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

E08 BN

18340052 何泽

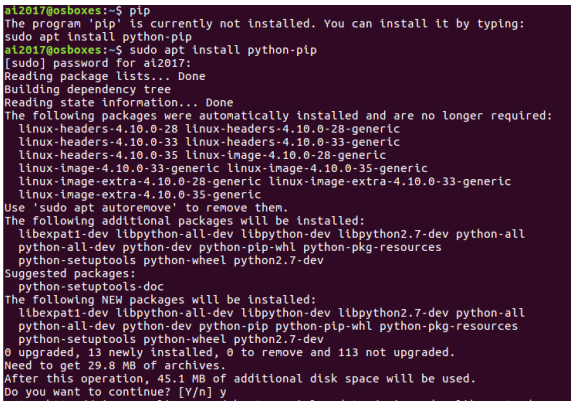
2020 年 10 月 29 日

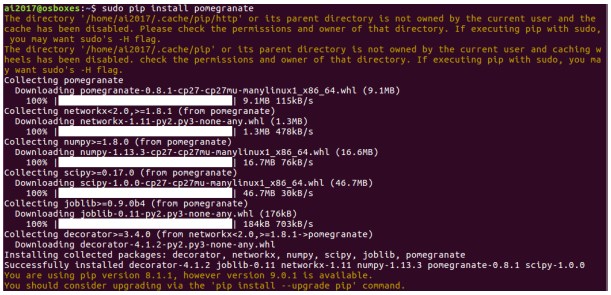
目录

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`.

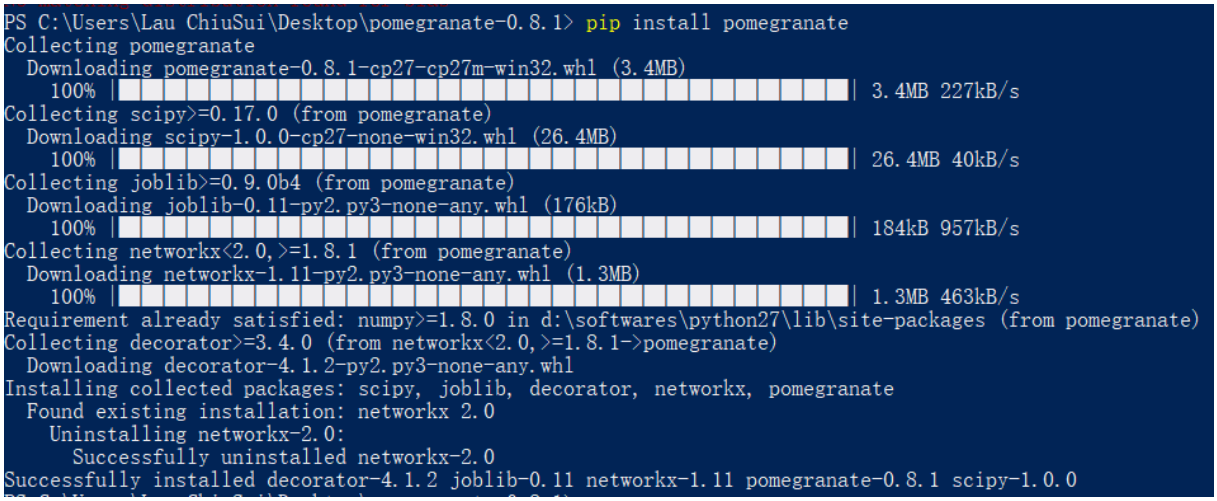




Under Windows

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

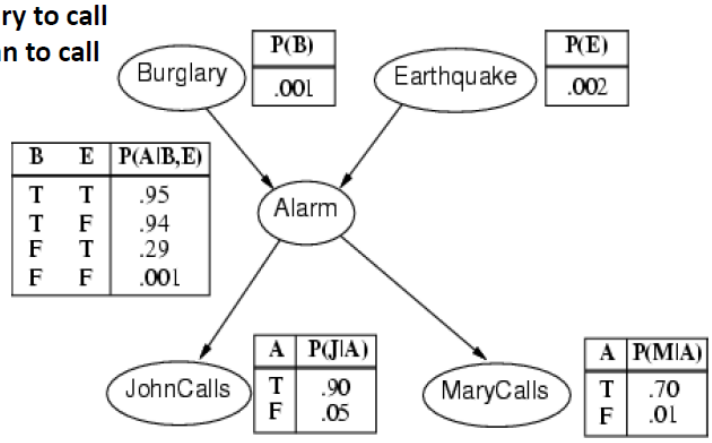
Please refer to `Tutorial_4_Bayesian_Networks.pdf`. I will explain it in class.

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')

Please solve the 2 tasks and hand in a file named E08_YourNumber.pdf, and send it to ai_2020@foxmail.com

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

4 Codes and Results

4.1 Burglary

Code:

```
from pomegranate import *

Burglary = DiscreteDistribution( {'T':0.001, 'F':0.999} )
Earthquake = DiscreteDistribution( {'T':0.002, 'F':0.998} )
Alarm = ConditionalProbabilityTable(
    [
        ['T', 'T', 'T',0.95],
        ['T', 'F', 'T',0.94],
        ['F', 'T', 'T',0.29],
        ['F', 'F', 'T',0.001],
        ['T', 'T', 'F',0.05],
        ['T', 'F', 'F',0.06],
        ['F', 'T', 'F',0.71],
        ['F', 'F', 'F',0.999],
    ],
    [Burglary, Earthquake]
)

JohnCalls = ConditionalProbabilityTable(
    [
        ['T', 'T',0.90],
        ['F', 'T',0.05],
        ['T', 'F',0.10],
        ['F', 'F',0.95],
    ],
    [Alarm]
)

```

```
MaryCalls = ConditionalProbabilityTable(
    [
        ['T', 'T', 0.70],
        ['F', 'T', 0.01],
        ['T', 'F', 0.30],
        ['F', 'F', 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")

model = BayesianNetwork("Burglary")
model.add_states(s1,s2,s3,s4,s5)
model.add_transition(s1,s3)
model.add_transition(s2,s3)
model.add_transition(s3,s4)
model.add_transition(s3,s5)
model.bake()
marginals = model.predict_proba({})

print("P(Alarm) = {}".format(marginals[2].parameters[0]["T"]))

p2 = model.predict_proba({'MaryCalls': 'F'})[3].parameters[0]["T"] *
    marginals[4].parameters[0]["F"]
print("P(J && ~M) = {}".format(p2))

print("P(A | J && ~M) = {}".format(model.predict_proba({'JohnCalls': 'T',
    'MaryCalls': 'F'})[2].parameters[0]["T"]))

print("P(B | A) = {}".format(model.predict_proba({'Alarm': 'T'})[0].
    parameters[0]["T"]))

p5 = model.predict_proba({'JohnCalls': 'T', 'MaryCalls': 'F'})[0].
```

```
    parameters[0]["T"]
print("P(B | J && ~M) = {}".format(p5))

print("P(J && ~M | ~B) = {}".format((1-p5) * p2 / marginals[0].
    parameters[0]["F"]))
```

Result:

```
P(Alarm) = 0.0025164420000009344
P(J && ~M) = 0.05005487546100036
P(A | J && ~M) = 0.013573889331311458
P(B | A) = 0.3735512282818994 6
P(B | J && ~M) = 0.005129858133403523
P(J && ~M | ~B) = 0.04984794900000027
```

4.2 Diagnosing

Code:

```
from pomegranate import *

PatientAge = DiscreteDistribution({'A': 0.10, 'B': 0.30, 'C': 0.60})

CTScanResult = DiscreteDistribution({'IS': 0.7, 'HS': 0.3})

MRIScanResult = DiscreteDistribution({'IS': 0.7, 'HS': 0.3})

Anticoagulants = DiscreteDistribution({'T': 0.5, 'F': 0.5})

StrokeType = ConditionalProbabilityTable(
    [
        ['IS', 'IS', 'IS', 0.8],
        ['IS', 'HS', 'IS', 0.5],
        ['HS', 'IS', 'IS', 0.5],
        ['HS', 'HS', 'IS', 0.0],

        ['IS', 'IS', 'HS', 0.0],
        ['IS', 'HS', 'HS', 0.4],
```

```
            ['HS', 'IS', 'HS', 0.4],
            ['HS', 'HS', 'HS', 0.9],

            ['IS', 'IS', 'SM', 0.2],
            ['IS', 'HS', 'SM', 0.1],
            ['HS', 'IS', 'SM', 0.1],
            ['HS', 'HS', 'SM', 0.1],
        ],
        [CTScanResult, MRIScanResult]
)

Mortality = ConditionalProbabilityTable(
        [
            ['IS', 'T', 'F', 0.28],
            ['HS', 'T', 'F', 0.99],
            ['SM', 'T', 'F', 0.10],
            ['IS', 'F', 'F', 0.56],
            ['HS', 'F', 'F', 0.58],
            ['SM', 'F', 'F', 0.05],
            ['IS', 'T', 'T', 0.72],
            ['HS', 'T', 'T', 0.01],
            ['SM', 'T', 'T', 0.90],
            ['IS', 'F', 'T', 0.44],
            ['HS', 'F', 'T', 0.42],
            ['SM', 'F', 'T', 0.95],
        ],
        [StrokeType, Anticoagulants]
)

Disability = ConditionalProbabilityTable(
        [
            ['IS', 'A', 'N', 0.80],
            ['HS', 'A', 'N', 0.70],
            ['SM', 'A', 'N', 0.90],
            ['IS', 'B', 'N', 0.60],
            ['HS', 'B', 'N', 0.50],
            ['SM', 'B', 'N', 0.40],
            ['IS', 'C', 'N', 0.30],
```

```
            ['HS', 'C', 'N', 0.20],
            ['SM', 'C', 'N', 0.10],

            ['IS', 'A', 'M', 0.10],
            ['HS', 'A', 'M', 0.20],
            ['SM', 'A', 'M', 0.05],
            ['IS', 'B', 'M', 0.30],
            ['HS', 'B', 'M', 0.40],
            ['SM', 'B', 'M', 0.30],
            ['IS', 'C', 'M', 0.40],
            ['HS', 'C', 'M', 0.20],
            ['SM', 'C', 'M', 0.10],

            ['IS', 'A', 'S', 0.10],
            ['HS', 'A', 'S', 0.10],
            ['SM', 'A', 'S', 0.05],
            ['IS', 'B', 'S', 0.10],
            ['HS', 'B', 'S', 0.10],
            ['SM', 'B', 'S', 0.30],
            ['IS', 'C', 'S', 0.30],
            ['HS', 'C', 'S', 0.60],
            ['SM', 'C', 'S', 0.80],
        ],
        [StrokeType, PatientAge]
)

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

model = BayesianNetwork("Diagnosing Problem")
model.add_states(s1, s2, s3, s4, s5, s6, s7)
model.add_edge(s2, s5)
model.add_edge(s3, s5)
```

```
model.add_edge(s5, s6)
model.add_edge(s4, s6)
model.add_edge(s5, s7)
model.add_edge(s1, s7)
model.bake()

print('p1= ', model.predict_proba({'PatientAge': 'B', 'CTScanResult': '
    IS'})[5].parameters[0]['T'])

print('p2= ', model.predict_proba({'PatientAge': 'C', 'MRIScanResult': '
    HS'})[6].parameters[0]['M'])

print('p3= ', model.predict_proba({'PatientAge': 'C', 'CTScanResult': '
    HS', 'MRIScanResult': 'IS'})[4].parameters[0]['SM'])

print('p4= ', model.predict_proba({'PatientAge': 'A'})[3].parameters[0][
    'F'])
```

Result:

```
p1=  0.5948499999999999
p2=  0.2600000000000002
p3=  0.10000000000000044
p4=  0.5
```