School of Data and Computer Science, Sun Yat-sen University

E09 Bayesian Network

17341068 Yangfan Jiang

November 7, 2019

Contents

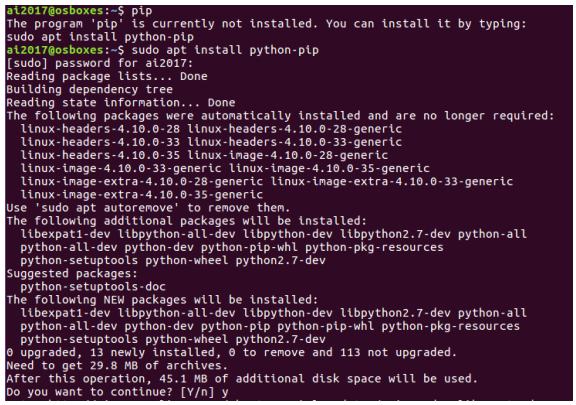
1 Pomegranate Installation

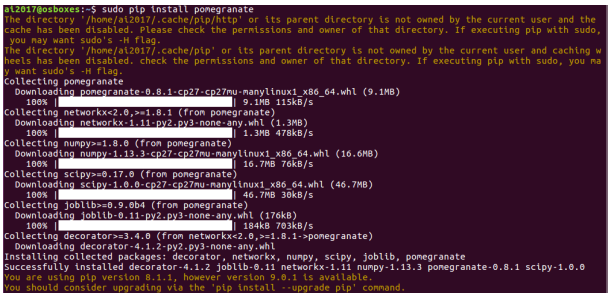
Under Linux:

1. Install `python` first (**python 2**, not python 3).

2. Run `sudo apt-get install python-pip` to install `pip`.

3. Run `sudo pip install pomegranate` to install `pomegranate`.

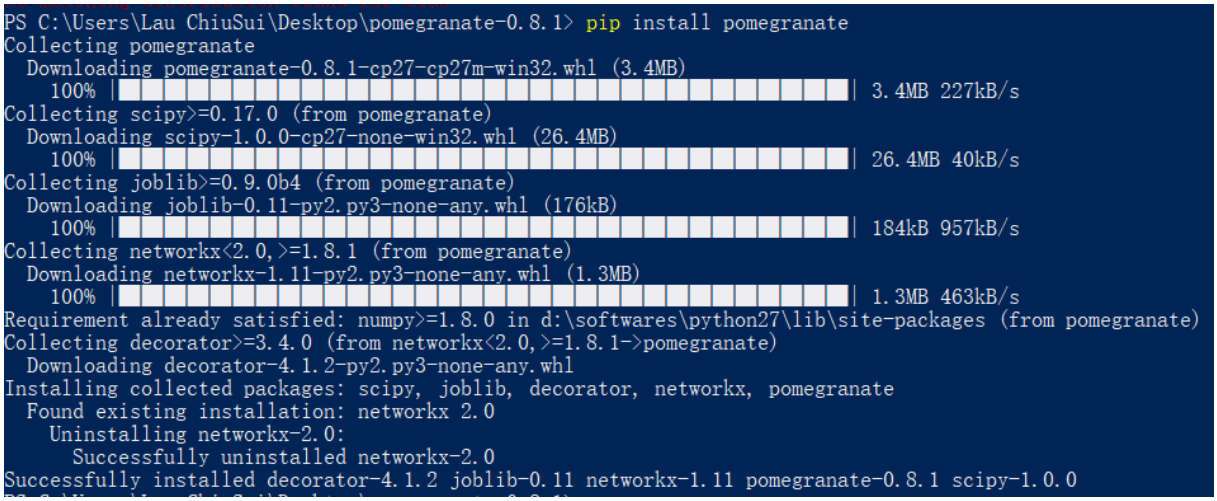




Under Windows

You can also run `pip install pomegranate` if you have installed `pip`. If you don't know how to install `pip`, please click `https://jingyan.baidu.com/article/e73e26c0d94e0524adb6a7ff.html`.

For more, please click the homepage of Pomegranate - `https://github.com/jmschrei/pomegranate` for help.



2 Building Bayesian Network

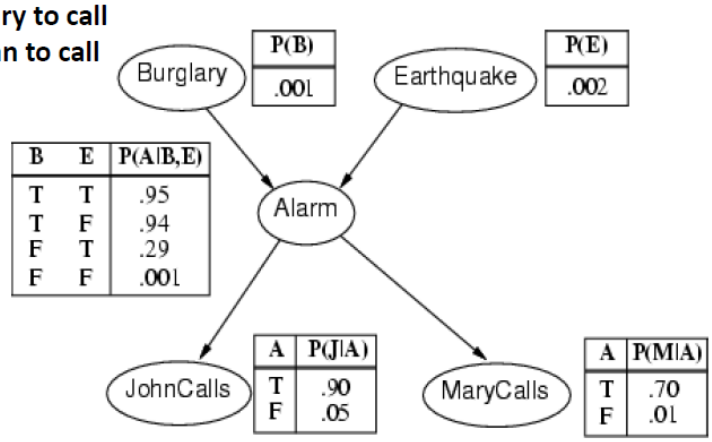
Please refer to `Tutorial_4_Bayesian_Networks.pdf`. I will explain it in class.

3 Tasks

3.1 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



Please code to calculate:

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

```
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
```

3.2 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
```

Please solve the 2 tasks and hand in a file named E09_YourNumber.pdf, and send it to ai_201901@foxmail.com

4 Codes and Results

4.1 Burglary

For problem 6, using the bayesian formula, we have

$$P(J\overline{M}|\overline{B}) = \frac{P(\overline{B}|J\overline{M}) \cdot P(J\overline{M})}{P(\overline{B})}$$

4.1.1 Codes

Initialization

```
from pomegranate import *

Burglary = DiscreteDistribution( {'T':.001, 'F':.999} )
```

```
Earthquake = DiscreteDistribution( {'T':.002, 'F':.998} )

Alarm = ConditionalProbabilityTable(
        [['T','T','T',.95],
         ['T','T','F',.05],
         ['T','F','T',.94],
         ['T','F','F',.06],
         ['F','T','T',.29],
         ['F','T','F',.71],
         ['F','F','T',.001],
         ['F','F','F',.999]], [Burglary, Earthquake] )

JohnCalls = ConditionalProbabilityTable(
            [['T','T',.90],
             ['T','F',.10],
             ['F','T',.05],
             ['F','F',.95]], [Alarm] )

MaryCalls = ConditionalProbabilityTable(
            [['T','T',.70],
             ['T','F',.30],
             ['F','T',.01],
             ['F','F',.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')

model = BayesianNetwork("Burglary Problem")

model.add_states(s1, s2, s3, s4, s5)
```

```
37
38 model.add_transition(s1, s3)
39 model.add_transition(s2, s3)
40 model.add_transition(s3, s4)
41 model.add_transition(s3, s5)
42
43 model.bake()
```

Solution

```
1  # 1.P(A)
2  marginals = model.predict_proba({})
3  P1 = marginals[2].parameters[0]['T']
4  print('P(Alarm) =', P1)
5
6  # 2.P(JM') = P(J|M')*P(M')
7  P_NOT_M = marginals[4].parameters[0]['F']
8  P_J_under_NOT_M = model.predict_proba({'MaryCalls':'F'})[3].
9                                                  parameters[0]['T']
10 P2 = P_J_under_NOT_M*P_NOT_M
11 print('P(J&&~M) =', P2)
12
13 # 3.P(A|J&&~M)
14 P3 = model.predict_proba({'JohnCalls':'T', 'MaryCalls':'F'})[2].
15                                                  parameters[0]['T']
16 print('P(A|J&&~M) =', P3)
17
18 # 4.P(B|A)
19 P4 = model.predict_proba({'Alarm':'T'})[0].parameters[0]['T']
20 print('P(B|A) =', P4)
21
22 # 5.P(B|J&&~M)
23 P5 = model.predict_proba({'JohnCalls':'T', 'MaryCalls':'F'})[0].
24                                                  parameters[0]['T']
```

```
print('P(B|J&&~M) =', P5)

# 6.P(J&&~M|~B)
P_b = marginals[0].parameters[0]['F']
P6 = (1-P5)*P2/P_b
print('P(J&&~M|~B) =', P6)
```

4.1.2 Result

```
P(Alarm) = 0.002516442000000935
P(J&&~M) = 0.050054875461000355
P(A|J&&~M) = 0.01357388933131146
P(B|A) = 0.3735512282818995
P(B|J&&~M) = 0.005129858133403528
P(J&&~M|~B) = 0.049847949000000266
```

Figure 1: Burglary Result

4.2 Diagnosing

4.2.1 Code

Inititlization

```
from pomegranate import *

PatientAge = DiscreteDistribution( {'0-30':.10,
                                    '31-65':.30,
                                    '65+':.60} )
CTScanResult = DiscreteDistribution( {'Ischemic Stroke':.70,
                                      'Hemmorraghic Stroke':.30} )
MRIScanResult = DiscreteDistribution( {'Ischemic Stroke':.70,
                                       'Hemmorraghic Stroke':.30} )
Anticoagulants = DiscreteDistribution( {'Used':.50, 'Not used':.50} )

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')

model = BayesianNetwork("Diagnosing")

model.add_states(s1, s2, s3, s4, s5, s6, s7)

model.add_transition(s2, s5)
model.add_transition(s3, s5)
model.add_transition(s4, s6)
model.add_transition(s5, s6)
model.add_transition(s1, s7)
model.add_transition(s5, s7)

model.bake()
```

Solution

```
P1 = model.predict_proba({'PatientAge':'31–65',
                          'CTScanResult':'Ischemic Stroke'})[5].
                          parameters[0]['True']

P2 = model.predict_proba({'PatientAge':'65+',
                          'MRIScanResult':'Hemmorraghic Stroke'})[6].
                          parameters[0]['Moderate']

P3 = model.predict_proba({'PatientAge':'65+',
                          'CTScanResult':'Hemmorraghic Stroke',
                          'MRIScanResult':'Ischemic Stroke'})[4].
                          parameters[0]['Stroke Mimic']

P4 = model.predict_proba({'PatientAge':'0–30'})[3].
```

```
15                                    parameters[0]['Not used']
```

4.2.2 Result

```
P1 = 0.5948499999999999
P2 = 0.2600000000000001
P3 = 0.10000000000000044
P4 = 0.5
```

Figure 2: Diagnosing Result