E08 Bayesian Network

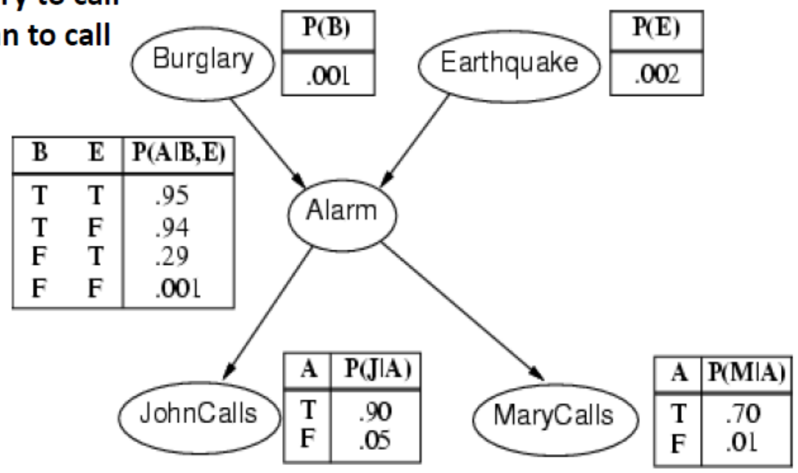
姓名	学号	日期
张天祎	18340215	2020.10.28

1 Tasks

1.1 Burglary

- **A burglary can set the alarm off**
- **An earthquake can set the alarm off**
- **The alarm can cause Mary to call**
- **The alarm can cause John to call**

Note that these tables only provide the probability that Xi is true.
(E.g., Pr(A is true|B,E))
The probability that Xi is false is 1- these values



P(B)
.001

P(E)
.002

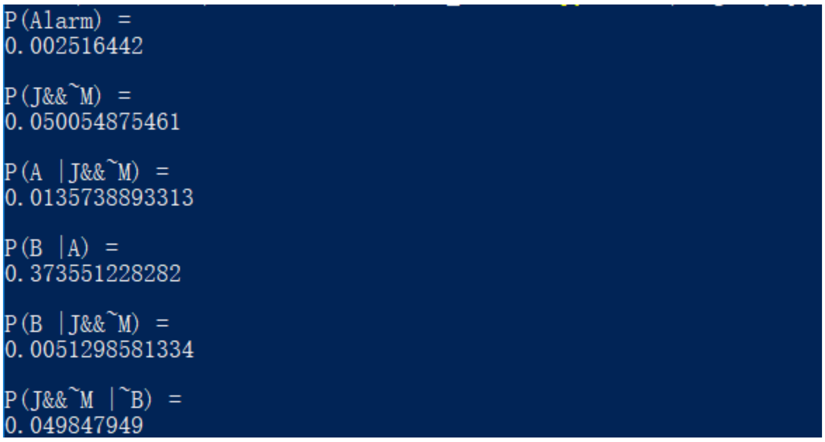
B	E	P(A|B,E)
T	T	.95
T	F	.94
F	T	.29
F	F	.001

A	P(J|A)
T	.90
F	.05

A	P(M|A)
T	.70
F	.01

Please code to calculate:

1. P(A)
2. P(J$\bar{M}$)
3. P(A|J$\bar{M}$)
4. P(B|A)
5. P(B|J$\bar{M}$)
6. P(J$\bar{M}$|$\bar{B}$)



```
P(Alarm) =
0.002516442

P(J&&~M) =
0.050054875461

P(A |J&&~M) =
0.0135738893313

P(B |A) =
0.373551228282

P(B |J&&~M) =
0.0051298581334

P(J&&~M |~B) =
0.049847949
```

1.2 Diagnosing

Variables and their domais

(1)PatientAge:['0-30','31-65','65+'] (2)CTScanResult:['Ischemic Stroke','Hemmorraghic Stroke'] (3)MRIScanResult: ['Ischemic Stroke','Hemmorraghic Stroke'] (4)StrokeType: ['Ischemic Stroke','Hemmorraghic Stroke', 'Stroke Mimic'] (5)Anticoagulants: ['Used','Not used'] (6)Mortality: ['True', 'False'] (7)Disability: ['Negligible', 'Moderate', 'Severe']

CPTs

- Note: [CTScanResult,MRIScanResult,StrokeType] means:P(StrokeType='...' | CTScanResult='...'^MRIScanResult='...')

(1) [PatientAge]

['0-30', 0.10], ['31-65', 0.30], ['65+', 0.60]

(2) [CTScanResult]

['Ischemic Stroke',0.7], ['Hemmorraghic Stroke',0.3]

(3) [MRIScanResult]

['Ischemic Stroke',0.7], ['Hemmorraghic Stroke',0.3]

(4) [Anticoagulants]

[Used',0.5], ['Not used',0.5]

(5) [CTScanResult, MRIScanResult,StrokeType])

['Ischemic Stroke','Ischemic Stroke','Ischemic Stroke',0.8], ['Ischemic Stroke','Hemmorraghic Stroke','Ischemic Stroke',0.5],
['Hemmorraghic Stroke','Ischemic Stroke','Ischemic Stroke',0.5], ['Hemmorraghic Stroke','Hemmorraghic Stroke','Ischemic Stroke',0],

['Ischemic Stroke','Ischemic Stroke','Hemmorraghic Stroke',0], ['Ischemic Stroke','Hemmorraghic Stroke','Hemmorraghic Stroke',0.4], ['Hemmorraghic Stroke','Ischemic Stroke','Hemmorraghic Stroke',0.4], ['Hemmorraghic Stroke','Hemmorraghic Stroke','Hemmorraghic Stroke',0.9],

['Ischemic Stroke','Ischemic Stroke','Stroke Mimic',0.2], ['Ischemic Stroke','Hemmorraghic Stroke','Stroke Mimic',0.1],
['Hemmorraghic Stroke','Ischemic Stroke','Stroke Mimic',0.1], ['Hemmorraghic Stroke','Hemmorraghic Stroke','Stroke Mimic',0.1],

(6) [StrokeType, Anticoagulants, Mortality]

['Ischemic Stroke', 'Used', 'False',0.28], ['Hemmorraghic Stroke', 'Used', 'False',0.99], ['Stroke Mimic', 'Used', 'False',0.1], ['Ischemic Stroke','Not used', 'False',0.56], ['Hemmorraghic Stroke', 'Not used', 'False',0.58], ['Stroke Mimic', 'Not used', 'False',0.05],

['Ischemic Stroke', 'Used' ,'True',0.72], ['Hemmorraghic Stroke', 'Used', 'True',0.01], ['Stroke Mimic', 'Used', 'True',0.9], ['Ischemic Stroke', 'Not used' ,'True',0.44], ['Hemmorraghic Stroke', 'Not used', 'True',0.42], ['Stroke Mimic', 'Not used', 'True',0.95]

(7) [StrokeType, PatientAge, Disability]

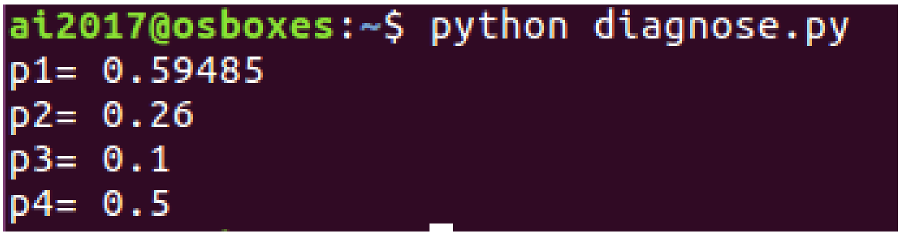
['Ischemic Stroke', '0-30','Negligible', 0.80], ['Hemmorraghic Stroke', '0-30','Negligible', 0.70], ['Stroke Mimic', '0-30', 'Negligible',0.9], ['Ischemic Stroke', '31-65','Negligible', 0.60], ['Hemmorraghic Stroke', '31-65','Negligible', 0.50], ['Stroke Mimic', '31-65', 'Negligible',0.4], ['Ischemic Stroke', '65+' , 'Negligible',0.30], ['Hemmorraghic Stroke', '65+' , 'Negligible',0.20], ['Stroke Mimic', '65+' , 'Negligible',0.1],

['Ischemic Stroke', '0-30' ,'Moderate',0.1], ['Hemmorraghic Stroke', '0-30' ,'Moderate',0.2], ['Stroke Mimic', '0-30' ,'Moderate',0.05], ['Ischemic Stroke', '31-65','Moderate',0.3], ['Hemmorraghic Stroke', '31-65','Moderate',0.4], ['Stroke Mimic', '31-65','Moderate',0.3], ['Ischemic Stroke', '65+' ,'Moderate',0.4], ['Hemmorraghic Stroke', '65+' ,'Moderate',0.2], ['Stroke Mimic', '65+' ,'Moderate',0.1],

['Ischemic Stroke', '0-30' ,'Severe',0.1], ['Hemmorraghic Stroke', '0-30' ,'Severe',0.1], ['Stroke Mimic', '0-30' ,'Severe',0.05], ['Ischemic Stroke', '31-65','Severe',0.1], ['Hemmorraghic Stroke', '31-65','Severe',0.1], ['Stroke Mimic', '31-65','Severe',0.3], ['Ischemic Stroke', '65+' ,'Severe',0.3], ['Hemmorraghic Stroke', '65+' ,'Severe',0.6], ['Stroke Mimic', '65+' ,'Severe',0.8]

Calculation

Please code to calculate the following probability value:
p1 = P(Mortality='True'|PatientAge='31-65'^CTScanResult='Ischemic Stroke')
p2 = P(Disability='Moderate'|PatientAge='65+'^MRIScanResult='Hemmorraghic Stroke')
p3 = P(StrokeType='Stroke Mimic'|PatientAge='65+'^CTScanResult='Hemmorraghic Stroke'^MRIScanResult='Ischemic Stroke')
p4 = P(Anticoagulants='Not used'|PatientAge='0-30')



```
ai2017@osboxes:~$ python diagnose.py
p1= 0.59485
p2= 0.26
p3= 0.1
p4= 0.5
```

2 Notes

Please send E08 YourNumber.zip which should contain the codes(ai_2020@foxmail.com) .

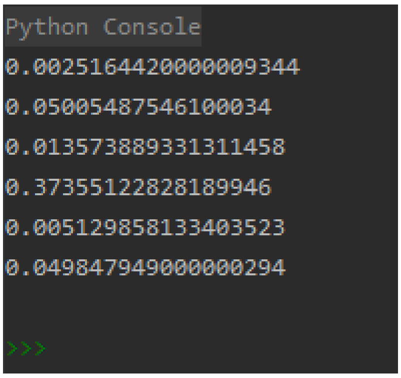
3 Codes and Results

```
import math
from pomegranate import *

# 1
Burglary = DiscreteDistribution({'True': 0.001, 'False': 0.999})
Earthquake = DiscreteDistribution({'True': 0.002, 'False': 0.998})

Alarm = ConditionalProbabilityTable(
    [['True', 'True', 'True', 0.95],
     ['True', 'True', 'False', 0.05],
     ['True', 'False', 'True', 0.94],
     ['True', 'False', 'False', 0.06],
     ['False', 'True', 'True', 0.29],
     ['False', 'True', 'False', 0.71],
     ['False', 'False', 'True', 0.001],
     ['False', 'False', 'False', 0.999]],
    [Burglary, Earthquake]
)

JohnCalls = ConditionalProbabilityTable(
    [['True', 'True', 0.90],
     ['True', 'False', 0.10],
     ['False', 'True', 0.05],
     ['False', 'False', 0.95]],
    [Alarm]
)

MaryCalls = ConditionalProbabilityTable(
    [['True', 'True', 0.70],
     ['True', 'False', 0.30],
     ['False', 'True', 0.01],
     ['False', 'False', 0.99]],
    [Alarm]
)

s1 = State(Burglary, name='Burglary')
s2 = State(Earthquake, name='Earthquake')
s3 = State(Alarm, name='Alarm')
s4 = State(JohnCalls, name='JohnCalls')
s5 = State(MaryCalls, name='MaryCalls')

network = BayesianNetwork("1")
network.add_states(s1, s2, s3, s4, s5)
network.add_transition(s1, s3)
network.add_transition(s2, s3)
network.add_transition(s3, s4)
network.add_transition(s3, s5)
network.bake()

print(network.predict_proba({})[2].parameters[0]['True'])
print(network.predict_proba({})[3].parameters[0]
['True']*network.predict_proba({'JohnCalls': 'True'})[4].parameters[0]['False'])
print(network.predict_proba({'JohnCalls': 'True', 'MaryCalls': 'False'})
[2].parameters[0]['True'])
print(network.predict_proba({'Alarm': 'True'})[0].parameters[0]['True'])
print(network.predict_proba({'JohnCalls': 'True', 'MaryCalls': 'False'})
[0].parameters[0]['True'])
print(network.predict_proba({'Burglary': 'False'})[3].parameters[0]['True']*
      network.predict_proba({'Burglary': 'False', 'JohnCalls': 'True'})
[4].parameters[0]['False'])
```

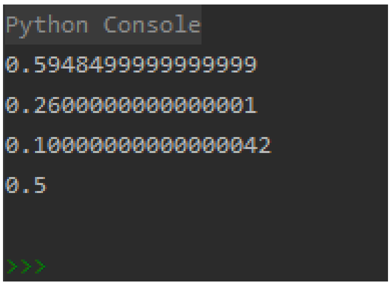
```
import math
from pomegranate import *

# 2
PatientAge = DiscreteDistribution({'0-30': 0.10, '31-65': 0.30, '65+': 0.60})
CTScanResult = DiscreteDistribution({'Ischemic Stroke': 0.7, 'Hemmorraghic
Stroke': 0.3})
MRIScanResult = DiscreteDistribution({'Ischemic Stroke': 0.7, 'Hemmorraghic
Stroke': 0.3})
Anticoagulants = DiscreteDistribution({'Used': 0.5, 'Not used': 0.5})
```

```
StrokeType = ConditionalProbabilityTable(
    [['Ischemic Stroke', 'Ischemic Stroke', 'Ischemic Stroke', 0.8],
    ['Ischemic Stroke', 'Hemmorraghic Stroke', 'Ischemic Stroke', 0.5],
    ['Hemmorraghic Stroke', 'Ischemic Stroke', 'Ischemic Stroke', 0.5],
    ['Hemmorraghic Stroke', 'Hemmorraghic Stroke', 'Ischemic Stroke', 0],
    ['Ischemic Stroke', 'Ischemic Stroke', 'Hemmorraghic Stroke', 0],
    ['Ischemic Stroke', 'Hemmorraghic Stroke', 'Hemmorraghic Stroke', 0.4],
    ['Hemmorraghic Stroke', 'Ischemic Stroke', 'Hemmorraghic Stroke', 0.4],
    ['Hemmorraghic Stroke', 'Hemmorraghic Stroke', 'Hemmorraghic Stroke', 0.9],
    ['Ischemic Stroke', 'Ischemic Stroke', 'Stroke Mimic', 0.2],
    ['Ischemic Stroke', 'Hemmorraghic Stroke', 'Stroke Mimic', 0.1],
    ['Hemmorraghic Stroke', 'Ischemic Stroke', 'Stroke Mimic', 0.1],
    ['Hemmorraghic Stroke', 'Hemmorraghic Stroke', 'Stroke Mimic', 0.1]],
    [CTScanResult, MRIScanResult]
)

Mortality = ConditionalProbabilityTable(
    [['Ischemic Stroke', 'Used', 'False',0.28],
    ['Hemmorraghic Stroke', 'Used', 'False',0.99],
    ['Stroke Mimic', 'Used', 'False',0.1],
    ['Ischemic Stroke','Not used', 'False',0.56],
    ['Hemmorraghic Stroke', 'Not used', 'False',0.58],
    ['Stroke Mimic', 'Not used', 'False',0.05],
    ['Ischemic Stroke',  'Used' ,'True',0.72],
    ['Hemmorraghic Stroke', 'Used', 'True',0.01],
    ['Stroke Mimic', 'Used', 'True',0.9],
    ['Ischemic Stroke',  'Not used' ,'True',0.44],
    ['Hemmorraghic Stroke', 'Not used', 'True',0.42 ],
    ['Stroke Mimic', 'Not used', 'True',0.95]],
    [StrokeType, Anticoagulants]
)

Disability = ConditionalProbabilityTable(
    [['Ischemic Stroke',   '0-30','Negligible', 0.80],
    ['Hemmorraghic Stroke', '0-30','Negligible', 0.70],
    ['Stroke Mimic',        '0-30', 'Negligible',0.9],
    ['Ischemic Stroke',     '31-65','Negligible', 0.60],
    ['Hemmorraghic Stroke', '31-65','Negligible', 0.50],
    ['Stroke Mimic',        '31-65', 'Negligible',0.4],
    ['Ischemic Stroke',     '65+'  , 'Negligible',0.30],
    ['Hemmorraghic Stroke', '65+'  , 'Negligible',0.20],
    ['Stroke Mimic',        '65+'  , 'Negligible',0.1],
    ['Ischemic Stroke',     '0-30' ,'Moderate',0.1],
    ['Hemmorraghic Stroke', '0-30' ,'Moderate',0.2],
    ['Stroke Mimic',        '0-30' ,'Moderate',0.05],
    ['Ischemic Stroke',     '31-65','Moderate',0.3],
    ['Hemmorraghic Stroke', '31-65','Moderate',0.4],
    ['Stroke Mimic',        '31-65','Moderate',0.3],
    ['Ischemic Stroke',     '65+'  ,'Moderate',0.4],
    ['Hemmorraghic Stroke', '65+'  ,'Moderate',0.2],
    ['Stroke Mimic',        '65+'  ,'Moderate',0.1],
    ['Ischemic Stroke',     '0-30' ,'Severe',0.1],
    ['Hemmorraghic Stroke', '0-30' ,'Severe',0.1],
    ['Stroke Mimic',        '0-30' ,'Severe',0.05],
    ['Ischemic Stroke',     '31-65','Severe',0.1],
    ['Hemmorraghic Stroke', '31-65','Severe',0.1],
    ['Stroke Mimic',        '31-65','Severe',0.3],
    ['Ischemic Stroke',     '65+'  ,'Severe',0.3],
    ['Hemmorraghic Stroke', '65+'  ,'Severe',0.6],
    ['Stroke Mimic',        '65+'  ,'Severe',0.8]],
    [StrokeType, PatientAge]
)

s1 = State(PatientAge, name='PatientAge')
s2 = State(CTScanResult, name='CTScanResult')
s3 = State(MRIScanResult, name='MRIScanResult')
s4 = State(Anticoagulants, name='Anticoagulants')
s5 = State(StrokeType, name='StrokeType')
s6 = State(Mortality, name='Mortality')
s7 = State(Disability, name='Disability')

network = BayesianNetwork("2")
network.add_states(s1, s2, s3, s4, s5, s6, s7)
network.add_transition(s2, s5) # 有向图，不能写反
network.add_transition(s3, s5)
network.add_transition(s5, s6)
network.add_transition(s4, s6)
network.add_transition(s5, s7)
network.add_transition(s1, s7)
network.bake()
print(network.predict_proba({'PatientAge': '31-65', 'CTScanResult': 'Ischemic Stroke'})[5].parameters[0]['True'])
print(network.predict_proba({'PatientAge': '65+', 'MRIScanResult': 'Hemmorraghic Stroke'})[6].parameters[0]['Moderate'])
print(network.predict_proba({'PatientAge': '65+', 'CTScanResult': 'Hemmorraghic Stroke', 'MRIScanResult': 'Ischemic Stroke'})[4].parameters[0]['Stroke Mimic'])
print(network.predict_proba({'PatientAge': '0-30'})[3].parameters[0]['Not used'])
```

两问结果分别如下：



```
Python Console
0.0025164420000009344
0.050054875461000034
0.013573889331311458
0.37355122828189946
0.005129858133403523
0.049847949000000294


>>>
```



```
Python Console
0.5948499999999999
0.2600000000000001
0.10000000000000042
0.5

>>>
```

正确无误。