Knowledge Representation

SEMESTER 1 2018 - Assignment 2

INSTRUCTOR: Dr Ruairí O'Reilly

ruairi.oreilly@cit.ie

Topics schedule

Week		
1	No lecture given	No lecture given
2	Intro to AI	Intelligent Agents
3	Logical Agents (i)	Logical Agents (ii)
4	Problem Solving & Search*	Problem Solving & Search
5	Visualisation with MatPlotLib (i)	Visualisation with MatPlotLib (i)
6	First Order Logic*	Inference in First-Order Logic
7	Knowledge Representation	Quantifying Uncertainty
8	Bayesian networks (i)	Bayesian networks (ii)
9	Inference in Bayesian networks	
10	Inference in Bayesian networks	
11	Temporal Probability Models	
12	TBD	
13	In Class Assignment	
14	Assignment 2 due - 21/12/18 @ 23:59	

* Note: Delivery differs slightly between Full-Time and Part-Time

Completion Date

Submission date is based on your class group Full-time Day or Part-time Night:

Full-time Day - Friday the 21st of December @ 23:59

Part-time Night - Friday the 21st of December @ 23:59

Probability & Bayesian Networks

Expected submission details:

- A single file containing your work. You must submit the solution as a Jupyter notebook file (.ipynb).

- All dependencies are assumed to be accessible locally in the same folder. (Any issues with path resolution will result in loss of marks)

- Each question is expected to have one or more functions associated with it, see the bottom of the assignment for the expected format for these functions on Canvas/Blackboard.

- Articulating your understanding of the underlying material is important, make good use of comments in code and text in the .ipynb file.

i) Probability Distribution - basics

A) You have an unbiased six-sided dice *a*. The die is rolled twice to generate the outcomes X1 and X2. Using the code made available from the AIMA data repo, calculate the probability of generating SnakeEyes (1,1 - each 1 is rolled in succession rather than two dice together) and print out the probability:

Expected output: "Probability of Snake Eyes is ***X***" where X is the probability

ii) Constructing a Bayesian Network

Construct a Bayes net using the BayesNet class for the following scenario:

You have a daily commute to work, a number of considerations that can affect your commute. You also have a temperamental boss, if your late he/she will typically berate you over the phone which will leave you feeling dejected for the day. Sometimes you use the Motorway to make up time and avoid being late

Variables: Traffic (T), Rain (R), Motorway (M), Late (L), BossCalls (B)

Network Topology:

- Sometimes you decide to take the Motorway
- Rain can result in you being late
- Traffic can result in you being late.
- Being late can cause your boss to call.

P (R)
.41

P(T)
.15

P(M)
.01

L	P(B|L)
T	.8
F	.1

R	T	M	P(L|R,T,M)
T	T	T	.80
T	T	F	.98
T	F	T	.2
T	F	F	.3
F	T	T	.25
F	T	F	.24
F	F	T	.001
F	F	F	.05

Qi) Draw the Bayesian Network

Qii) Using the BayesNode code from the AIMA repository create a Bayesian Network (BN) based on this scenario.

Qiii) Write a query to output the CPT for the "Late" node.

Qiii) Using the BN from Qii, write the python query to answer the following queries:

a) You took the Motorway
b) The boss does not call given that you are late
c) You are late when its raining & there is traffic as you took the Motorway

iii) Exact Inference in Bayesian Networks

A - Inference by Enumeration

Qi) Implement the following queries:

a) It is raining when the Boss calls **X**% of the time
   It is not raining when the Boss calls **X**% of the time
b) There is traffic when the Boss calls around **X**% of the time.
   There is no traffic when the Boss calls **X**% of the time
c) I am using the Motorway when the Boss calls around **X**% of the time
   I am not using the Motorway when the Boss calls around **X**% of the time

d) The Boss calls when it is raining and there is Traffic around **X**% of the time
   The Boss does not call when it is raining and there is Traffic around **X**% of the time

Qii) Explain how inference by enumeration works? Particularly in relation to your answer for the prior question (iii - Qi)

Marking rubric - TBC

Qts	1h1 (>70%)	2H1 (60-69%)	Pass (40-59%)	Fail (<40%)
i - X%				
ii - X%				
iii - X%				
Doc req	Excellent understanding demonstrated via	Good understanding demonstrated via	Minimal understanding demonstrated, acceptable	Limited understanding demonstrated, poor

	documentation and articulation of underlying concepts.	documentation and articulation of underlying concepts.	documentation and limited articulation of underlying concepts.	documentation and little/no articulation of underlying concepts.

Code format

This is to simplify the correction enabling me to call functions containing your solutions from a singular location and checking the output rather than running all lines procedurally in the .ipynb file.

```
from probability import *
from utils import print_table
from notebook import psource, pseudocode, heatmap

# Complete solution to i) Probability Distribution - basics
def qOne():

# Part of solution to ii) Constructing a Bayesian Network - it is recommended
# you use a BN() function to return your Bayesian Network.
def qTwo(bn):

# Complete solution to iii) Exact Inference in Bayesian Networks
def qThree(bn):

# Complete solution to i) Probability Distribution - basics
def BN():

beliefNetwork = BN()

qOne()
qTwo(beliefNetwork)
qThree(beliefNetwork)
```

```
from probability import *
from utils import print_table
from notebook import psource, pseudocode, heatmap

# Complete solution to i) Probability Distribution - basics
def qOne():

# Part of solution to ii) Constructing a Bayesian Network - it is recommended you use a BN()
function to return your
# Bayesian Network.
def qTwo(bn):

# Complete solution to iii) Exact Inference in Bayesian Networks
def qThree(bn):

# Complete solution to i) Probability Distribution - basics
def BN():

beliefNetwork = BN()

qOne()
qTwo(beliefNetwork)
qThree(beliefNetwork)
```