

Problem, answer, ...

**Lecture 18
Belief Network**

Now we have the basic probability framework, we will discuss a systematic way to express probability knowledge.

Belief network allows us to express the probabilities we know about a domain, and all the conditional independence relations.

Reference:

- Textbook Chapter 15, section 15.1–15.2

AI(0270)

AI(0270)-18.1

- Problem: we can **get all conditional probability** we need if we have the full joint of **all** random variables, but a **full joint is too much to ask for**.

- Answer: **directly work with conditional probabilities!**

So we use conditional probabilities and Bayes' rule, but then...

- Problem: we **encounter conditional probabilities that are difficult to estimate**. Also, there are a **lot** of such probabilities to estimate.

In our last example, we have just one. But if we do more complicated problems, there are much more.

- Answer: many times we can make reasonable **conditional independence assumption** to **avoid specifying** these probabilities individually.

But as we put it into real use...

- Problem: there are **too many independence assumption** to specify!

Complications of "conditional independence"

- Conditional independence is **more complicated** than what it initially sounds.
- E.g., now we assume that **given Cavity, Toothache and Catch are independent**.
But this assumption is actually just an approximation. E.g., perhaps if the cavity is more serious, probability of both toothache and catch goes up. Then they are not really independent.
- What if we don't know Cavity? If we have Toothache, then **probability of Cavity increases**, and **as a result** probability of Catch also goes up.
- So if we don't know whether Cavity is true or false, then Toothache and Catch depends on each other.
- We want a **system** that allow us to **express** all the conditional independence assumptions, and allow us to calculate the posterior probabilities.

AI(0270)-18.2

The belief network

- The basic trick we have used is that **Cavity is the direct cause of Catch and Toothache**.
- Roughly speaking, the **belief network captures such direct-cause relations**.
- The network contains **nodes**, each representing a random variable.
We treat propositions as random variables which can be either True or False.
- The network contains **edges**, which represent the "direct causes" that influence the probability of the variable.
With no cycles.
- E.g., our network would have an arrow from Cavity to Toothache and to Catch, meaning that both Toothache and Catch is directly influenced by the knowledge of Cavity.
- But what is meant by **directly influenced by** another variable? We will come back to this in a moment.

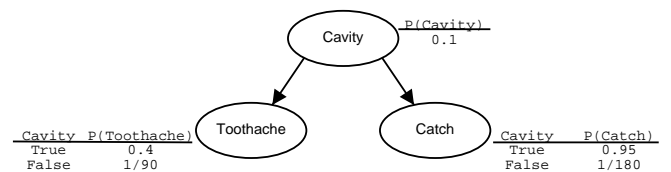
AI(0270)-18.3

The Conditional Probability Table (CPT)

- The final important information to calculate posterior probabilities is the **values of prior and causal probabilities**.
- One way to express them is a **conditional probability table (CPT)**.
- For a variable which is caused by n other variables (i.e., have n edges pointing to it), it has entries which **enumerates the possibilities of all its value** under each **combination of these n variables**.
- For a variable which has no edge pointing to it, it has a table of just **one entry** for each of its value, giving the prior probability.
- E.g., in the belief network of dental problem, the Cavity node will have a table containing just two entries, $P(Cavity)$ and $P(\neg Cavity)$.
- The Toothache node have a table containing four entries: $P(Toothache | Cavity)$, $P(Toothache | \neg Cavity)$, $P(\neg Toothache | Cavity)$ and $P(\neg Toothache | \neg Cavity)$.

AI(0270)-18.4

Examples

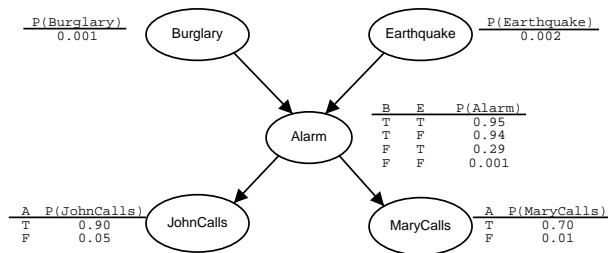


- Note that we simplify this a bit by **not writing out the probability $P(\neg Cavity | \dots)$** , because we can always subtract $P(Cavity | \dots)$ from 1 to get the probability.
- Also, this is a simple case: each variable is dependent on at most 1 boolean variable, so each table contain just one or two lines.
- We will use a more complicated example...

AI(0270)-18.5

Example 2

- 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)-18.6

AI(0270)-18.7

What it really mean? Two interpretations

- But... what is really **meant** by the network we have written?
- The “directly causing” interpretation can only give a intuitive idea about what is the network, because **our axioms of probability does not have the idea of a variable “directly influencing” another.**
- So we must **turn the intuitive idea into something defined**, i.e., prior probability, conditional probability and conditional independence.
- It turns out that **there are two interpretations:**
 - We can view the network as a **way to define the full joint** without using a lot of numbers.
 - We can view the network as a **way to express a set of conditional independence.**

The simplified-big-joint view

- We can view the network as providing a way for us to specify the **full joint**, i.e., for the prior probabilities when **values of all variables are specified.**
- How it is specified? Basically, each CPT will give a probability. The value of the joint is simply their **product**.
- E.g., $P(B \wedge \neg E \wedge A \wedge J \wedge \neg M)$ is...
 - $P(B) = 0.001$
 - $P(\neg E) = 0.998$
 - $P(A | B, \neg E) = 0.94$
 - $P(J | A) = 0.90$
 - $P(\neg M | A) = 0.01$
- $P(B \wedge \neg E \wedge A \wedge J \wedge \neg M) =$ all multiplied together = 0.00000844308.

AI(0270)-18.8

AI(0270)-18.9

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.**
i.e., the causes are before the consequences.
- E.g., in our network, B, E, A, J, M.
- By repeatedly applying the “product rule” (i.e., definition of conditional probability), we know
 $P(B, E, A, J, M) = P(B)P(E|B)P(A|B, E)P(J|A, E, A)P(M|B, E, A, J)$
 We write comma instead of \wedge to save space.
- The joint interpretation tells us that
 $P(B, E, A, J, M) = P(B)P(E)P(A|B, E)P(J|A)P(M|A)$
 We write comma instead of \wedge to save space.
- The joint interpretation tells us that $P(E|B) = P(E)$, $P(J | B, E, A) = P(J | A)$, and $P(M | B, E, A, J) = P(M | A)$.

Conditional independence view

So now we have a **sequence of conditional probabilities...**

- E is (unconditionally) independent on B.
This is only for the prior probabilities, not if given more evidence! E.g., given A, B and E are no longer independent.
- Given A, J is independent on B and E.
- Given A, M is independent on B, E and J.

In general, the last equation tells the following property:

- Given **all** parents of a variable X, X is **independent** of all its **predecessors** in the list of variables.

We call this the **conditional independence interpretation** of the belief network. Of course, the two interpretations are equivalent.

AI(0270)-18.10

AI(0270)-18.11

Building a belief network

Now that we understand what belief network really means, we can think about how to **build** a belief 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 .

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**.
- This easily get us to multiply connected networks, which are difficult to handle.
- Let's see what will happen if we choose a bad ordering, e.g., M, J, A, B, E.

AI(0270)-18.12

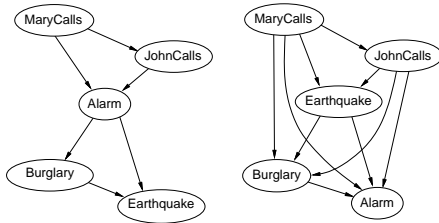
Network resulting from bad ordering

- We first add M. No edge to add.
- Then we add J. Does it depend on M? Well... yes, because if Mary calls, it is more likely that John calls as well.
- We now add A. Can we delete the dependence on either M or J? No... each of them increases the likelihood of the alarm ringing.
- We add B now. Can we delete the dependence on M and J? Yes... since if we already know whether the alarm is ringing, it is not important whether Mary or John actually calls.
- We add E now. Can we delete the dependence on M and J? Yes, with similar reason. Can we delete B? Not quite. If we know burglary, then the probability of earthquake goes **down** even if alarm is ringing!
The burglary explains why the alarm is ringing, so the probability of earthquake goes back to around its prior probability—explaining away.

AI(0270)-18.13

Our resulting network

So we get the network on the left...

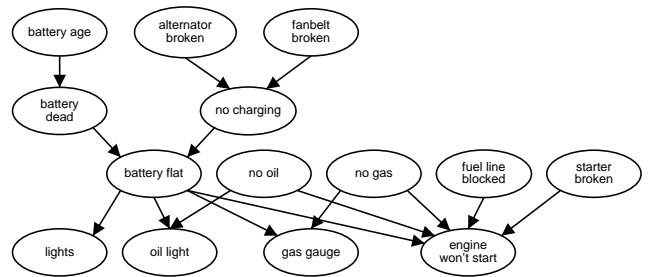


There are 6 edges rather than 4. If we choose a **really wrong ordering** (M, J, E, B, A), everything can depend on all others (right), and we get a really bad network that requires **as many probability values as the full joint**.

Still correct, but **express all conditional independence relations**.

AI(0270)-18.14

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, alternater broken and fanbelt broken.

AI(0270)-18.15

It's still much to specify...

- We note that at times, **nodes with large fan-in** (number of incoming edges) are **unavoidable**.
- E.g., "engine won't start" will depend on 5 variables.
- Then we need to specify and store 32 numbers for the CPT...
- but in most cases, we don't really need to.
- Most of the time **conditional probabilities fall into one of several patterns**.
- E.g., in this case, if any of the "causes" occurs, the engine has **no way to start!** So the CPT just has one entry being useful (engine won't start for miscellaneous reasons), all other entries are 0 and 1.
- If a random variable is a **logical combination** of other variables, then there is no randomness at all, and we can represent the CPT by the logical formula: **deterministic nodes**.

AI(0270)-18.16

Another Example: Noisy-OR

- Sometimes the logical combination does not exactly represent the relations among the variables because there are **some randomness**.
- E.g., we might want to say something similar to $Cold \vee Flu \vee Malaria \Rightarrow Fever$, but we want to allow some uncertainty.
- If the uncertainty interacts **irregularly**, then we have no choice but writing out all probabilities.
- But usually we can assume that they have independent inference. E.g., if $P(\neg Fever|Cold) = 0.6$ and $P(\neg Fever|Flu) = 0.2$, then we might want to assume $P(\neg Fever|bad\ Cold, Flu) = 0.6 \times 0.2 = 0.12$.
- We call this a **Noisy-OR** pattern: the probability of a consequence is increased "independently" when each cause is observed.
- This reduce the number of probability values to specify from 8 to 3.

AI(0270)-18.17

Conditional independence again...

- The conditional independence view of the network **define some conditional independence relationship** that is assumed by our network.
- But by these conditional independence, there are usually **other conditional independence that results** from these conditional independence.
- E.g., in our (tiny) dental problem example:
 - By construction we know $P(\text{Catch}|\text{Cavity}, \text{Toothache}) = P(\text{Catch}|\text{Cavity})$.
 - But if we apply Bayes' rule, we would also know that $P(\text{Toothache}|\text{Cavity}, \text{Catch}) = P(\text{Toothache}|\text{Cavity})$.
- Is there a way for us to **directly read out** such **independence relations** from our network?

AI(0270)-18.18

More ways for independence: d-separation

Let's consider the **singly connected case** first, i.e., there is **at most one undirected paths** between each pair of nodes.

- Consider a path between X and Y. If we know an intermediate node Z as evidence, with arrows pointing **to Z from one side and out of Z from the other**, then X and Y are independent.
- Consider a path between X and Y. **If we know an intermediate node Z as evidence**, with arrows pointing **from Z in both directions**, then X and Y are independent.
- Consider a path between X and Y. **If we know an intermediate node Z** with arrows pointing **to Z in both directions**, then X and Y are independent **if we don't know Z and all its descendants**.

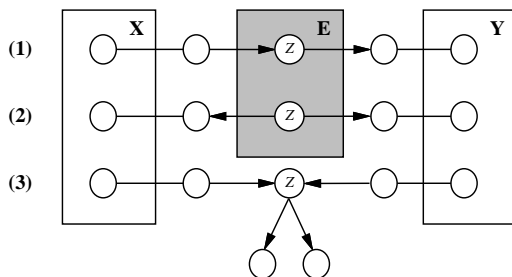
We say set the evidence **d-separate** the two nodes X and Y.

Note the wording! Here, the "set of evidence" can be a larger set that contains (or, in the last case, not contain) Z.

AI(0270)-18.19

Graphically

This summarize the last 3 complicated sentences...

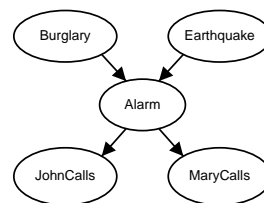


X and Y are the two nodes to be separated, E is a node in the set of evidences. In the last case, we want Z and all descendants **not** in the set of evidence.

AI(0270)-18.20

Example

Let's look at our example again.



Rule 1: given Alarm, Burglary is independent of MaryCalls.

Rule 2: given Alarm, JohnCalls is independent of MaryCalls.

Rule 3: given none of Alarm, JohnCalls and MaryCalls, Burglary and Earthquake are independent.

In more complicated networks, there are more complicated independences.

Exercise: try to do it for the car problem network!

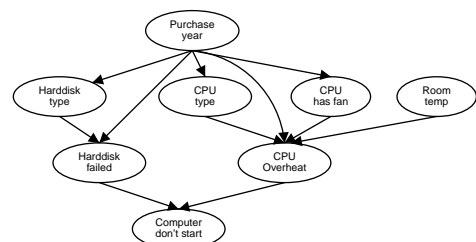
AI(0270)-18.21

What for the multiply-connected cases?

- The idea readily extends to multiply connected networks...
- except that we require **all undirected paths** to be d-separated by the evidences.
- So two nodes X and Y are independent given a set of evidence, if:
 - for all undirected paths $[X, V_1, V_2, \dots, V_n, Y]$ between X and Y,
 - there exists some node V_i , where $1 \leq i \leq n$
 - Such that X, V_i , Y is in one of the configurations shown in the last page, given the evidence nodes.

AI(0270)-18.22

Example: computer problem



Knowing CPU Overheat, is Computer Won't Start independent of CPU has fan? No.

Path [Computer don't start - harddisk failed - purchase year - CPU has fan] not d-separated by evidence CPU Overheat.

What if we also know purchase year?

AI(0270)-18.23