TDT4171 Assignment 1

Anders Fagerli

February 2, 2019

1 5-card Poker Hands

a) How many atomic events are there in the joint probability distribution (i.e., how many 5-card hands are there)?

- We have *n!* possible orderings of a n card deck, where we wish to see how many combinations of r card hands there are from the deck. The number of combinations is given by,

$$\binom{n}{r} = \frac{n!}{(n-r)!r!} \tag{1}$$

With 52 cards and a 5-card poker hand we have,

$$\binom{52}{5} = \frac{52!}{(52-5)!5!} = 2598960 \tag{2}$$

atomic events.

b) What is the probability of each event?

- The probability of each event in a uniform model is the fraction of the events n happening out of all possible events N,

$$P = \frac{n}{N} = \frac{1}{2598960} = 3.85 \cdot 10^{-7} \tag{3}$$

c) What is the probability of being dealt a royal straight flush? Four of a kind?

- **Royal straigh flush:** We have 4 possible events, one for each suit

$$P = \frac{n}{N} = \frac{4}{2598960} = 1.54 \cdot 10^{-6} \tag{4}$$

- **Four of a kind:** An ordering of 4 out of 5 cards, so we have more than one possible ordering for each four of a kind. First card can be picked in,

$$\binom{13}{1} = 13 \tag{5}$$

ways. Three of the next cards must be of same rank, with only

$$\binom{3}{3} = 1, \tag{6}$$

way. The last card can be any of the other cards, with

$$\binom{48}{1} = 48 \tag{7}$$

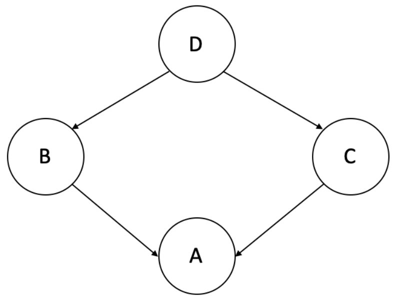
combinations. The result is then the number of possible events over all events,

$$P = \frac{n}{N} = \frac{\binom{13}{1}\binom{3}{3}\binom{48}{1}}{2598960} = \frac{624}{2598960} = 2.4 \cdot 10^{-4} \tag{8}$$

2 Bayesian Network Construction

1. $p(A, B, C, D) = p(A|B, C)p(B|D)p(C|D)p(D)$

$P(D)$	$P(\neg D)$
0.5	0.5



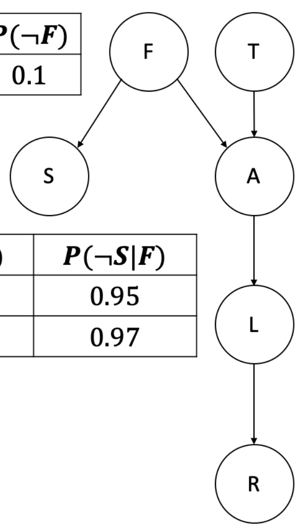
D	$P(B\|D)$	$P(\neg B\|D)$
T	0.3	0.7
F	0.2	0.8

D	$P(C\|D)$	$P(\neg C\|D)$
T	0.95	0.05
F	0.9	0.1

B	C	$P(A\|B,C)$	$P(\neg A\|B,C)$
T	T	0.03	0.97
T	F	0.2	0.8
F	T	0.6	0.4
F	F	0.8	0.2

2. $(F, T, A, S, L, R) = p(F)p(T)p(A|F, T)p(S|F)p(L|A)p(R|L)$

$P(F)$	$P(\neg F)$
0.9	0.1



$P(T)$	$P(\neg T)$
0.7	0.3

F	T	$P(A\|F,T)$	$P(\neg A\|F,T)$
T	T	0.99	0.01
T	F	0.51	0.49
F	T	0.51	0.49
F	F	0.01	0.99

F	$P(S\|F)$	$P(\neg S\|F)$
T	0.05	0.95
F	0.03	0.97

A	$P(L\|A)$	$P(\neg L\|A)$
T	1	0
F	0	1

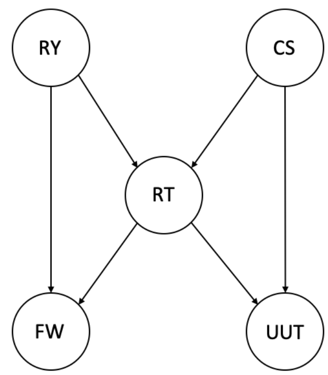
L	$P(R\|L)$	$P(\neg R\|L)$
T	0.1	0.9
F	0.15	0.85

3. $p(RainToday, RainYesterday, FloorWet, UseUmbrellaToday, CloudSky) = p(RainYesterday)p(CloudSky)p(RainToday|RainYesterday, CloudSky)p(FloorWet|RainToday, RainYesterday)$ $p(UseUmbrellaToday|RainToday, CloudSky)$

RT = RainToday, RY = RainYesterday, FW = FloorWet, UUT = UseUmbrellaToday, CS = CloudSky

$P(RY)$	$P(\neg RY)$
0.3	0.7

$P(CS)$	$P(\neg CS)$
0.5	0.5



RY	CS	$P(RT\|RY, CS)$	$P(\neg RT\|RY, CS)$
T	T	0.8	0.2
T	F	0.1	0.9
F	T	0.6	0.4
F	F	0.05	0.95

RY	RT	$P(FW\|RY, RT)$	$P(\neg FW\|RY, RT)$
T	T	0.99	0.01
T	F	0.6	0.4
F	T	0.9	0.1
F	F	0.05	0.95

CS	RT	$P(UUT\|CS, RT)$	$P(\neg UUT\|CS, RT)$
T	T	0.9	0.1
T	F	0.3	0.7
F	T	0.85	0.15
F	F	0.01	0.99

Explanation: Since all variables are binary, they are either true or false. The only constraint on the tables is that the total probability of a variable must equal 1, and since the variables in network 1 and 2 are ambiguous, the probabilities are given seemingly random values. This could also be done for network 3, but as the variables have describing names that relate to the weather, the probabilities should somewhat comply with what is expected.

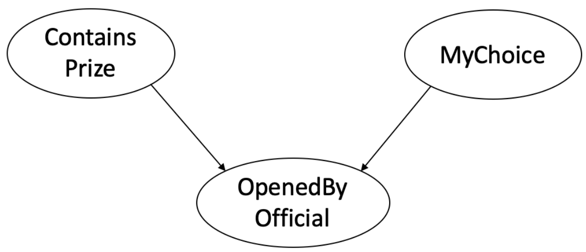
3 Bayesian Network Application

You are are confronted with three doors A, B, and C. Behind exactly one of the doors there is $10000. The money is yours if you choose the correct door. After you have made your first choice of door but still not opened it, an official comes in. He works according to some rules:

1. He starts by opening a door. He knows where the prize is, and he is not allowed to open that door. Furthermore, he cannot open the door you have chosen. Hence, he opens the door with nothing behind.
2. Now there are two closed doors, one of which contains the prize. The official will ask you if you want to alter your choice (i.e., to trade your door for the other one that is not open).

Should you do that? Yes.

	CP
$P(A)$	1/3
$P(B)$	1/3
$P(C)$	1/3



MC
A
B
C

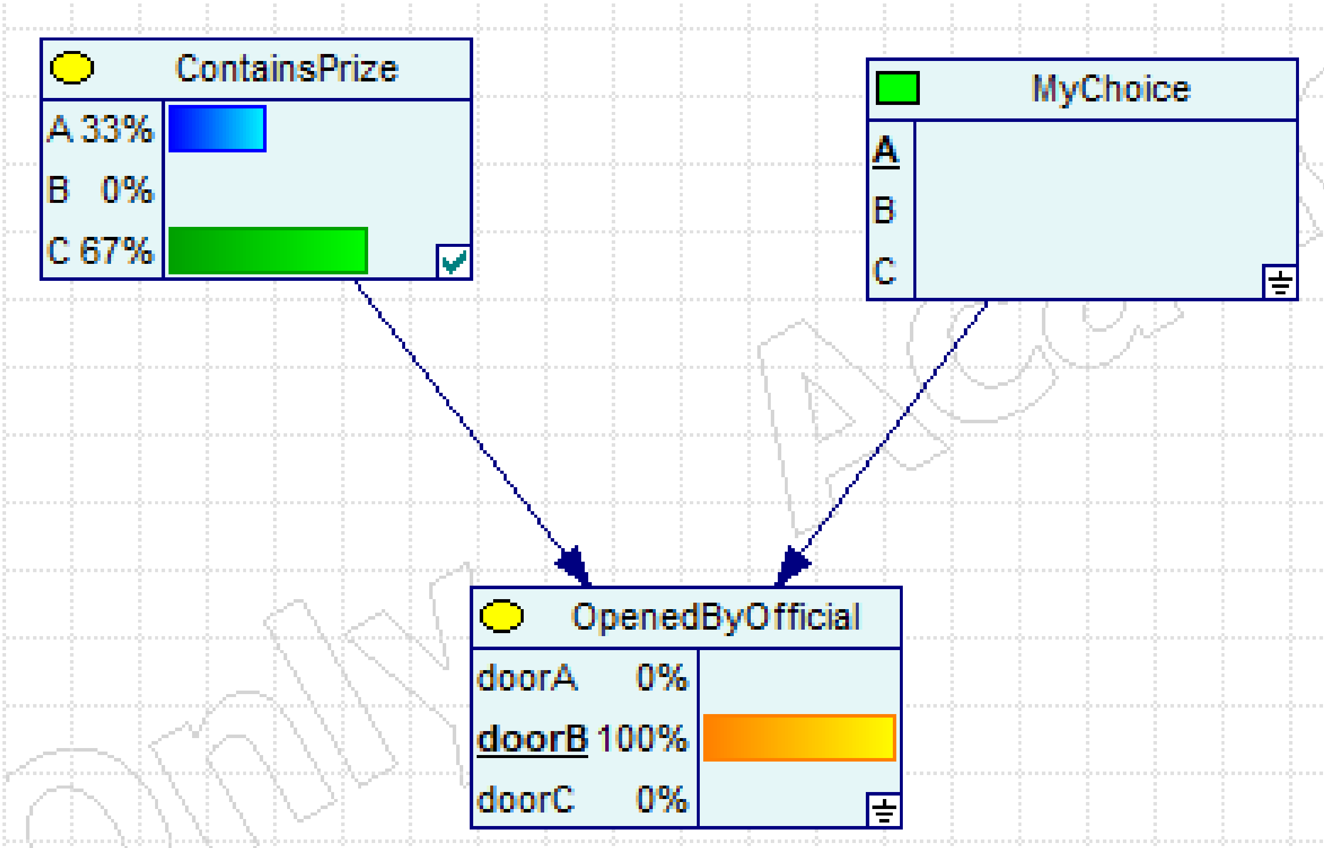
CP	A			B			C		
MC	A	B	C	A	B	C	A	B	C
$doorA$	0	0	0	0	0.5	1	0	1	0.5
$doorB$	0.5	0	1	0	0	0	1	0	0.5
$doorC$	0.5	1	0	1	0.5	0	0	0	0

The network can be constructed as above, with each of the three doors initially having equal probability of containing the prize. *MyChoice* is a decision block, and will not contain any probabilities as we decide which door to open. The door the official opens depends on our initial choice and which door contains the prize. After the official opens a door, the probability of each door containing a prize in *ContainsPrize* will change, as seen in the GeNIe network below. Here we have chosen to open door A, resulting in the official opening door B and the probability of door C containing the prize jumping up to 2/3. By studying the probability table for *OpenedByOfficial*, it is apparent that switching doors is the best choice.

Consider the case where we decide on door A.

- **Prize is in door A:** The official will open either door B or door C, and switching doors will result in you losing.
- **Prize is in door B:** The official will open door C, and switching to door B will result in you winning.
- **Prize is in door C:** The official will open door B, and switching to door C will result in you winning.

We can see that in 2/3 of the cases, switching doors will result in a win. An intuitive reason behind the result can more easily be seen if we increase the number of doors to a thousand doors. The official will then open all doors but two, the one containing the prize and the one you initially chose. It should then be obvious that switching the doors is the best choice.


ContainsPrize
A 33%
B 0%
C 67%
MyChoice
A
B
C
OpenedByOfficial
doorA 0%
doorB 100%
doorC 0%