6CCS3AIN, Tutorial 03 (Version 1.0)

Several of these questions make use of the Bayesian network in Figure 1.

1. Write down a Bayesian network that captures the knowledge that (a) Smoking causes cancer; and (b) Smoking causes bad breath. Given the information that:

$$P(cancer|smoking) = 0.6$$
$$P(badBreath|smoking) = 0.95$$

   and $P(smoking) = 0.2$ use a Naive Bayes model to determine $P(smoking, cancer, badBreath)$.

   Assume that all three variables are binary: $Smoking$ has values $smoking$ and $\neg smoking$, $Cancer$ has values $cancer$ and $\neg cancer$, and $BadBreath$ has values $badBreath$ and $\neg badBreath$.

2. Write down a Bayesian network that captures the knowledge that (a) a Late Start can cause a student to Fail their Project, and (b) Ignoring Advice can cause a student to Fail their Project. Use the binary variables $LateStart$, with values l and $\neg l$, $IgnoreAdvice$, with values i and $\neg i$, and $FailProject$, with values f and $\neg f$.

   You know that $LateStart$ and $IgnoreAdvice$ are non-interacting cuases of $FailProject$, so use a Noisy-Or model to build the conditional probability table relating the three variables from:

$$P(f|l) = 0.7$$
$$P(f|i) = 0.8$$

3. How many numbers do we need to specify all the necessary probability values for the network in Figure 1?

   How many would we need if there were no conditional independencies (if we didn't have the network)?

4. Compute the joint probability $P(m, \neg t, h, s, \neg c)$

5. Use the enumeration algorithm to compute the probability of $P(m|h, s)$.

6. Use prior sampling to compute the joint probability over m, t, h, s and c.

   Use the sequence of random numbers in Table 1. Give the results of the first 5 samples only.

   If you need more random numbers than exist in Table 1, start that sequence again from the beginning.

7. Use rejection sampling to compute $P(m|h, s)$

   Use the sequence of random numbers in Table 1, starting at the beginning of the sequence. This time you should show the results of the first 5 samples that aren't rejected.

   If you need more random numbers than exist in Table 1, start that sequence again from the beginning.

8. Use importance sampling to compute $P(m|h, s)$.

   Use the sequence of random numbers in Table 1, starting at the beginning of the sequence, and again report the results of the first five samples.

   If you need more random numbers than exist in Table 1, start that sequence again from the beginning.

9. Compute $P(m|h, s)$ using Gibbs sampling.

   Use the sequence of random numbers in Table 1, and give the results of the first 5 samples only.

   If you need more random numbers than exist in Table 1, start that sequence again from the beginning.

10. For the optional computational part of the tutorial, download the file `wetGrass.py` from KEATS.

    This provides code, using `pomegranate`[1], that implements the "Wet Grass" example from Lecture/Week 3. You can run the example using:

[1] For details on `pomegranate` see Tutorial 2.

python wetGrass.py

That should give you this as output:

```
Evidence is:  Grass is wet.

Here's the prediction:
Cloudy, Sprinkler Off, Rain, Wet grass.
And by the numbers:
Cloudy:  (('c', 0.6010250059300832), ('l', 0.3989749940699168))
Sprinkler:  (('f', 0.5976922222293125), ('n', 0.40230777777068755))
Rain:  (('r', 0.7772459441092383), ('n', 0.2227540558907617))
```

The first part of `wetGrass.py` sets up the Bayesian network. This is a straightforward extension of what we had for `underwear.py`.

The second part then runs a query on the model. The variable `scenario`:

scenario = [[None, None, None, 'w']]

specifies the observed values (the evidence), if any, of the variables in the model — the order of the variable is the order specified in the `model.add_states()` command. Here we are setting the variable `WetGrass` to value `w`, denoting that the grass is wet, and not saying anything about the values of `Cloudy`, `Sprinkler` and `Rain`. This:

```
Evidence is:  Grass is wet.
```

reflects the input back — code does this in a somewhat general way[2].

The code uses two methods to do inference given the evidence. First `model.predict()` identifies the most likely (highest probability) value of each variable in the model:

```
Here's the prediction:
Cloudy, Sprinkler Off, Rain, Wet grass.
```

and then `model.predict_proba()` is used to generate the probabilities of all the values of the non-evidence variables:

```
And by the numbers:
Cloudy:  (('c', 0.6010250059300832), ('l', 0.3989749940699168))
Sprinkler:  (('f', 0.5976922222293125), ('n', 0.40230777777068755))
Rain:  (('r', 0.7772459441092383), ('n', 0.2227540558907617))
```

It turns out that `model.predict_proba()` handles evidence variables differently to non-evidence variables, and so the code doesn't output any information about evidence variables (we already know the value taken by evidence variables after all).

The final part of the code does some minimal formatting on what is generated by `model.predict()` and `model.predict_proba()` to produce the output above.

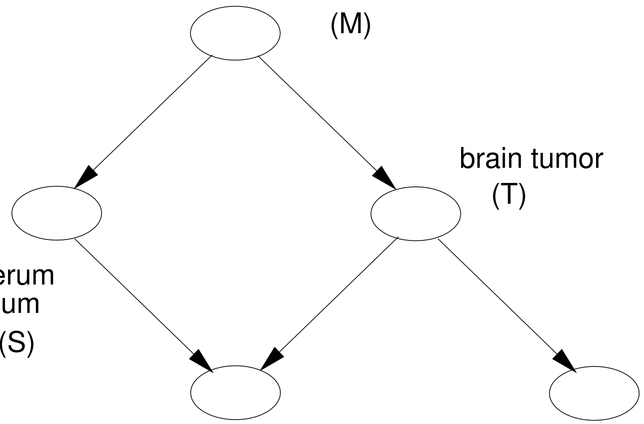
(a) Use `wetGrass.py` to calculate the probability of the variables in the model if:
   i. The grass is wet and the sprinkler is on.
   ii. The sprinkler is on and the sky is cloudy.
   iii. It is raining.

(b) Use `pomegranate` to build the model in Figure 1. Use this model to calculate $P(m|h, s)$.

[2] The documentation for `pomegranate` is not great, but from what I can tell, there is no function that gives access to the states in a model once the model has been built, so we can't retrieve the variables, and their names and values in a general way. (We could build our own structures to keep track of this, of course). Hence the way that they are hard-coded.

0.14	0.57	0.01	0.43	0.59	0.50	0.12	0.54	0.97	0.51	0.49	0.67	0.96	0.55	0.89	0.21	0.34

Table 1: Some random numbers between 0 and 1.



$P(M)$
0.1

M	$P(S \mid M)$
T	0.8
F	0.2

M	$P(T \mid M)$
T	0.7
F	0.1

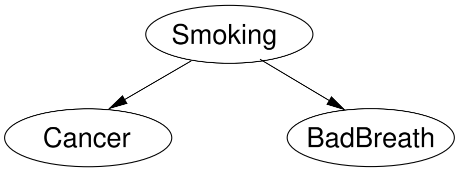
S	T	$P(C \mid S, T)$
T	T	0.95
T	F	0.85
F	T	0.85
F	F	0.01

T	$P(H \mid T)$
T	0.9
F	0.7

Figure 1: The example Bayesian network.

6CCS3AIN, Tutorial 03 Answers (Version 1.0)

1. The Bayesian network is:



We are told that:

$$
\begin{aligned}
Psmoking) &= 0.2 \\
P(cancer|smoking) &= 0.6 \\
P(badBreath|smoking) &= 0.95
\end{aligned}
$$

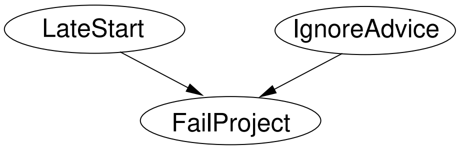
Now, the Naive Bayes model tells us that:

$$\mathbf{P}(Cause, Effect_1, \ldots, Effect_n) = \mathbf{P}(Cause) \prod_i \mathbf{P}(Effect_i|Cause)$$

so:

$$
\begin{aligned}
P(smoking, cancer, badBreath) &= P(smoking) \cdot P(cancer|smoking) \cdot P(badBreath|smoking) \\
&= 0.2 \cdot 0.6 \cdot 0.95 \\
&= 0.114
\end{aligned}
$$

2. The Bayesian network is:



3. With no network, 5 variables need:

$$2^5 - 1 = 31$$

numbers. With the network, we need:

$$1 + 2 + 2 + 4 + 2 = 11$$

4. To get the joint probability over the set of variables in the network we apply the chain rule as on the slides. We have:

$$
\begin{aligned}
P(\neg m, \neg t, \neg h, \neg s, \neg c) &= P(\neg h|\neg t) \cdot P(\neg c|\neg s, \neg t) \cdot P(\neg t|\neg m) \cdot P(\neg s|\neg m) \cdot P(\neg m) \\
&= 0.9 \cdot 0.8 \cdot 0.9 \cdot 0.99 \cdot 0.3 \\
&= 0.192456
\end{aligned}
$$

Note that the 0.15 in the second line above is computed from $1 - P(c|s, \neg t)$. Similarly 0.3 is computed from $1 - P(t|m)$.

5. From the slides we have:

$$
\begin{aligned}
\mathbf{P}(M|h, s) &= \frac{\mathbf{P}(M, h, s)}{P(h, s)} \\
&= \alpha \mathbf{P}(M, h, s) \\
&= \alpha \sum_t \sum_c \mathbf{P}(M, h, s, t, c)
\end{aligned}
$$

Factorising as in Q4, we then have:

$$\begin{aligned}
\mathbf{P}(M|h,s) &= \alpha \sum_t \sum_c P(h|t) \cdot P(c|s,t) \cdot \mathbf{P}(t|M) \odot \mathbf{P}(s|M) \odot \mathbf{P}(M) \\
&= \alpha \mathbf{P}(M) \odot \mathbf{P}(s|M) \odot \sum_t \sum_c P(h|t) \cdot P(c|s,t) \cdot \mathbf{P}(t|M) \\
&= \alpha \begin{pmatrix} P(m) \cdot P(s|m) \sum_t \sum_c P(h|t) \cdot P(c|s,t) \cdot P(t|m) \\ P(\neg m) \cdot P(s|\neg m) \sum_t \sum_c P(h|t) \cdot P(c|s,t) \cdot P(t|\neg m) \end{pmatrix}
\end{aligned}$$

Let's say that:

$$\begin{aligned}
pm &= P(m) \cdot P(s|m) \sum_t \sum_c P(h|t) \cdot P(c|s,t) \cdot P(t|m) \\
&= P(m) \cdot P(s|m) \left(P(h|t) \cdot P(t|m) \cdot P(c|s,t) + P(h|t) \cdot P(t|m) \cdot P(\neg c|s,t) + \right. \\
&\qquad \left. P(h|\neg t) \cdot P(\neg t|m) \cdot P(c|s,\neg t) + P(h|\neg t) \cdot P(\neg t|m) \cdot P(\neg c|s,\neg t)\right) \\
&= 0.1 \cdot 0.8(0.9 \cdot 0.7 \cdot 0.95 + 0.9 \cdot 0.7 \cdot 0.05 + 0.7 \cdot 0.3 \cdot 0.85 + 0.7 \cdot 0.3 \cdot 0.15) \\
&= 0.0672
\end{aligned}$$

Similarly, let:

$$\begin{aligned}
pm' &= P(\neg m) \cdot P(s|\neg m) \sum_t \sum_c P(h|t) \cdot P(c|s,t) \cdot P(t|\neg m) \\
&= 0.9 \cdot 0.2(0.9 \cdot 0.1 \cdot 0.95 + 0.9 \cdot 0.1 \cdot 0.05 + 0.7 \cdot 0.9 \cdot 0.85 + 0.7 \cdot 0.9 \cdot 0.15) \\
&= 0.1296
\end{aligned}$$

Now:

$$\mathbf{P}(M|h,s) = \alpha \begin{pmatrix} pm \\ pm' \end{pmatrix} = \alpha \begin{pmatrix} 0.0672 \\ 0.1296 \end{pmatrix} = \begin{pmatrix} 0.34 \\ 0.66 \end{pmatrix}$$

6. To compute the first sample:

   - Sample m.

     $P(m)$ is 0.1, so we want to generate m one time in 10 and $\neg m$ 9 times in 10. If we generate a random number with equal probability of picking any number between 0 and 1, then that number will be less than or equal to 0.1 1 time in 10, and greater than 0.1 9 times in 10.

     So, our method for picking whether m is true or false is to compare a random number (picked between 0 and 1, inclusive, with equal probability for everry number in that range) with $P(m)$. If the random number is less than or equal to $P(m)$, then m, is true. Otherwise m is false.

     Our first random number is 0.14, so m is false.

   - Sample s.

     Given than we have $\neg m$, we now sample s given $\neg m$. $P(s|\neg m)$ is 0.2. Our random number is 0.57. So s is false.

   - Sample t.

     Similarly to the previous case, $P(t|\neg m) = 0.1$, and our random number is 0.01, less than 0.1 So t is true.

   - Sample c.

     We sample c given $\neg s$ and t. $P(c|\neg s, t) = 0.85$ and the next random number is 0.43, so c is true.

   - Sample h.

     Since t is true, we need to sample given t. $P(h|t) = 0.9$, our random number is 0.59, so h is true.

So we have a sample $(\neg m, \neg s, t, h, c)$.

Following the same procedure again we get, in turn, the samples:

$$\begin{aligned}
&(\neg m, s, \neg t, \neg c, h)\\
&(\neg m, \neg s, \neg t, \neg c, \neg h)\\
&(\neg m, \neg s, \neg t, \neg c, h,)\\
&(\neg m, \neg s, \neg t, \neg c, \neg h)
\end{aligned}$$

Therefore we estimate $\hat{P}(\neg m, \neg t, \neg h, \neg s, \neg c) = 2/5$, where we use the $\hat{P}$ to denote it's an estiamte.

The probabilities only become accurate after many iterations (as the number of iterations approaches infinity, the probability approaches the correct value).

7. Since we are computing $P(m|h, s)$ we will only use samples in which h and s are true. Using rejection sampling we proceed as before. We only need to consider the events where h and s both hold. There is only one. In this one m does not hold and so we estimate $\hat{P}(m|h, s) = 0$.

   Again, this is approximate, and will improve with more samples. Indeed, if you complete the 5 samples that the question asks for, you may get a better approximation.

8. For likelihood sampling, we start by picking an order in which we will evaluate the variables. We will use the same order as before.

   Then, starting at the beginning of the list of random numbers

   - Sample m
     m is false. (We use the 0.14 to get this — we are starting over with the random numbers again.).
   - s is true by definition.
     w is set to the value of $P(s|\neg m)$, so w is 0.2.
   - Sample t
     t is true.(We use 0.01 to get this.)
   - h is true by definition.
     Thus we update w with $P(h|t) = 0.9$. $w = 0.2 \cdot 0.9 = 0.18$.
   - Sample for c.
     c is true.

   So we have the sample $(\neg m, s, t, h, c)$ and the weight is 0.18.

   For the second sample: $(\neg m, s, \neg t, \neg c, h)$ and the weight is $0.2 \cdot 0.7 = 0.14$.

   For the third sample: $(\neg m, s, \neg t, c, h)$ and the weight is $0.2 \cdot 0.7 = 0.14$.

   For the fourth sample: $(\neg m, s, \neg t, \neg c, h)$ and the weight is $0.2 \cdot 0.7 = 0.14$.

   For the fifth sample: $(\neg m, s, \neg t, \neg c, h)$ and the weight is $0.2 \cdot 0.7 = 0.14$.

   So:

$$\hat{\mathbf{P}}(T|h, s) = \alpha \begin{pmatrix} 0.18 \\ 4 \cdot 0.14 \end{pmatrix} \approx \begin{pmatrix} 0.243 \\ 0.757 \end{pmatrix}$$

   Again I stress that these values are very approximate with so few samples. The values were very different after several thousand samples.

9. Again, I'm not going to post a solution for this optional bit of the tutorial, but if you did it, you can check the correctness of your solution against the value for $P(m|h, s)$ in question 5.