Unit 6
Decision Making Under Uncertainty – Bayes Nets

So far, we have been dealing with AI problems where we assume the values we sense or measure for state variables are all reliable and accurate. Unfortunately, outside of simulated environments, the AI problems we may need to solve will typically involve some amount of uncertainty.

A few possible sources of uncertainty include:

- Noise – this will affect the values of the state variables the agent uses to plan and take actions
- Imperfect model of the problem – Variables that are either not measurable, or have not been included in the state representation the agent has access to
- Limitations in sensors or in the ability of the agent to observe certain state variables
- Incorrect assumptions about the world – e.g. assuming a static environment where in reality the environment may be changing.

The question that concerns us is – ***how can we make optimal decisions under uncertainty***.

As it happens, the optimal tool for this is ***Probability***. We will formulate our decision making process as a ***probabilistic inference*** process, in which:

- Given all the available (and likely noisy) information we can obtain about the environment
- We attempt to ***infer the most likely state*** for the variables we are interested in

Decision making then uses these inferred states to take suitable actions.

In order for us to understand how this process works, first we have to review a few probability concepts we will need:

- *A Random Variable* – it's a variable whose value depends on the outcome of a random event. E.g. the current temperature in Toronto which will depend on the current, variable, and hard to predict weather.

* Random variables have a domain, and are represented by probability distributions. A PDF for continuous variables, and a PMF for discrete variables. Here we will focus on discrete random variables.
  The probability distribution tells us what is the likelihood of the random variable taking on any of the values in its domain.

* Most problems of interest will involve multiple random variables interacting in some way, so we need a way to represent the probability of observing that at any given time these variables have specific values.

- *Joint Distribution* – $p(x_1 = v_1, x_2 = v_2, ..., x_k = v_k)$ the joint probability distribution gives the likelihood that the set of variables in our problem takes on a specific set of values. Recall that our variables are discrete, so the joint probability distribution is given by a table with ***k*** dimensions, in this

table there's an entry for each possible combination of values for the variables in our problem, and this entry gives the probability of that specific combination happening.

Example: Suppose we have two very simple random variables – T (temperature), and W (weather). T can take the values {'hot','cold'}, and W can take values {'sunny','rainy'}. A joint probability table would for these two variables would look like:

p(T,W)	W='sunny'	W='rainy'
T='hot'	.4	.1
T='cold'	.2	.3

Notice that, $p(x_1, x_2, ..., x_k) \geq 0$ and $\sum_{x_1,x_2,...,x_k} p(x_1, x_2, ..., x_k) = 1$

The first statement simply says that all probabilities are positive or zero, and the second implies that all possible outcomes are accounted for – the sum of probabilities over the entire table adds up to 1.

One important factor to keep in mind is that the size of the joint probability table grows exponentially with the number of variables. For a problem with ***k*** variables, each of which has a domain of size ***d***, the joint probability table contains d^k entries.

- *Events* – These are ***possible outcomes*** (specific values for variables in our problem). We are typically interested in the probability of specific outcomes happening. E.g. 'What is the probability that the mouse will get eaten if it chooses to move to the right'.

In the example above, we could come up with a number of events we may find relevant:

$$p(T = hot, W = sunny)$$
$$p(T = cold)$$
$$p(W = rain)$$

Notice that we do not necessarily care about specifying the value of ***all*** the variables in the problem. We may not care about whether it's hot or cold outside, just whether or not it's likely to be rainy (in which case we may need an umbrella).

The probabilities for events can be computed directly from the joint probability table – by adding up the probabilities of all entries that match the event we are interested in. This is called ***marginalization***. In the example above, to obtain the probability that the weather is rainy, we go to the table and add up the probabilities for 'rainy and hot', and 'rainy and cold' - we marginalize over temperature since we don't care about its value.

- *Conditional Probability* – Most of the time, the kind of events we are interested in results from having *some* information about the state of the world – as noted earlier, the information we have may be noisy, or incomplete, and there may be variables we can't observe.

We are interested in estimating the value of *unobserved* variables *given the observed values for other variables* involved in our problem. We are in effect, trying to estimate the conditional probability

$$p(a|b) = \frac{p(a,b)}{p(b)}$$

Notice that we can estimate this probability if we have access to the joint probability table. In the earlier example:

$$p(W = rain|T = cold) = \frac{.3}{.5} = .6$$

Notice that this 60% probability that the weather is rainy *given that we have observed it's cold outside* is not in the joint probability table.

- *Probabilistic Inference* – Observe (measure) values for a subset of the variables in our problem. Then compute conditional probabilities for the remaining variables from the joint distribution. As more information becomes available (more variables have been measured, or more accurate measurements have been made) the conditional probabilities are updated accordingly.

- *Product Rule* – We can *factor* the joint distribution for a set of variables as follows:

$$\begin{aligned} p(x_1, x_2, x_3, ..., x_k) &= \prod_i p(x_i|x_1, x_2, ...x_{i-1}) \\ &= p(x_1)p(x_2|x_1)p(x_3|x_2, x_1) \cdots p(x_k|x_1, x_2, ...x_{k-1}) \end{aligned}$$

The factorization is not unique – we can factor the joint in any order we please.

- *Conditional Independence* – As noted above, the joint probability table grows exponentially with the number of variables in our problem. In practice it's often very difficult to fill this table.

In order to make the inference process more manageable, we often ***model*** specific dependencies between variables in our problem, and take advantage of ***conditional independence*** relationships that will allow us to ***factor the joint probability*** into smaller/easier to manage conditional probability tables.

A classic example of this idea is a problem containing three variables

S – Smoke has been detected in a room (True/False)
F – There is a fire in the room (T/F)
A – A smoke alarm is ringing (T/F)

The joint probability for this problem can be factored as

P(S,F,A)=p(S)p(F|S)p(A|F,S)

Clearly, the state of the alarm depends on ***both the smoke and the fire variables***. That makes sense, the presence of a fire would increase the likelihood a smoke alarm is sounding, likewise, the presence of smoke would increase the likelihood of the smoke alarm going off.

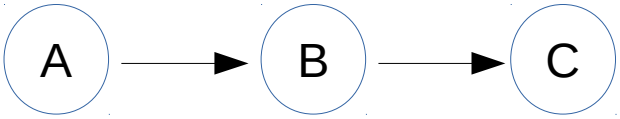
However, suppose we observe smoke (that is, we ***know that S=True***). This has the effect of de-coupling the state of the alarm from the value of the F (fire) variable. The presence of smoke has the effect of increasing the likelihood that the alarm will go off, whether or not there is a fire in the room!

We say that ***the value of A (alarm) is conditionally independent of F (fire), given the value for S (smoke)***. This may not seem like a very big change – but in a problem with many variables, conditional independence can greatly simplify our factorization of the joint probability.

Bayes Nets

Bayesian Networks (also called Belief Networks, or more succinctly Bayes Nets) are graphical models used to represent the dependencies between variables in a probabilistic inference problem, with the goal of allowing us to take advantage of conditional and marginal independence relationships so as to simplify the inference process.

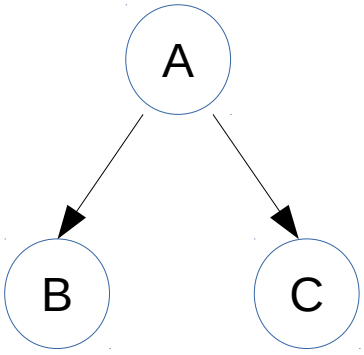
The building blocks of Bayes Nets are triplets of variables in specific configurations:



Indirect cause – The fire alarm example above fits this category, F (fire) causes S (smoke) which triggers an A (alarm).

The joint probability for the triplet of variables above is factored as

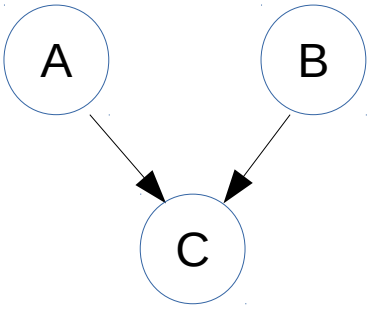
$p(A,B,C)=p(A)p(B|A)p(C|B)$



Common cause – Models two variables whose state depends on a single common cause. Classic example is a pair of symptoms of a disease known to cause both

The joint probability for the triplet of variables above is factored as

$p(A,B,C)=p(A)p(B|A)p(C|A)$



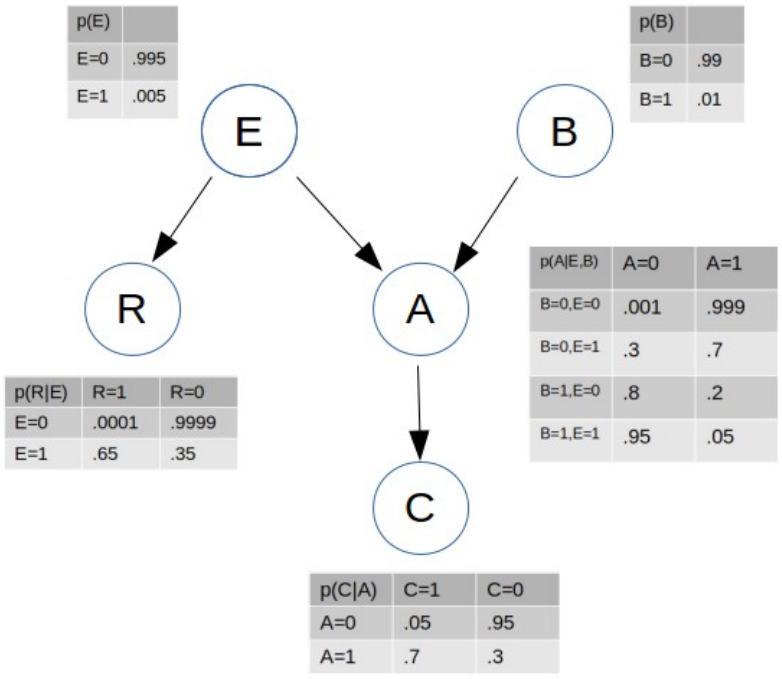
Common effect – A single variable whose state could be explained by two possible causes. An classic example is a specific symptom (e.g. fever) which could be explained by two different illnesses

The joint probability for the triplet of variables above is factored as

$p(A,B,C)=p(A)p(B)p(C|A,B)$

Bayes Nets consist of larger groups of variables related by one of the three types of relationships shown above.

Here's a sample Bayes Net showing how it models relationships between variables, how we would need to specify conditional probability tables for each variable involving only the variables the respective template indicates are relevant, and how the structure of the network results in a simpler factorization of the joint probability for the set of variables.



p(E)	
E=0	.995
E=1	.005

p(B)	
B=0	.99
B=1	.01

p(R|E)	R=1	R=0
E=0	.0001	.9999
E=1	.65	.35

p(A|E,B)	A=0	A=1
B=0,E=0	.001	.999
B=0,E=1	.3	.7
B=1,E=0	.8	.2
B=1,E=1	.95	.05

p(C|A)	C=1	C=0
A=0	.05	.95
A=1	.7	.3

Inference in Bayesian Nets

Once we have a model for our problem in the form of a Bayes Net, we can use it to perform inference. The kind of questions we wish to ask are of the type:

- What is the most likely value of variables A, B, ... X
  (for an arbitrary subset of variables in our problem)
- If I have observed values for certain variables, how does that change the most likely value for the rest?
- If I have to make a decision based on my current observations, what is the optimal decision I can make?

To answer these questions, we go back to the structure of the Bayes Net, and perform inference on it given any observations we have, and the structure of the network telling us how variables relate to each other.

The simplest process for performing inference on a given Bayes Net is to do ***sampling.*** In particular, we will random sample possible values for all the ***unobserved*** variables in the network using the associated ***conditional probability tables*** at each node in the net.

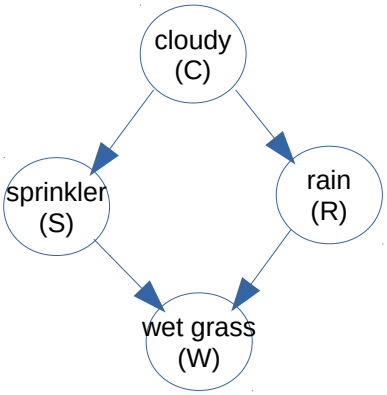
```
Bayes Net Sampling:

   For N samples (with N suitably large)
       - Randomly draw values for each of the unobserved variables in the network
       - Determine the weight of each sample (more on this below)

   For each variable
       - Create a histogram of possible values (by counting how many times each
         possible value was observed in our random samples, multiplied by the
         corresponding weight for each sample).
       - Normalize the histogram to obtain a probability distribution of values
         for that variable
       - Determine the most likely value from the histogram
```

Example:

Consider the Bayes Net shown below



This network could be used to answer questions such as:

- If the grass is wet, what's the probability the sprinklers were on?
- If it's cloudy, what's the probability it will rain?
- If the grass is wet, what's the probability it's cloudy?

And so on. Let's assume that each of these four variables is Boolean, it can be TRUE, or FALSE. To fully determine the possible outcomes for variables in the network, we need four probability tables:

P(C) - Which depends on nothing else on the network
P(S|C) - Sprinkler being on depends on whether it's cloudy or not
P(R|C) - Whether it is rainy or not depends on whether it's cloudy or not
P(W|S,R) - Whether the grass is wet will depend on both rain and sprinklers

Let's put values in these tables (it's just an example, I didn't get stats from anywhere)

P(C)	
C=0	.7
C=1	.3

P(S|C)	C=0	C=1
S=0	.8	.5
S=1	.2	.5

P(R|C)	C=0	C=1
R=0	.99	.2
R=1	.01	.8

P(W|S,R)	S=0,R=0	S=0,R=1	S=1,R=0	S=1,R=1
W=0	.99	.1	.05	.01
W=1	.01	.9	.95	.99

How do we ***get a random sample*** from these tables and the structure of the network?

* Start with variables that do not depend on anything (in this case, C)
  - We get a random number ***r*** , if ***r<=.7, C=0,*** otherwise ***C=1***
    e.g. r=.46271, C=0

  Our sample at this point looks like so:

  C=0 S=? R=? W=?

* Now sample variables that depend on only ***one*** other variable (in this case, S and R which depend on C).

  - For sampling ***S***, get a random number ***r***. Then look in the CPT for P(S|C) and find the column for ***C=0*** which is the value we got from our random sample for C. There we find that if ***r<=.8, S=0, otherwise S=1***.
    e.g. r=.9231, so S=1

  Our sample at this point looks like so:

C=0 S=1 R=? W=?

- For sampling ***R***, get a random number ***r***. Then look in the CPT for P(R|C) and find the column for ***C=0*** which is the value we got from our random sample for C. There we find that if ***r<=.99, R=0, otherwise R=1***.
e.g. r=.1231, so R=0

Our sample at this point looks like so:

C=0 S=1 R=0 W=?

* Now sample any variables that depend on ***two other variables***, in this case, ***W***.

- Get a random number ***r***. Then look in the CPT for P(W|S,R) and find the column for ***S=1, R=0*** which are the current values for these variables in our sample. We see that if ***r<=.05, W=0, otherwise W=1***.
e.g. r=.31415, so W=1

And we have a complete sample! It has values

C=0 S=1 R=0 W=1

The weight for this sample is 1.0 because ***all variables were randomly sampled from the corresponding tables.***

We repeat the process above until we have obtained *N* random samples for possible assignments to variables in our network, we'll have a table that looks like so:

C	S	R	W	Weight
0	1	0	1	1.0
0	0	0	0	1.0
1	0	1	1	1.0
.	.	.	.	.
.	.	.	.	.
1	1	0	0	1.0
0	1	0	0	1.0

Suppose we have a few hundred thousand samples, or a couple million. Then we can approximate very closely the actual probability of any arbitrary setting for any subset of variables in the network from this table.

For example, say we want to know p(R=1), this is just:

$$P(R=1) = \frac{\#\ of\ rows\ in\ the\ table\ with\ R{=}1}{\#\ of\ rows}$$

Say we want to find out P(C=1, W=0), this is just:

$$P(C=1, W=0) = \frac{\#\ of\ rows\ in\ the\ table\ with\ C{=}1, W{=}0}{\#\ of\ rows}$$

It becomes a simple matter of counting how often the outcome we're interested in actually happens in our random sample!

Of course this is just an approximation, and how good it is depends on how much time we want to spend doing random sampling. But with current computers, drawing millions of samples for even a large network is not unreasonable.

Still, there may be outcomes which are unlikely, and if we don't do enough sampling we may never actually observe them in our random sample set. So for some networks, and in some cases, it may require a fairly large amount of sampling to get accurate probabilities for low-probability events.

Note: We can also ask questions about conditional probabilities – by using the identity P(A|B)=P(A,B)/P(B). We can also use Bayes rule, together with the conditional probability definition, to infer posterior probabilities as needed.

What about observed variables?

If you pay attention to the samples above, you'll see that we have a weight column with all weights equal to 1.0. This is because in all our random samples, all variables were randomly sampled from their corresponding probability tables.

This is wasteful when we have observations about any of the variables in our network. For instance, say we have ***observed R=1.*** In other words, we actually know it's raining.

We can follow the random sampling process above, now, suppose we want to find out what's the probability of ***C=1***.

Before observing that it is rainy, we would have counted every row in which C=1, and divided by the total number of rows.

$$P(C=1) = \frac{\#\ of\ rows\ in\ the\ table\ with\ C{=}1}{\#\ of\ rows}$$

After observing that R=1, we have

$$P(C=1|R=1) = \frac{\#\ of\ rows\ in\ the\ table\ with\ C{=}1, R{=}1}{\#\ of\ rows with R{=}1}$$

This follows from the conditional probability definition, since we are now given the knowledge that R=1.

Inference still works exactly the same as we did before, and we ***could continue to use our regular sampling process, with equal weights for all samples, just like before***.

However, notice that a large proportion of the samples in our table will now go unused – they contribute nothing to our probability estimate because they happen to have R=0, the value of R we know not to be correct.

This can get pretty bad if the ***observed values for variables given to us*** have low probability of occurring. In this case, most of the entries in our sample table could be useless – they do not correspond to the observed values we have.

So, either we sample a whole lot, or we get smart about sampling!

Importance sampling for networks with observed variables

Let's do things smartly – we will modify (slightly) our sampling process to ***not generate any samples that correspond to the wrong values of observed variables***. Simply put, we will ***not sample*** the observed variables, ***using instead their observed value***.

That means ***every sample*** in our table will now have the correct values for observed variables, and no samples are wasted – we will be able to obtain a much better estimate of any probabilities we're interested in for the remaining variables. ***But***, the samples are no longer ***fair***. We are not sampling certain variables, so we need to account for the probability of these ***observed variables*** having the value they have, given the rest of the variables in the network.

We will do this by computing a weight for each sample that accounts for how likely the observed variable values are given all the values that were ***randomly sampled***.

Let's see an example of how this would work in the case above. Suppose we observe R=1

P(C)	
C=0	.7
C=1	.3

P(S|C)	C=0	C=1
S=0	.8	.5
S=1	.2	.5

P(R|C)	C=0	C=1
R=0	.99	.2
R=1	.01	.8

P(W|S,R)	S=0,R=0	S=0,R=1	S=1,R=0	S=1,R=1
W=0	.99	.1	.05	.01
W=1	.1	.9	.95	.99

We use random sampling (as described above) to get random assignments for C, S, and W (note that for W we will use R=1). Suppose we obtain the following sample:

C=0 S=0 R=1 W=1

The value for R is shown in red because it was ***not sampled***, it's ***fixed for all our samples because it was observed***.

The sample above is a perfectly possible sample, however, we need to weight it by the probability that R=1 given the values for C, S, and W in that specific random sample.

R depends only on C, so we head over to the probability table for P(R|C), and we look at the column for C=0 (which is the value in this specific random sample), there we find that P(R=1 | C=0)=.01.

So the weight for this sample is:

weight = 1.0 * P(R=1 | C=0) = 1.0 * .01 = .01

The 1.0 corresponds to the weight of a standard sample, in which all variables were randomly sampled. Then we multiply that for the conditional probability of R=1 given the sample's other values. The final weight is .01 which reflects the fact that there's a very low chance of observing rain if it's not cloudy!

Let's do another sample:

C=1 S=0 R=1 W=1

From P(R|C) we see that P(R=1|C=1)=.8

So the weight for this sample would be:

weight = 1.0 * P(R=1 | C=1) = 1.0 * .8 = .8

This weight is telling us the sample above is a lot more likely as a possible assignment to the variables in our network.

We would repeat this process N times, obtain N total samples all of which has R=1, and each of which has an appropriate weight, and then we can infer probabilities for any specific assignment of variables in the network by slightly modifying what we had before so it accounts for the weights of samples. For example, say we want to know P(C=0):

$$P(C=0) = \frac{\sum_{rows\ with\ C=0} w_{row}}{\sum_{all\ rows} w_{row}}$$

That is, instead of simply counting and dividing by the total number of samples, now we accumulate the weights for all rows with C=0, and divide that by the sum of the weights of all the rows in the table.

Similarly, you can compute the probability of any other possible outcome by adding up the weights of the rows with that specific outcome, and dividing by the sum of the weights of all the rows.

Conditional probabilities and Bayes rule can be applied like before!

The advantage of this process is that we do not waste samples on assignments that do not correspond to the observed values of variables. This means we'll obtain an accurate estimate of the probabilities we care about with far less sampling. But, we should be able to obtain the same probability estimates using the regular sampling process (random sampling all variables) if we sample enough!

In summary: Inference in Bayes Nets can be carried out by random sampling and counting – it's a general procedure, works for any network, and requires us only to have enough computing power/time to obtain enough samples that our probability estimates are meaningful. We can also use importance sampling to obtain good estimates with less sampling by using the observed values of some variables, and weighting the samples accordingly.

Now go and try it out!

Problems:

1) Figure out what the sampling process would need to do for the examples above, if we observe W=1, R=0. What would be the weight of a sample C=0, S=1, R=0, W=1?

2) Write a little program to estimate the probability of specific events in the network above. It has to carry out the random sampling process, and provide you with a table from which you can compute the corresponding probabilities.

You can do this in your favorite language :) - I don't have enough time at this point to give you Matlab starter, so off you go to do it however you prefer – but do it! Then you'll fully understand the sampling process and the probability estimation process.