Bayesian Networks

CSCI 2400 "Part 6"
Instructor: David Byrd

Disclaimer: Lecture notes can't and won't cover everything I say in class. You should attend class each day and use these for review or reinforcement.

9 Bayesian Networks

We have seen that with the full joint distribution (FJD) of some random variables (RV), we can answer any probability-related question within its domain. Full joint probabilities appear directly in the FJD; we can marginalize or sum out RVs to obtain single or (non-full) joint distributions; and we can compute conditional probabilities through normalization over a relevant subset of the RV universe.

We have also already observed a problem. The number of entries in the FJD grows exponentially with the number of RVs. Even if all RVs are boolean, then $E(N) = 2^N$, where E is the count of FJD entries and N is the number of RVs.

9.1 Generalized Bayes Model

One way to address the above problem is to prune the number of RVs needed to compute the FJD using independence and conditional independence. We first saw this in the probabilistic version of Wumpus World, in which an FJD with 4,096 entries was reduced to only eight cases that we needed to calculate.

Recall Bayes' Rule (or Law):

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

This is critical to virtually all probability-based AI, because what we *have* is unreliable sensors that produce readings based on the state of the world (i.e. $P(E|X)$) and what we *need* is to know the likely state of the world given our sensor readings (i.e. $P(X|E)$).

If we generalize Bayes' Rule on the assumption that there will be one state of the world, but we have many pieces of evidence, we obtain:

$$P(c|e_1, e_2, \ldots, e_n) = \frac{P(e_1, e_2, \ldots, e_n|c)P(c)}{P(e_1, e_2, \ldots, e_n)}$$

Here we label our state of the world (hidden cause) c and our various sensor readings (evidence) $e_i \forall i \in 1, 2, \ldots, n$. Observing that the denominator is simply some *normalization*

constant α that does not depend on our hidden cause c, we can rewrite this as:

$$P(c|e_1, e_2, \ldots, e_n) = \alpha P(e_1, e_2, \ldots, e_n|c)P(c)$$

To make the problem more tractable, we usually make the *naive Bayes assumption* that all evidence is conditionally independent given the state of the world. This is quite reasonable: if we (somehow) *know* there is a Wumpus in a given room, a negative reading from one of our Wumpus detectors makes us no less likely to expect a positive reading from a second detector. (Since we *know* there is a Wumpus, the first reading must have been a false negative error due to the sensor's own limitations. The second sensor might also read false, because it also has a non-zero false negative error rate, but this has nothing to do with the first sensor.) This only works for evidence! If we have multiple *causes*, we'll just have to condition on all of them jointly. Boo.

Using the above assumption, and recalling that the joint probability of *independent* RVs is their product, we can vastly simplify our formula:

$$P(c|e_1, e_2, \ldots, e_n) = \alpha P(c) \prod_{i=1}^{n} P(e_i|c)$$

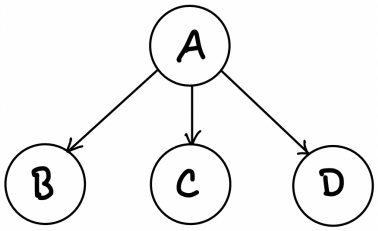
This is much better! Now when we have many pieces of evidence, we can determine their *individual* likelihoods, multiply them all together, multiply by the unconditional probability of the cause, normalize to a probability (sum of 1), and we're done! Where the (boolean) FJD grows at a rate of 2^n, the general Bayes model grows only at a rate of n: from exponential to (not just polynomial but) linear growth! This reduction of effort and data required is so significant that we always start with the naive Bayes assumption and *only* transition to a more complex model if we discover that it won't work.

9.2 Structure of a Bayes Net

Bayes nets are also called influence maps, causal networks, or belief networks. They are meant to be a *compact but equivalent* representation of a FJD in graphical form, taking advantage of conditional independence.

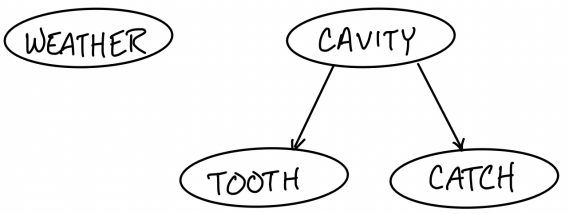
The nodes in a Bayes net represent known or unknown RVs. The edges represent conditional probability tables (CPTs). The direction of an arrow represents the direction of the influence or causal effect. Cycles are not permitted. (In other words, you must not be able to return to the same node by following a sequence of arrows. In *other* other words, a Bayes net is a directed acyclic graph (DAG).)

Here is a simple example of a Bayes net:



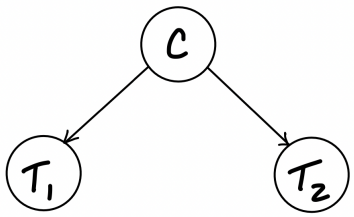
Even without any numeric probabilities, the structure of the Bayes net gives us useful information about the RVs contained therein. A is a common cause of B, C, and D. B, C, and D are conditionally independent given A. We can think of the base side of each arrow as being attached to a cause, and the pointy side as being attached to an effect.

RVs directly connected by an arrow have a direct probability relationship (i.e. a CPT). RVs indirectly connected by arrows have an indirect probability relationship (i.e. conditional independence). RVs not connected by arrows (even indirectly) are independent. For example:



You will recall from the probability section of the class that having a cavity made both a toothache and a dental-instrument "catch" more likely, that toothache and catch were independent when conditioned on cavity, and that all three were independent of the weather. The Bayes net shows this information at a glance, without needing any of the actual probabilities.

Always remember, we *assume* that, given a single cause, the effects are independent (naive Bayes assumption). Thus in a simple Bayes net representing the possibility of (actually) having cancer and the result of two different cancer detection tests:



The structure tells us that $T_1 \perp T_2|C$. That is, if you (somehow) already know you have cancer, then taking T_1 first and seeing a positive (or negative) result does *not* make you any more (or less) likely to later see a positive (or negative) result on T_2. Given that you have cancer, the likelihood of a positive result on T_1 was just its *true positive rate*. Regardless of the outcome, the likelihood of a positive result on T_2 is just *its* true positive rate. In other words, $P(T_2|T_1, C) = P(T_2|C)$.

Note that the structure does *not* tell us that $T_1 \perp T_2$! If the value of C is unknown, a positive result on T_1 absolutely *does* make it more likely you will see a positive result on T_2 later. The relationship is indirect, but still real. (The positive test has increased the likelihood that you have cancer, which *in turn* increases the likelihood of seeing a positive result on the second test.) In other words, $P(T_2|T_1) \neq P(T_2)$.

Because nodes indirectly connected with arrows are conditionally independent (given any "chokepoint" node in between), we can further observe that given all its parents and children, a node is then independent of all other nodes in the Bayes net. This is called the

Markov Blanket. Similarly, given its parents, a node is then independent of all its parents' predecessors. That is, whatever effect a grandparent node might have is already baked into the parent, and the child need not worry about it, *if conditioned on the parent*.

9.3 Bayes Net Example

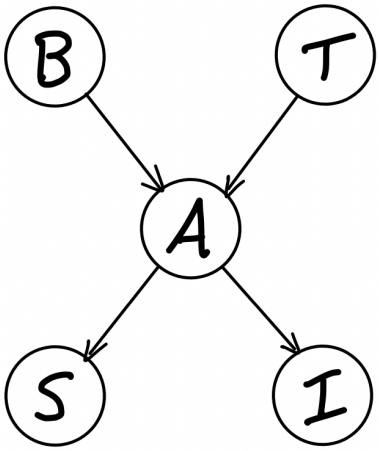
Imagine that you are traveling far from home, for business or pleasure, and are concerned about the safety of your house and the belongings stored there. You do have a home invasion alarm, but it came with the (older) house and doesn't have any smart app capabilities. It just makes *a whole lot of noise*. Luckily you have two friendly neighbors who are happy to call and let you know (or possibly complain) when your burglar alarm goes off. Let's call them Steve and Imani.

Steve is a pretty nosy neighbor who rarely leaves home, so he is almost certain to call you if the alarm goes off (i.e. low *false negative rate*). Unfortunately, Steve's hearing acuity is not great, and he gets a bit confused, so he also tends to mistake his own phone ringing for your burglar alarm (i.e. high *false positive rate*).

Imani loves listening to music and is also out clubbing many nights. She almost never mistakenly calls you (i.e. low false positive rate), but often doesn't notice the alarm going off (i.e. high false negative rate).

Burglaries are pretty rare in your neighborhood: on average someone might try to break into your house every few years. You also live in an area where powerful thunderstorms are rare, but do occur. Sometimes those storms rattle your windows hard enough to set off the alarm.

We can represent this complex example rather nicely as a Bayes net with five nodes (RVs), four edges, three CPTs, and two unconditional probabilities:



We need an unconditional probability for each RV with no "parent" in the graph: $P(B) = 0.001$ and $P(T) = 0.002$. This tells us how common burglaries and thunderstorms are.

We need a CPT for each RV that does have one or more parents. The structure of the Bayes Net tells us the exact *shape* of our CPTs. Additional parents increase the size of a CPT because we need the joint probability of the parent nodes. Here they are:

B	T	$P(a\|B,T)$
T	T	0.950
T	F	0.940
F	T	0.290
F	F	0.001

A	$P(s\|A)$
T	0.90
F	0.05

A	$P(i\|A)$
T	0.70
F	0.01

Note that, as in the past, we are using capital letters to indicate RVs and lowercase letters to indicate particular events/outcomes. For example, the RV for thunderstorm is T and has two possible outcomes, t or $\neg t$ (either there is, or is not, a thunderstorm).

These ten probability entries (eight conditional, two unconditional) allow us to calculate any combination of the $2^5 = 32$ entries we would find in the FJD. Assuming we have to measure or sample these probabilities, that will also be much easier! "Of all the times your alarm went off, how often did Imani call you?" versus "How frequently does it happen that there is a burglary during a thunderstorm, and your alarm doesn't go off, but Steve and Imani both call you anyway?" If we didn't know any probabilities to start with, the first question would be much easier to estimate than the second.

So we can compute any arbitrary entry in $P(B, T, A, I, S)$ without needing that FJD. Why? We are computing a joint probability, and each node (RV) is conditionally independent of all the others *given its parents*. Thus, so long as we *condition on the parents*, we can simply multiply together the separate likelihood of each outcome:

$$P(X_1, X_2, \ldots, X_n) = \prod_{i=1}^{n} P\big(X_i|\text{Parents}(X_i)\big)$$

9.3.1 Example Question 1

What is the probability that your burglar alarm does off despite there being no burglary or thunderstorm, but Imani and Steve both call anyway?

Mathematically speaking, this question is asking for $P(a, \neg b, \neg t, i, s)$. We can use the previous equation directly:

$$\begin{aligned} P(a, \neg b, \neg t, i, s) &= P(a|\neg b, \neg t)P(\neg b)P(\neg t)P(i|a)P(s|a) \\ &= 0.001(0.999)(0.998)(0.7)(0.9) \\ &= 0.00063 \end{aligned}$$

This amounts to a 0.06% chance: quite unlikely! We obtained the result, as earlier suggested, by looking at the appropriate entries in each CPT and multiplying them together. We were able to do this because each RV is independent of the others *when conditioned on its parents*.

9.3.2 Example Question 2

Given that Steve and Imani both just called you, what is the probability that there is a burglary happening?

This question is a little trickier. Our question does not reference the alarm or a possible thunderstorm, so we will have to *marginalize* those RVs out. The question also asks about a

conditional probability, which means we will have a normalization constant to worry about (because we are considering a limited subset of the universe, and thus our possible outcome likelihoods will not sum to one).

$$
\begin{aligned}
P(b|i,s) = \quad & \frac{P(b,i,s)}{P(i,s)} \\
= \alpha \quad & P(b,i,s) \\
= \alpha \quad & \sum_{x=\{a,\neg a\}} \sum_{y=\{t,\neg t\}} P(b,i,s,x,y) \\
= \alpha \quad & \sum_{x=\{a,\neg a\}} \sum_{y=\{t,\neg t\}} \big[P(x|b,y)P(b)P(y)P(i|x)P(s|x)\big] \\
= \alpha \big[\quad & P(a|b,t)P(b)P(t)P(i|a)P(s|a) + \\
& P(\neg a|b,t)P(b)P(t)P(i|\neg a)P(s|\neg a) + \\
& P(a|b,\neg t)P(b)P(\neg t)P(i|a)P(s|a) + \\
& P(\neg a|b,\neg t)P(b)P(\neg t)P(i|\neg a)P(s|\neg a)\big] \\
= \alpha \big[\quad & 0.95(0.001)(0.002)(0.7)(0.9) + \\
& 0.05(0.001)(0.002)(0.01)(0.05) + \\
& 0.94(0.001)(0.998)(0.7)(0.9) + \\
& 0.06(0.001)(0.998)(0.01)(0.05)\big] \\
= \alpha(\quad & 0.0005922)
\end{aligned}
$$

In the above example, we first carefully read the question and understand what is being requested: *given* that Steve and Imani have called, what is the likelihood of a burglary? In other words: what is $P(b|i,s)$? We then recall the definition of conditional probability: $P(x|y) = \frac{P(x,y)}{P(y)}$, which can be read as: "Out of all the times y happened, how often did x *also* happen?" Then, since the denominator sums over all possible values of b (which just normalizes the probabilities to sum to one), we can replace it with a normalization constant α as a reminder.

Now we must marginalize out the RVs we do not care about by summing over all possible value combinations of a and t. We are now summing over four possible combinations of a full joint probability, which we saw how to do in the previous example, and which proceeds in the same way.

Importantly, when we reach the answer of $\alpha(0.0005922)$, the normalization constant should remind us that *we are not finished*. This number will be *proportionally correct* relative

to $P(\neg b, i, s)$, but it is not a probability, and so we cannot really evaluate its meaning (is it a lot or a little?) without also calculating $P(\neg b, i, s)$, which we can do in exactly the same way to obtain $P(\neg b, i, s) = \alpha(0.0014917)$

At a glance, we can see that the value of $P(\neg b, i, s)$ is about three times that of $P(b, i, s)$, so we might estimate our final answer will be about 25% (as the two probabilities must sum to one). We can compute the exact answer by dividing each of the two possibilities (for b and $\neg b$) by their sum:

$$\begin{aligned} P(b, i, s) &= 0.005922 \\ P(\neg b, i, s) &= 0.0014917 \\ P(i, s) &= 0.005922 + 0.0014917 \\ P(b|i, s) &= \frac{P(b, i, s)}{P(i, s)} = 0.2841787 \\ P(\neg b|i, s) &= \frac{P(\neg b, i, s)}{P(i, s)} = 0.7158213 \end{aligned}$$

The two conditional probabilities do indeed sum to one, and our answer (for the positive burglary case) is 28.4%.

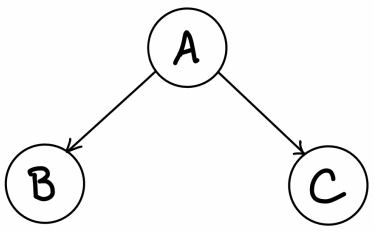
9.4 Why Bayes Nets in AI?

Given the rather mechanical nature of the math we've just performed, hopefully you can see that it would be easy to write a computer program to take a Bayes Net and perform these calculations, and that is exactly how this is applied to AI: convert a Bayes Net into an equation and solve it.

Having seen a couple of examples, there are obviously two necessary components of a Bayes Net: a graph to show the topology and a set of CPTs to fill in the numbers, with one CPT per node giving $P(node|parents)$, and noting that *parents* can be empty as with burglary and thunderstorm in our example. As we showed earlier, the Bayes Net *encodes* the FJD of all variables, even though it does not directly specify it:

$$P(X_1, X_2, \ldots, X_n) = \prod_{i=1}^{n} P\big(X_i|\text{Parents}(X_i)\big)$$

To help understand why Bayesian Networks are a good choice for AI, consider a simple, generic example:



Here are the related CPTs:

P(a)=0.3

A	$P(b\|A)$
T	0.8
F	0.5

A	$P(c\|A)$
T	0.6
F	0.2

Suppose that, under the usual Bayes Net assumption of $B \perp C|A$, our AI must compute: $P(a, b, \neg c)$:

$$\begin{aligned} P(a, b, \neg c) &= P(a)P(b|a)P(\neg c|a) \\ &= 0.3(0.8)(0.4) \\ &= 0.096 \end{aligned}$$

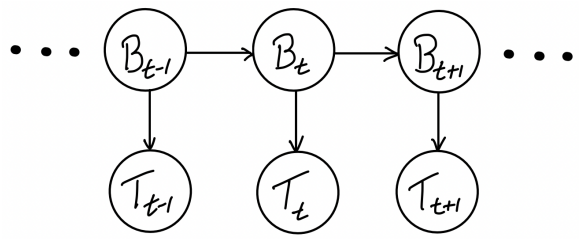
With straightforward reference to the CPTs, our intelligent agent can answer 9.6%, and then use this as part of whatever decision it is trying to make (expected utility, perhaps).

Here's the key: what is the RV A? The presence of a disease? A coin flip? Which opponent a video game agent has decided to attack? What about RVs B and C; what do they represent? Positive tests for a disease? Weighted/unfair coin flips? The chance a video game attack successfully strikes a target? *It makes no difference at all!* Bayesian Networks form the basis of many approaches to AI because they are simple to construct and use, and because they are *generic*: we can write one general subroutine that takes a Bayes Net and a (mathematical) question as parameters, and produces correct and useful answers, without any regard for what any of the RVs or probabilities actually mean.

10 Dynamic Bayesian Networks

One thing our current Bayes Nets *don't do* is to handle the passage of time. We can make multiple queries across time, but they will be approached independently, such that prior decisions and outcomes do not affect the current decision. This becomes a particular problem if all of our RVs do not exist at the same time, but there *are* RV dependencies (Bayes Net arrows, CPTs) that span across them.

For example, if we measure a person's blood sugar B_t at frequent periodic discrete time intervals $t \in \{1, 2, \ldots, N\}$, it is clear that the value of B_{t-1} will influence the value of B_t. Our blood sugar usually changes smoothly over time, so there will be a strong relationship between readings at nearby times. This produces a challenge for Bayes Nets, because we now need to establish a CPT (an "arrow" in the Bayes Net) between the *same* RV at different times:



Note that we separate the *hidden cause* of our real blood sugar values B_t from the *visible effect* of our blood sugar test result T_t (finger prick, etc). This is important, because it acknowledges that our sensors, observations, or evidence can always introduce their own errors. They *approximate* the underlying true value, but can never exactly, reliably reveal it. Without this, we could not model the accuracy of our sensors, which would result in less reliable agents.

We usually represent the hidden causes (true, unmeasurable state of the world) as a series of RVs X_t, and the observations or evidence obtained from our sensors as a series of RVs E_t. Even high-fidelity sensors have some frequency or refresh rate (10Hz, or ten times per second, for example) and *we do not know* what happens in between, so we consider t to be regular, discrete "time steps": $0, 1, 2, \ldots, n$.

10.1 Some Necessary Assumptions

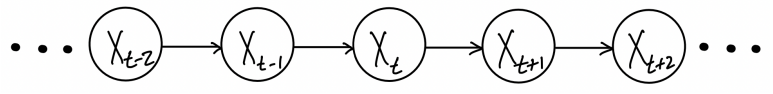
Let's break down a few issues with the approach suggested in the previous section and see how best to proceed.

10.1.1 Markov Assumption

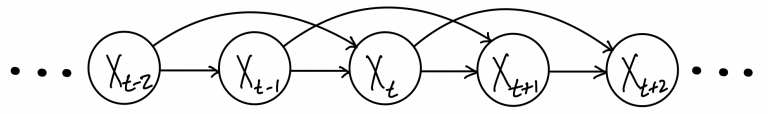
At this point, we have a tremendous complexity problem. So far as we know, X_t depends on *all* $X_0, X_1, \ldots, X_{t-1}$. This would require us to maintain, calculate, and deal with the full time series *forever*, e.g. $P(X_{10000}|X_0, X_1, \ldots, X_{9999})$. This is *unbounded* (and unreasonable). So, like the Naive Bayes Assumption from earlier, we will make a simplifying assumption that drastically reduces the complexity of our problem:

Markov assumption: X_t depends on some bounded subset of $X_{0:t-1}$.

We name them for the number of time steps in the bounded subset. The *first-order Markov assumption* states that the current state depends only on the single previous state: $P(X_t|X_{0:t-1}) = P(X_t|X_{t-1})$. Visualized as a Bayes Net:



The *second-order Markov assumption* states that the current state depends on the two previous states: $P(X_t|X_{0:t-1}) = P(X_t|X_{t-1}, X_{t-2})$. Again as a Bayes Net:



And so on. Problems that satisfy the Markov assumption, that the future is conditionally independent of the past given the present, are called *Markov processes* or *Markov chains*.

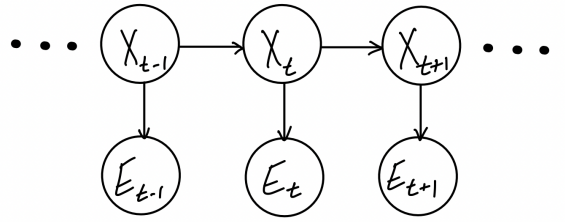
10.1.2 Sensor Markov Assumption

A second issue is that we can't measure any X_t *ever*. It is a hidden variable, unobservable by definition. We need to get our *evidence* (e.g. sensor readings) back in there somehow.

Now we have a complexity problem again. As far as we know for sure, our current sensor readings E_t also depend on the entire history of the world $X_{0:t}$, and might also depend on all prior sensor readings $E_{0:t-1}$.

Sensor Markov assumption: E_t depends only on X_t.

Thus we assert that $P(E_t|X_{0:t}, E_{0:t-1}) = P(E_t|X_t)$. Note that the naive Bayes assumption had gotten us halfway there already (conditioned on the state of the world, all evidence is independent of all other evidence). The Sensor Markov assumption is only adding independence of evidence from prior states of the world. So we can say that: "Our current evidence can be generated solely from the current state of the world." We can now consider our ever-evolving Bayes Net to be:



10.1.3 Stationary Process Assumption

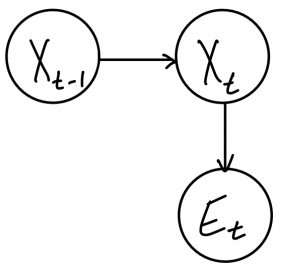
The final issue is that we still have a potentially infinite (or at least unbounded) set of RVs and CPTs required to solve our problem. The probability distribution $P(X_{1000}|X_{999})$ may not be the same as $P(X_{999}|X_{998})$, and so we would need a separate CPT for every time transition or time step sensor reading. We will assume that they *are* the same:

Stationary process assumption: $P(X_t|X_{t-1}) = P(X_{t-1}|X_{t-2}) = P(X_{t-2}|X_{t-3}) \ldots$.

Thus we assume that we are dealing with a *stationary process*. The state can change, and the evidence can change, but the *probability distributions* for those RVs do not change over time.

10.1.4 Bayesian Prior Assumption

Given this model, a first-order Markov assumption, a sensor Markov assumption, and a stationary process assumption, what part of the Bayes Net do I need to compute current state of the world X_t?



That's it! (Plus the associated CPTs for the "arrows", of course.)

One major problem remains. We can now compute any X_t given X_{t-1} and E_t. How can we obtain X_{t-1} ? We will need X_{t-2}, which would require X_{t-3}... *Uh oh!* We can run time forward forever if we so choose, but we have *no way to get started*.

We have obviously not been obtaining evidence (or sensor readings) *forever*, so let's consider the time at which our first sensor reading occurred to be time $t = 1$. At least for our current purposes, that will be the "beginning of time".

What was the state of the world before we obtained our first ever sensor reading? Who knows? (How would we know anything before our very first piece of evidence?) This seems like a fine place to introduce the concrete state of the world probability that we need to begin our calculations. We call it the:

> **Bayesian prior assumption:** Prior to any evidence, there must be some baseline $P(X_0)$ representing the initial state of the world.

What Bayesian prior might we choose? If we have any useful domain knowledge related to the problem (i.e. a heuristic), we might use that to choose appropriate values. Most often, we will use the:

> **Uniform prior assumption:** All possible values for X_0 are equally likely.

This is easy to defend. Since we have yet to obtain any evidence, we effectively know nothing about the state of the world, so we assume all possibilities are equally likely (e.g. 50/50 or 33/33/33, etc).

By contrast to the *prior probability distribution*, we call the unknown hidden RVs we are calculating after time $t = 0$ our *posterior probability estimates*. (Prior meaning before; posterior meaning after.)

One more bit of terminology: as with Bayes Nets sometimes being called Belief Networks, you will often hear our probability distributions X_t called *beliefs*. We never know the true state of the world for sure, but we always maintain some (probability-based) belief about the state of the world, and we use this to inform our decisions.

Finally, our "infinite" (or at least unbounded) probability time series can now be represented by three nodes, two CPTs, and one prior assumption. We call the CPT $P(X_t|X_{t-1})$

our *transition model*, the CPT $P(E_t|X_t)$ our *sensor model*, and the whole beautiful thing (which is just a Bayes Net for a temporal distribution) is now considered a *Dynamic Bayesian Network* (DBN).

10.2 Filtering

At this point, we might reasonably ask, "so what can we *do* with a Dynamic Bayes Net?" There are several common tasks:

- **Filtering:** compute our current beliefs about the world given all available evidence. That is, $P(X_t|e_{1:t})$.
- **Prediction:** estimate our beliefs about a *future* state of the world, given all *currently-available* evidence. That is, $P(X_{t+k}|e_{1:t})$.
- **Most likely explanation:** Find the sequence of past states most likely to result in the current state, given a sequence of past evidence. That is, maximize $P(X_{0:t-1}|X_t, e_{1:t})$, or "how did I get here?"

Most likely explanation tasks arise frequently in Hidden Markov Models (HMMs), in which the world stochastically transitions through multiple unknown states over time, and only some states produce observable evidence, and even then only sporadically. Thus after receiving some number of observations, we would like to reconstruct the most likely "path" through states over time, including those which did not produce any visible "output".

In this class, we focus primarily on the first application (filtering), and note that the second (prediction) is virtually identical, save that we are missing a few "recent" sensor readings as of the time we wish to predict.

10.3 Exact Filtering

Exact filtering, also known as state estimation, is the process of starting at time zero with a prior assumption and repeatedly updating our belief state based on newly-received evidence, until we are able to compute our beliefs at some current time t.

Assume that we have already run time forward from 0 to t. Given X_0, we have estimated X_i at each time step $i \in 1, 2, \ldots, t$ using the evidence $e_{1:i}$ that was available at that time. Now we have a current estimate X_t and have just received new evidence e_{t+1}. How do we estimate X_{t+1} ?

We covered the complete derivation of filtering in class (and you do not need to be able to reproduce it, just understand it), and ended up with:

$$P(X_{t+1}|e_{1:t+1}) = \alpha P(e_{t+1}|X_{t+1}) \sum_{x_t} P(X_{t+1}|x_t) P(x_t|e_{1:t})$$

This dense equation contains everything we care about. X_{t+1} is the new "posterior" belief (about the state of the world) we are trying to compute based on $e_{1:t+1}$, all of the evidence we have ever seen. $P(e_{t+1}|X_{t+1})$ is our sensor model, and incorporates our sensor Markov assumption. x_t iterates over all possible states of the world at the previous time step.

$P(X_{t+1}|x_t)$ is our transition model, and incorporates our first-order Markov assumption. $P(x_t|e_{1:t})$ is our most recent "prior" belief about the state of the world.

Thus with the filtering equation, and given our current belief, our sensor model, and our transition model, we can incorporate new evidence and update our beliefs accordingly. We can do this as many times as is necessary, because we have a recursive relationship, such that each new posterior belief we compute becomes our prior belief for the *next* time step.

10.3.1 Umbrella Example

To cement our understanding of exact filtering, let's consider a simple but detailed example.

Assume that we live in, and guard, a top-secret underground bunker. Each day, we observe our boss (who does not live here) enter the facility using the secret mountain elevator. Sometimes, the boss is carrying an umbrella; sometimes not. We wonder if it is raining today...

In this problem, the set of hidden state of the world RVs X that we care about contains only R, a boolean that represents whether it is currently raining. The set of relevant evidence RVs E that we can observe contains only U, a boolean that represents whether the boss is currently carrying an umbrella. Thus we must model R_t and U_t, and we will assume t is in discrete days (a reasonable time period to care about changes in the weather). Our problem should fit the DBN approach well: whether it rains today is influenced by whether it rained yesterday (stormfronts exist), whether the boss carries an umbrella today depends on whether it is raining *today*, and there's no specific reason to think these probability relationships would change over time (stationary process).

Our three-node graphical model for DBNs never changes, but we do need to specify the relevant CPTs for our exact problem:

R_{t-1}	$P(R_t = T\|R_{t-1})$
T	0.7
F	0.3

R_t	$P(U_t = T\|R_t)$
T	0.9
F	0.2

Here, we assume that we have done some research into weather patterns in our region and determined that weather (in terms of rain) is about 70% consistent from day to day. This is our *transition model*. Based on our knowledge of our boss, we have estimated that they will bring an umbrella on 90% of rainy days (occasionally surprised by rain), and on 20% of non-rainy days (occasionally looks like it will rain, but doesn't). This is our *sensor model*.

To use Bayesian inference, we must always have an initial prior assumption. Here, that will be R_0, the probability distribution of rain on the day *before* we make our first umbrella observation U_1. If we know that we live in a particularly wet or dry area, we might choose an unequal prior assumption (probably raining or probably not raining), but for this example we will use the common *uniform prior*, and assume it is equally likely to be raining or not. That is: $P(X_0) =< 0.5, 0.5 >$, using the notation of a probability vector (essentially a 1-D

tuple) that reports all possible outcomes. (Here it will be True and False, always in that order.)

The question: The boss shows up with an umbrella on the first two days we are paying attention. What is the probability it is raining today (the second day)?

First we consider the question and carefully translate it to an equivalent mathematical expression: $P(R_2|U_1 = T, U_2 = T)$. Now we must start at day 0 with our initial prior assumption and work forward using the exact filtering equation. Before seeing the first umbrella, we have no idea if it is raining, so we start with a uniform prior assumption:

$$P(R_0) =< 0.5, 0.5 >$$

Now time passes from day 0 to day 1. Working from *right to left* in the filtering equation, we start with our prior belief and apply the transition model to obtain a preliminary belief for day 1:

$$\begin{aligned} P(R_1) &= P(R_1|R_0)P(R_0) \\ &= \sum_{r_0 \in R_0} P(R_1|r_0)P(r_0) \\ &=< 0.7, 0.3 > (0.5) + < 0.3, 0.7 > (0.5) \\ &=< 0.35, 0.15 > + < 0.15, 0.35 > \\ &=< 0.5, 0.5 > \end{aligned}$$

Well, that makes sense. It is day 1, but we have not yet incorporated any evidence, so we still have no idea if it is raining. Now we observe the umbrella, and continue working right to left in the filtering equation, adjusting our preliminary belief to better fit the new evidence:

$$\begin{aligned} P(R_1|U_1 = T) &= \alpha P(U_1 = T|R_1)P(R_1) \\ &= \alpha < 0.9, 0.2 >< 0.5, 0.5 > \\ &= \alpha < 0.45, 0.1 > \\ &=< \frac{0.45}{0.55}, \frac{0.1}{0.55} > \\ &=< 0.818, 0.182 > \end{aligned}$$

Seeing the umbrella once has changed our belief about the possibility of rain upwards from a prior estimate of 50% to a posterior estimate of almost 82%. This again makes sense, as the boss is very likely to bring an umbrella when it is raining, and not very likely to bring one when it isn't. Now time passes again, from day 1 to day 2, and we start back at the right side of our filtering equation using our most recent posterior as our new prior:

$$\begin{aligned} P(R_2|U_1 = T) &= P(R_2|R_1, U_1 = T)P(R_1|U_1 = T) \\ &= \sum_{r_1 \in R_1} P(R_2|r_1, U_1 = T)P(r_1|U_1 = T) \\ &=< 0.7, 0.3 > (0.818) + < 0.3, 0.7 > (0.182) \\ &=< 0.5726, 0.2454 > + < 0.0546, 0.1274 > \\ &=< 0.6272, 0.3728 > \end{aligned}$$

The transition model has caused our beliefs to drift back towards a 50/50 "unknown" split. This fits our intuition in two ways: first, our transition model says it is equally likely to stop raining if it was, or to start raining if it wasn't; second, if we kept running time forward without any new evidence (i.e. predicting the future), we should expect to eventually arrive back at "we have no idea". However, we then see the umbrella for the second time, and update our preliminary day 2 belief with this evidence and our sensor model:

$$\begin{aligned} P(R_2|U_1 = T, U_2 = T) &= \alpha P(U_2 = T|R_2, U_1 = T)P(R_2|U_1 = T) \\ &= \alpha < 0.9, 0.2 >< 0.6272, 0.3728 > \\ &= \alpha < 0.56448, 0.07456 > \\ &=< \frac{0.56448}{0.63904}, \frac{0.07456}{0.63904} > \\ &=< 0.883, 0.117 > \end{aligned}$$

So at the end of day 2, after seeing the umbrella twice, our belief about the state of the world is: "There is an 88.3% chance it is raining today." Given that storm fronts and rainy seasons exist, our estimation of the likelihood of rain increases each day that we see the umbrella.

10.3.2 Exact Filtering Drawback

The primary issue with exact filtering is that it can be quite computationally expensive. We must always start at time zero and compute a belief update for every day, one at a time, up to whatever day we are interested in. Additionally, every day, we must sum over every possible state of the world. Our trivial example had two possible states of the world. If there were ten boolean variables we cared about, there would be over one thousand possible states to sum across. Over one million for 20 booleans; over one billion for 30 booleans; you get the idea, and we must do that at *each* time step. For complex worlds, this can get really unreasonable.

Acknowledgements and thanks to Professors Mark Riedl and Jim Rehg of the Georgia Tech School of Interactive Computing.