} P(b)P(E)P(A \mid b, E)P(j \mid A)P(m \mid A).$$
- Moving unrelated variables outside sums, we get

$$P(b) \sum_E P(E) \sum_A P(A \mid b, E)P(j \mid A)P(m \mid A).$$
- Now we will compute things from the **right**:
 - Find $P(m \mid A)$ for all possibilities of A . Call it $\mathbf{f}_M(A)$. It is a vector. E.g., here it is ($a : 0.7, \neg a : 0.01$).
 - Find $P(j \mid A)$ for all possibilities of A . Call it $\mathbf{f}_J(A)$.
 - Find $P(A \mid b, E)$ for all possibilities of A and E . Call it $\mathbf{f}_A(A, E)$.
 Suffix is part of sum that we have considered, arguments are those variables that are not fix ed.

AI(0270)-9.28

Let's formalize it...

- We multiply $f_M(A), f_J(A)$ and $f_A(E)$ for each combination of A and E . Call the result $f_{AJM}(A, E)$.
- We call this **pointwise multiplication**: an independent product is calculated for every combination of values of the unknown variables.
- Find $\sum_A P(A \mid b, E)P(j \mid A)P(m \mid A)$ from $f_{AJM}(E)$; by summing all values for A . Call the result $f_{AJM}(E)$.
- We call this **summing out** the \mathbf{f} values involving A .
- We continue by finding $P(E)$, denoted as $f_E(E)$.
- Then we do pointwise multiplication between $f_E(E)$ and $f_{AJM}(E)$, resulting in $f_{E AJM}(E)$.
- We then sum out E to get $f_{E AJM}()$, and multiply by $P(b)$.

AI(0270)-9.29

The variable elimination algorithm

The algorithm is called “variable elimination”:

- At the beginning, setup an empty set S of known f values.
- Use the reverse ordering of the Bayesian network variables to process the variables.
- For each variable V :
 - Find its parents $Parents(V)$ that are not known, and extract the corresponding values in the CPT. Add the resulting list to S as a known f value.
 - If V is not known, sum-out V , pointwise multiply as needed.
- Do pointwise multiplication on the result.

Actually, all values of the queried variables can be done at the same time, which will result in more savings.

AI(0270)-9.30

How fast is it?

- The critical part of the algorithm is for pointwise multiplication.
- Each pointwise multiplication has the potential to just adds up the number of arguments in the f value.
 - If this number of arguments is n , then the value contains 2^n values.
- In each sum-out, the number of arguments is reduced by 1.
- The efficiency of the algorithm depends on whether the acquisition or removal of variable is faster in the f values.
- If the Bayesian network is a **tree**, the speed is about the same. This results in a **linear** time algorithm.
- In a general graph, the speed of acquisition is much faster at the beginning, resulting in an **exponential** time algorithm.
- But the problem is NP-hard, so sometimes we need exponential time.

AI(0270)-9.31

Approximation algorithms

- But probabilistic reasoning is special: if you don't require exact answer, it can be done much more efficiently .
- An especially simple way: just do **many experiments!**
- E.g., for the burglary alarm domain, we can start choosing a value for B , using the CPT of B .
 - At probability 0.999 to choose $\neg b$, 0.001 to choose b .
- Suppose we chose $\neg b$. Then we choose a value for E , say getting e .
- Then we choose a value for A , **using the current choice of B and E** . So we choose a at probability 0.29.
- If at any point the chosen value is inconsistent with the evidence, we drop it. At the end we count the number of experiments with b against the number of experiments with $\neg b$.

AI(0270)-9.32

Problem of experiments

- There is a critical problem of this methodology—we simply drop too many experiments!
- E.g., since the actual probability of $j \wedge m$ is 0.0021, we can expect to drop 998 experiments in 1000 of them. The algorithm is not really that fast.
- There are many ways to deal with the problem. E.g., we can say that we **never drop** the experiment. Instead, if we go through a variable that is known (e.g., j), we always choose j , but give the experiment a weight that depend on its probability.
- E.g., in the example before, if we choose a at the third step, then we will give the experiment a weight of $0.9 \times 0.7 = 0.63$. If instead it is $\neg a$, it get a weight of 0.005. (At the end we give the ratio of the sum of weights.)
- This introduce other complication, so there are better algorithms. Bottom line: we can do it more efficiently if approximation is allowed.

AI(0270)-9.33