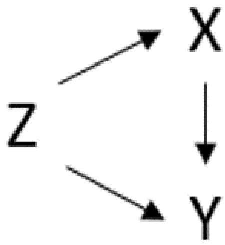
Advanced Artificial Intelligence Assignment Report

The solution to a) was derived from the Artificial Neural Networks, workshop 2's example of collecting probabilities from data.



$X=0$	$Z=1$	$P(Y \mid X,Z)$	$P(Y \mid X,\neg Z)$
T	T	0.81	0.19
T	F	0.48	0.52
F	T	0.83	0.17
F	F	0.55	0.45
$P(X=1)$	$P(Z=1)$	$P(Y \mid X,Z)$	$P(Y \mid X,\neg Z)$
T	T	0.76	0.24
T	F	0.55	0.45
F	T	0.87	0.13
F	F	0.54	0.46
$P(X=2)$	$P(Z=1)$	$P(Y \mid X,Z)$	$P(Y \mid X,\neg Z)$
T	T	0.91	0.09
T	F	0.55	0.45
F	T	0.77	0.23
F	F	0.54	0.46
Z	0.728	0.272	

No treatment		
Z	$P(x0 \mid Z)$	
TRUE	0.61	0.39
FALSE	0.04	0.96
Placebo		
Z	$P(x1 \mid Z)$	
TRUE	0.49	0.51
FALSE	0.08	0.92
Medicine		
Z	$P(x2 \mid Z)$	$P(x2/X \mid \neg Z)$
TRUE	0.09	0.91
FALSE	0.902	0.098
Y	0.746	0.254

This is possible for the values of X as well which give us a table of 12 elements. There are 12 outputs because X can be 0, 1, or 2. Therefore we have to derive the probabilities for Y|(X, Z) for all possible values of x. Similarly the same has to be done for Z = 1, 0 as demonstrated in the tables.

Given the full joint probability $P(x, y, z)$ of a set of random variables, it is possible to create a product of conditional probabilities from the expansion of the joint probability.

This procedure is called **chain rule** and the process involves decomposing the joint probability into a product of conditional probabilities.

The conditional probabilities are all written as the subject variable given its parents if it has any. If the variable does not have any parents then it represents itself.

According to the CPT, P (z) has no parents but the parent of x is z and the parent of y is x and z.

Hence: $P(x, y, z) = P(z)P(x|z)P(Y|x, z)$

To compute the intervention's probability for part b, I wrote a function that takes in the values of x,y and z and then uses the adjustment formula to compute the intervention for all given values of X=0,1,2 and Z = 1,0. The **adjustment formula** is also referred to as **casual effect rule**

This function also uses the Likelihood weighting to fix the value of Z = true and Y = true.

The function then returned the probability of Y being true and not being true for all cases of x and the two cases of z.

```
def y_do_x(y_x_z, y_x_not_z, y_not_x_z,y_not_x_not_z,z):
    y1_x1 = (y_x_z * z[t]) + (y_x_not_z * z[f])

    y1_x0 = (y_not_x_z * z[t]) + (y_not_x_not_z * z[f])

    return y1_x1, y1_x0
    #Computes Y TRUE/FALSE given Z is TRUE
    #AND Y TRUE/FALSE Given Z is false
```

```
y_do_x(p_y_x0_z1,p_y_x0_z0,p_y_x1_z1,p_y_x1_z0,Z)
```

```
(array([0.28259259, 0.71740741]), array([0.2998885, 0.7001115]))
```

```
y_do_x(p_y_x1_z1,p_y_x1_z0,p_y_x0_z1,p_y_x0_z0,Z)
```

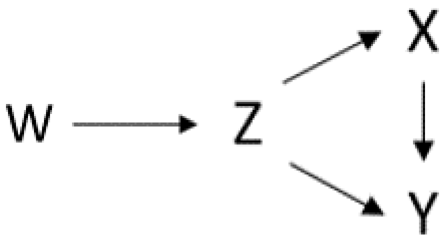
```
(array([0.2998885, 0.7001115]), array([0.28259259, 0.71740741]))
```

```
y_do_x(p_y_x2_z1,p_y_x2_z0,p_y_x2_z1,p_y_x2_z0,Z)
```

```
(array([0.18778731, 0.81221269]), array([0.18778731, 0.81221269]))
```

In the output window, the first array considers Y=1 and x = 1. This probability is indicated as Y = True in index [0] and Y = False in index [1]. The second array considers Y = 1 and x "**is not**" 1. This is indicated as Y = True in index [0] and Y = False in index [1]

For part C I based my solution on the workshop 3 example based on the condition that the new CPT was as follows:



The full probability distribution for $P(w, x, y, z)$ is $P(w)P(z|w)P(x|z)P(Y|x, z)$:

W in the full probability has no Parents and therefore the pre- and post-intervention are the same:

$$\mathbf{P}(y|do(w)) = \alpha\, \mathbf{P}(y|w)$$

Therefore it is impossible to use the adjustment formula on **W** to compute an Average casual Effect. Instead we compute: **P** (**Y**|**W**) for all possible values of **Y** and **W**

$$\begin{aligned}
\mathbf{P}(Y|w) &= \alpha\, \mathbf{P}(Y, w) \\
&= \alpha \sum_{x,z} \mathbf{P}(Y, x, z, w) \\
&= \alpha \sum_{x,z} \mathbf{P}(Y|x, z)P(x|z)P(z|w)P(z) \\
&= \alpha P(w) \sum_{x} P(x|z) \sum_{z} \mathbf{P}(Y|x, z)P(z|w)
\end{aligned}$$

```
#Y|do(x),W
def Y_W(y_x_z, y_not_x_z, y_not_x_not_z, y_x_not_z, x_z,x_not_z,z_w,w):
    y1_w1 = w[t]*(z_w[t,t]*(y_x_z[t] *x_z[t] + y_not_x_z*x_z[f])
                + z_w[f,t]*(y_x_not_z[t] *x_not_z[t] + y_not_x_not_z[t] *x_not_z[f]))

    y1_w0 = w[f]*(z_w[t,f]*(y_x_z[t] *x_z[t] + y_not_x_z*x_z[f])
                + z_w[f,f]*((y_x_not_z[t] *x_not_z[t] + y_not_x_not_z[t] *x_not_z[f])))

    return y1_w1 - y1_w0
```

```
sum1 = Y_W(p_y_x1_z1,P_of_not_x1_given_z1,p_y_not_x1_z0,p_y_x1_z0,p_x1_z1,p_x1_z0,P_Z_W,P_W)#y/x1
sum0 = Y_W(p_y_x0_z1,P_of_not_x0_given_z1,p_y_not_x0_z0,p_y_x0_z0,p_x0_z1,p_x0_z0,P_Z_W,P_W)#y/x0
sum2 = Y_W(p_y_x2_z1,P_of_not_x2_given_z1,p_y_not_x2_z0,p_y_x2_z0,p_x2_z1,p_x2_z0,P_Z_W,P_W)#y/x2

Output = sum0+sum1+sum2
Output
```

```
0.7082547442053746
```

References

Nicola, B. (2021) Lecture 3 – Bayesian Networks, University of Lincoln, 25th October. Available from: https://blackboard.lincoln.ac.uk/webapps/blackboard/content/listContent.jsp?course_id=_166752_1&content_id=_5852525_1 [Accessed: 02/12/21]

Nicola, B. (2021) Lecture 3 – Bayesian Networks II, University of Lincoln, 1st November. Available from: https://blackboard.lincoln.ac.uk/bbcswebdav/pid-6776705-dt-content-rid-12420500_2/xid-12420500_2 [Accessed: 02/12/21]

Nicola, B. (2021) Lecture 6 – Casual Inference I, University of Lincoln, 15th November. Available from: https://blackboard.lincoln.ac.uk/bbcswebdav/pid-6796404-dt-content-rid-12597360_2/xid-12597360_2 [Accessed: 02/12/21]