

```
#### Name: William Pulkownik
#### Date: 1MAY2020
#### Course: SP20-CPSC-47000-001
#### Assianment: MP4.py
```

```
In [1]: import pyAgrum as gum
import pyAgrum.lib.notebook as gnb
import os
```

Step 2 - Create an empty BayesNet with a specified name

```
In [2]: bn=gum.fastBN("fraud->jewelry:fraud->gas:age[3]->jewelry<-sex:")
```

```
In [16]: #Fraud Conditional Probability Table
bn.cpt('fraud').fillWith([.99,.01])
#Age CPT
bn.cpt('age').fillWith([.25,.4,.35] )
#Sex CPT
bn.cpt('sex').fillWith([.5,.5])
#Gas CPT
bn.cpt('gas')[{'fraud':0}] = [.99,.01]
bn.cpt('gas')[{'fraud':1}] = [.8,.2]
#Jewelry CPT
bn.cpt('jewelry')[{'age': 0, 'sex': 0, 'fraud': 0}] = [0.9999, 0.0001]
bn.cpt('jewelry')[{'age': 0, 'sex': 0, 'fraud': 1}] = [0.95, 0.05]
bn.cpt('jewelry')[{'age': 0, 'sex': 1, 'fraud': 0}] = [0.9995, 0.0005]
bn.cpt('jewelry')[{'age': 0, 'sex': 1, 'fraud': 1}] = [0.95, 0.05]

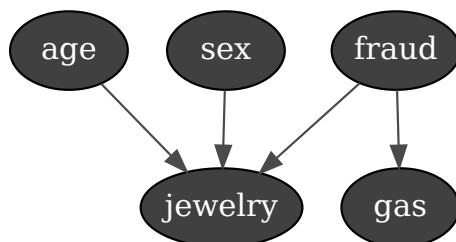
bn.cpt('jewelry')[{'age': 1, 'sex': 0, 'fraud': 0}] = [0.9996, 0.0004]
bn.cpt('jewelry')[{'age': 1, 'sex': 0, 'fraud': 1}] = [0.95, 0.05]
bn.cpt('jewelry')[{'age': 1, 'sex': 1, 'fraud': 0}] = [0.998, 0.002]
bn.cpt('jewelry')[{'age': 1, 'sex': 1, 'fraud': 1}] = [0.95, 0.05]

bn.cpt('jewelry')[{'age': 2, 'sex': 0, 'fraud': 0}] = [0.9998, 0.0002]
bn.cpt('jewelry')[{'age': 2, 'sex': 0, 'fraud': 1}] = [0.95, 0.05]
bn.cpt('jewelry')[{'age': 2, 'sex': 1, 'fraud': 0}] = [0.999, 0.001]
bn.cpt('jewelry')[{'age': 2, 'sex': 1, 'fraud': 1}] = [0.95, 0.05]

#bn.cpt('')[{'':}] = []
```

```
In [28]: bn
```

```
Out[28]:
```



B) list the variables, max and minimize possibility of fraud, list Probabilities

In [15]: `bn.cpt('jewelry').var_names`

Out[15]: ['sex', 'age', 'fraud', 'jewelry']

maximize fraud

In [26]: `ie.setEvidence({'age':0, 'sex':0, 'gas':1, 'jewelry': 1}) # hard evidence
ie.makeInference()
ie.posterior('fraud')`

Out[26]:

fraud	
0	1
0.0098	0.9902

minimize fraud

In [27]: `ie.setEvidence({'age':2, 'sex':1, 'gas':0, 'jewelry': 0}) # hard evidence
ie.makeInference()
ie.posterior('fraud')`

Out[27]:

fraud	
0	1
0.9923	0.0077

C) Compute the probabilities

In [19]: `ie=aum.LazyPropagation(bn)`

`P(Fraud|gas,jewelry)`

In [21]: `ie.setEvidence({'gas':1, 'jewelry': 1}) # hard evidence
ie.makeInference()
ie.posterior('fraud')`

Out[21]:

fraud	
0	1
0.0704	0.9296

`P(Fraud|gas,jewelry)`

In [22]: `ie.setEvidence({'gas':1, 'jewelry': 0}) # hard evidence
ie.makeInference()
ie.posterior('fraud')`

Out[22]:

fraud	
0	1
0.8389	0.1611

`P(Fraud|-gas,jewelry)`

```
In [23]: ie.setEvidence({'gas':0, 'jewelry': 1}) # hard evidence
         ie.makeInference()
         ie.posterior('fraud')
```

Out[23]:

fraud	
0	1
0.6521	0.3479

P(Fraud|-gas,-jewelry)

```
In [24]: ie.setEvidence({'gas':1, 'jewelry': 0}) # hard evidence
         ie.makeInference()
         ie.posterior('fraud')
```

Out[24]:

fraud	
0	1
0.8389	0.1611