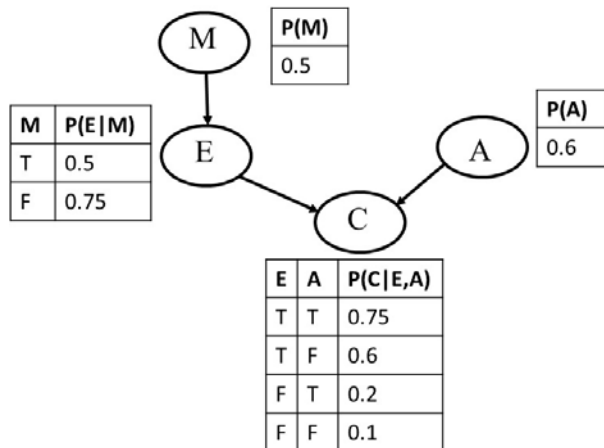# COMP3308/3608 Artificial Intelligence

Week 11 Tutorial exercises
Probabilistic Reasoning. Bayesian Networks.

Exercise 1. Bayesian network (Homework)

Consider the Bayesian network below where all variables are binary:

Compute the following probability and show your calculations: P(M=T, E=T, A=F, C=T).

Solution:
P(M=T, E=T, A=F, C=T) = P(m, e, ~a, c)=P(m)P(e|m)P(~a)P(c|e,~a)=0.5*0.5*(1-0.6)*0.6=0.06

Exercise 2. Probabilistic reasoning

Consider the dental example discussed at the lecture; its full joint probability distribution is shown below:

	toothache		¬ toothache	
	catch	¬ catch	catch	¬ catch
cavity	.108	.012	.072	.008
¬ cavity	.016	.064	.144	.576

Calculate the following:
   a) P(toothache)
   b) **P**(Cavity)
   c) **P**(Toothache |cavity)
   d) **P**(Cavity| toothache ∨ catch)

Solution:
   a) P(toothache)=P(Toothache=true)=0.108+0.012+0.016+0.064=0.2

   Note that:
   toothache is a shorthand for Toothache=true and
   ~toothcahe is a shorthand for Toothache=false

b) **P**(Cavity) is the probability distribution of the variable Cavity – i.e. a data structure, containing the probability for each value of Cavity. As Cavity has 2 values (true and false), **P**(Cavity) will have 2 elements – 1 for Cavity=true and 1 for Cavity=false. For binary variables, by convention, the first element is for true.

P(Cavity) = < P(Cavity=true), P(Cavity=false) >=
=< (0.108+0.012+0.072+0.008), (0.016+0.064+0.144+0.576) >=<0.2, 0.8>

c) **P**(Toothache |cavity) =
=< P(Toothache=true | Cavity=true), P(Toothache=false | Cavity=true) > =
=< P(toothache | cavity), P(~toothache | cavity) >

Method 1: Without α (i.e. by calculating the denominator)
Let's compute the first term:
P(toothache | cavity)=
= P(toothache, cavity)/P(cavity)= (from the def. of conditional probability: P(A|B)=P(A,B)/P(B))
=(0.108+0.012)/(0.108+0.012+0.072+0.008)=0.12/0.2=0.6

Now the second term:
P(~toothache | cavity)=
=P(~toothache, cavity)/P(cavity)=(0.072+0.008)/0.2=0.08/0.2=0.4

=> **P**(Toothache |cavity) =<0.6, 0.4>

Method 2: With α (i.e. without calculating the denominator)
Let's compute the two terms in another way, using the normalization term α:

P(toothache | cavity)= α P(toothache, cavity)=
=α (0.108+0.012) = α*0.12

P(~toothache | cavity)= α P(~toothache, cavity)=
=α (0.072+0.008) = α*0.08

=> **P**(Toothache |cavity) =α<0.12, 0.08>

Computing the normalization term: 0.12+0.08=0.2

=> **P**(Toothache |cavity) =<0.12/0.2, 0.08/0.2> =<0.6, 0.4>

d) **P**(Cavity| toothache ∨ catch) = < P(cavity| toothache ∨ catch), P(~cavity| toothache ∨ catch) >

Here we will do it without α; you can do this with α as an exercise at your own time.

Let's compute the first term:
P(cavity| toothache ∨ catch) = P(cavity, (toothache ∨ catch))/P(toothache ∨ catch)

Computing the denominator: P(toothache ∨ catch)=0.108+0.012+0.016+0.064+0.072+0.144=0.416

=> P(cavity| toothache ∨ catch)=(0.108+0.012+0.072)/0.416=0.4615

Now the second term:
P(~cavity| toothache ∨ catch) = P(~cavity, (toothache ∨ catch))/P(toothache ∨ catch)=

=(0.016+0.064+0.144)/0.416=0.5384

=> **P**(Cavity| toothache ∨ catch)=<0.4615, 0.5384>

Exercise 3. *Bayesian networks*

(Based on J. Kelleher, B. McNamee and A. D'Arcy, Fundamentals of ML for Predictive Data Analytics, MIT, 2015)

Given is the following description of the causal relationships between storms, burglars, cats and house alarms:

1. Stormy nights are rare.
2. Burglary is also rare, and if it is a stormy night, burglars are likely to stay at home (burglars don't like going out in storms).
3. Cats don't like storms either, and if there is a storm, they like to go inside.
4. The alarm on your house is designed to be triggered if a burglar breaks into your house, but sometimes it can be triggered by your cat coming into the house, and sometimes it might not be triggered even if a burglar breaks in (it could be faulty or the burglar might be very good).

a) Define the topology of a Bayesian network that encodes these relationships.

b) Using the data from the table below, create the Conditional Probability Tables (CPTs) for the Bayesian network from the previous step.

ID	STORM	BURGLAR	CAT	ALARM
1	false	false	false	false
2	false	false	false	false
3	false	false	false	false
4	false	false	false	false
5	false	false	false	true
6	false	false	true	false
7	false	true	false	false
8	false	true	false	true
9	false	true	true	true
10	true	false	true	true
11	true	false	true	false
11	true	false	true	false
13	true	true	false	true

c) Compute the probability that the alarm will be on, given that there is a storm but we don't know if a burglar has broken in or where the cat is.

Solution:
a) The Bayesian network below represents the causal relationships.
- There are 4 variables: Storms, Burglar, Cat and Alarm
- Storms directly affect the behavior of burglars and cats, so there is a link from Storm to Burglar and from Storm to Cat.
- The behavior of burglars and cats both affect whether the alarm is triggered, and hence there is a link from Burglar to Alarm and from Cat to Alarm.
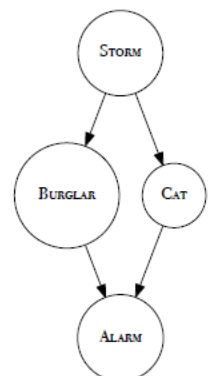
b) The CPTs are:

For node Storm (S):

s	~s
0.31	0.69

For node Burglar (B):

	b	~b
s	0.25	0.75
~s	0.33	0.67

For node Cat (C):

	c	~c
s	0.75	0.25
~s	0.22	0.78

For node Alarm (A):

		a	~a
b	c	1	0
b	~c	0.67	0.33
~b	c	0.25	0.75
~b	~c	0.2	0.8

c) We need to compute P(A=true| S=true).

$$P(a \mid s) = \frac{P(a,s)}{P(s)} = \frac{\sum\limits_{b,c} P(a,s,B_b,C_c)}{P(s)}$$

B and C are the hidden variables; $\sum\limits_{b} B_b$ means summation over the different values B can take.

Computing the nominator:

$$P(a \mid s) = \sum_{b,c} P(a,s,B_b,C_c) =$$

$$= \sum_{b,c} P(a \mid B_b,C_c)P(s)P(B_b \mid s)P(C_c \mid s) =$$ Using the Bayesian network assumption

$$= P(s)\sum_{b,c} P(a \mid B_b,C_c)P(B_b \mid s)P(C_c \mid s) =$$ Moving P(s) out of the summations
We have 4 sums, for:

$$= P(s)[P(a \mid b,c)P(b \mid s)P(c \mid s) +$$ b,c
$$P(a \mid b,\neg c)P(b \mid s)P(\neg c \mid s) +$$ b,~c
$$P(a \mid \neg b,c)P(\neg b \mid s)P(c \mid s) +$$ ~b,c
$$P(a \mid \neg b,\neg c)P(\neg b \mid s)P(\neg c \mid s)] =$$ ~b,~c

$$= 0.31 * [1 * 0.25 * 0.75 +$$
$$0.67 * 0.25 * 0.25 +$$
$$0.25 * 0.75 * 0.75 +$$
$$0.2 * 0.75 * 0.25] =$$
$$= 0.31 * (0.1875 + 0.0418 + 0.1406 + 0.0335) =$$
$$= 0.1251$$

The denominator: P(s)=0.31 (directly from CPT)

$$\Rightarrow P(a \mid s) = \frac{p(a,s)}{P(s)} = \frac{0.1251}{0.31} = 0.4035$$

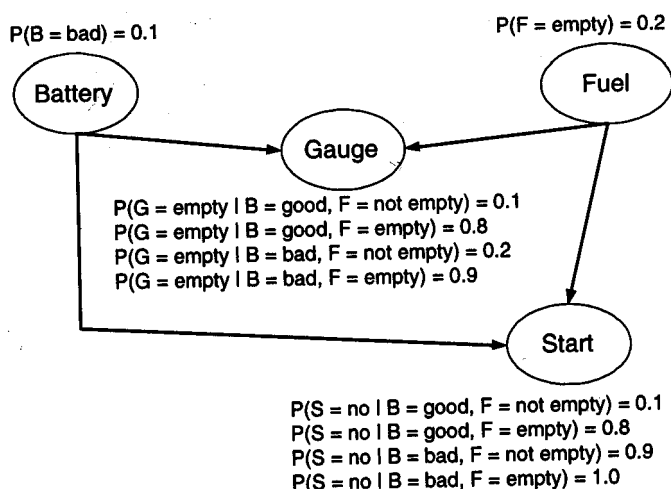Task: As an exercise at home, compute the same probability using α.

Note: We can slightly reduce the calculations by separating the two summations and pushing the summation over c in, like this:

$$P(a \mid s) = \sum_{b,c} P(a,s,B_b,C_c) =$$

$$= \sum_{b,c} P(a \mid B_b, C_c) P(s) P(B_b \mid s) P(C_c \mid s) =$$

$$= P(s) \sum_b P(B_b \mid s) \sum_c P(C_c \mid s) P(a \mid B_b, C_c) = \quad \text{<- Pushing the summation over c in:}$$

$$= 0.31 * \sum_b P(B_b \mid s)[0.75 * P(a \mid B_b, c) + 0.25 * 0.25 * P(a \mid B_b, \neg c)] = \quad \text{<- Summation for c and ~c}$$

$$= 0.31 * [0.25 * (0.75 * 1 + 0.25 * 0.67) + 0.75 * (0.75 * 0.25 + 0.25 * 0.2)] = \quad \text{<-Summation for b and ~b}$$

$$= 0.31 * (0.25 * 0.9175 + 0.75 * 0.2375) =$$

$$= 0.31 * 0.4075 =$$

$$= 0.1263$$

$$P(a \mid s) = \frac{p(a,s)}{P(s)} = \frac{0.1263}{0.31} = 0.4074$$

Exercise 4. *Using Bayesian networks for classification*

The figure below shows the structure and probability tables of a Bayesian network. Battery (B), Gauge (G) and Fuel (F) are attributes, and Start (S) is the class. If the evidence E is **B=good, F=empty, G=empty**, which class will the Bayesian network predict: **S=yes** or **S=no**?

P(B = bad) = 0.1

P(F = empty) = 0.2

Battery

Fuel

Gauge

P(G = empty | B = good, F = not empty) = 0.1
P(G = empty | B = good, F = empty) = 0.8
P(G = empty | B = bad, F = not empty) = 0.2
P(G = empty | B = bad, F = empty) = 0.9

Start

P(S = no | B = good, F = not empty) = 0.1
P(S = no | B = good, F = empty) = 0.8
P(S = no | B = bad, F = not empty) = 0.9
P(S = no | B = bad, F = empty) = 1.0

Hint: Compute the joint probabilities P(E, S=yes) and P(E, S=no), normalize them to obtain the conditional probabilities P(S=yes|E) and P(S=no|E), compare them and take the class with the higher probability as the class for the new example.

Solution:

1) Computing P(E, S=yes) and P(E, S=no):

P(E, S=yes)=P(B=good, F=empty, G=empty, S=yes)=
=P(B=good)P(F=empty)P(G=empty|B=good,F=empty)P(S=yes|B=good,F=empty)

As P(B=bad)=0.1 => P(B=good)=1-0.1=0.9
P(S=yes|B=good,F=empty)=1-0.8=0.2

=>P(E,S=yes)=0.9*0.2*0.8*0.2=0.0288

Similarly:
P(E, S=no)=P(B=good, F=empty, G=empty, S=no)=
=P(B=good)P(F=empty)P(G=empty|B=good,F=empty)P(S=no|B=good,F=empty)=
=0.9*0.2*0.8*0.8=0.1152

2) Normalisation:

P(S=yes|E)=0.0288/(0.0288+0.1152)=0.0288/0.144=0.2
P(S=no|E)=0.1152/(0.0288+0.1152)=0.8

3) Decision:
As 0.8>0.2, the Bayesian network predicts class **no** for the new example.

Another way to solve this – a smarter way for our example:

We need to compute these 2 conditional probabilities and compare them
P(S=yes| B=good, F=empty, G=empty)
P(S=no| B=good, F=empty, G=empty)

We know the values of the parents B and F (B=good, F=empty). The Bayesian network assumption says that "A node is conditionally independent of its non-descendants, given its parents". So if we know the values of the parents, the value of G is irrelevant (S is conditionally independent of G), i.e:
P(S=yes| B=good, F=empty, G=empty) = P(S=yes| B=good, F=empty)
P(S=no| B=good, F=empty, G=empty) = P(S=no| B=good, F=empty)

And we already have the values of these 2 probabilities from the CPT:
P(S=yes| B=good, F=empty)=1-0.8=0.2
P(S=no| B=good, F=empty)=0.8

Hence: P(S=yes| B=good, F=empty, G=empty)=0.2 and P(S=no| B=good, F=empty, G=empty)=0.8.
All done – no computations at all in this case!

Exercise 5. *Using Weka – Naïve Bayes and Bayesian networks*

Load the **weather nominal** data. Choose 10-fold cross validation. Run the Naïve Bayes classifier and note the accuracy. Run Bayesian net with the default parameters. Right-click on the history item and select Visualize graph. This graph corresponds to Naïve Bayes classifier, that's why the accuracy results are the same. This is because the parameter **maximum_number_of_parents_of_a node** is set to 1 in the K2 algorithm used to automatically learn the Bayesian network. Change it to 2 (for example) and a more complex Bayesian net will be generated and the accuracy will change (improves in this case).

Exercise 6. *Probabilistic reasoning (Advanced only) - to be done at home*

Twenty people flip a fair coin. What is the probability that at least 4 of them will get heads?

(From J. Kelleher, B. McNamee and A. D'Arcy, Fundamentals of ML for Predictive Data Analytics, MIT, 2015)

Hint: The probability of getting k outcomes, each with a probability p, in a sequence of n binary experiments is:

$$\binom{n}{k} * p^k * (1-p)^{n-k}$$

Solution:

- Probability that <u>at least</u> 4 people will get heads = 1 – probability that <u>less than</u> 4 people will get heads
- Probability that <u>less than</u> 4 people will get heads = sum of:
    1) Probability that <u>exactly</u> 3 people will get heads
    2) Probability that <u>exactly</u> 2 people will get heads
    3) Probability that <u>exactly</u> 1 person will get heads
    4) Probability that <u>exactly</u> 0 persons will get heads

We calculate each of these probabilities using the binomial formula:

1) $\binom{20}{3} * 0.5^3 * (1-0.5)^{20-3} = \dfrac{20!}{3!(20-3)!} * 0.5^3 * 0.5^{17} = 0.001087189$

2) $\binom{20}{2} * 0.5^2 * (1-0.5)^{20-2} = \dfrac{20!}{2!(20-2)!} * 0.5^2 * 0.5^{18} = 0.000181198$

3) $\binom{20}{1} * 0.5^1 * (1-0.5)^{20-1} = \dfrac{20!}{1!(20-1)!} * 0.5^1 * 0.5^{19} = 0.0000190735$

4) $\binom{20}{0} * 0.5^0 * (1-0.5)^{20-0} = \dfrac{20!}{0!(20-0)!} * 1 * 0.5^{20} = 0.0000009536$ (Note that 0!=1)

The last probability can also be calculated in another way: with 20 people flipping a coin, there are 2^{20} possible outcomes => P(no heads)=P(all tails)=$1/2^{20}$ (the same as above).

Then, the probability that <u>less than</u> 4 people will get heads is the sum of these 4 probabilities = 0.0012884241

Finally, the probability of <u>at least</u> 4 people getting heads is 1-0.0012884241=0.9987115759. This is very high: 99.87%.