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

E09 Bayesian Network

18364066 Yanzuo Lu

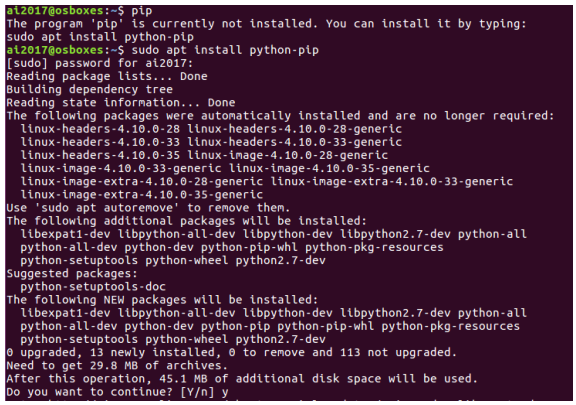
October 30, 2020

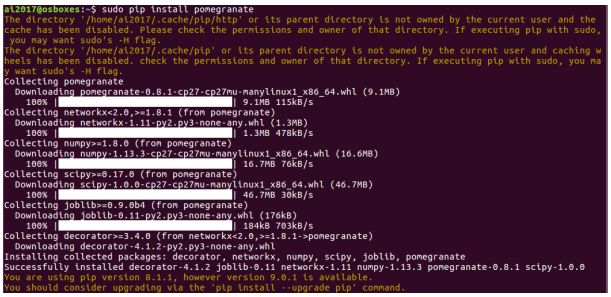
Contents

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`.

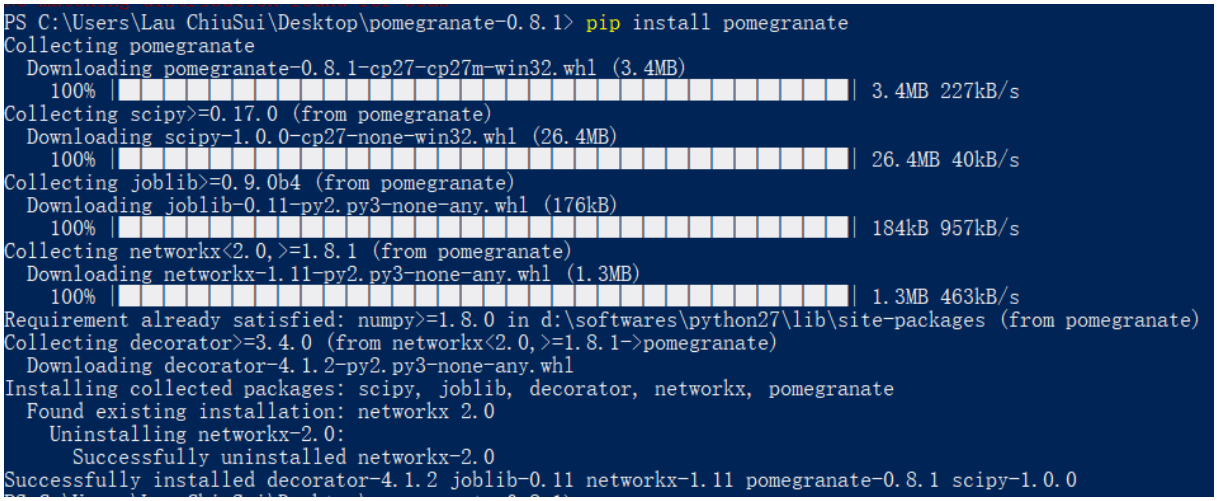




Under Windows

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

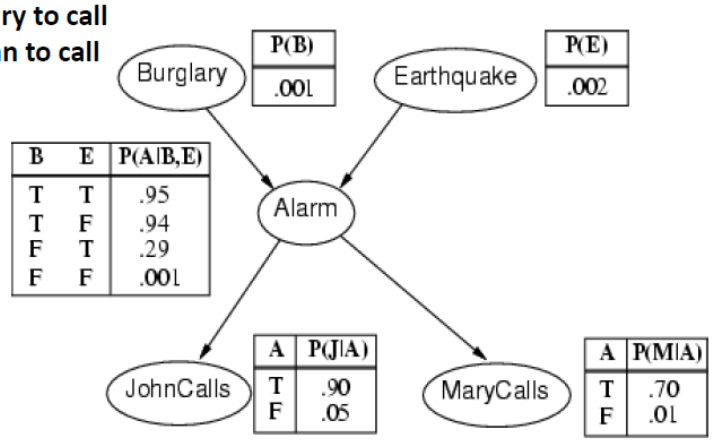
Please refer to `Tutorial_4_Bayesian_Networks.pdf`. I will explain it in class.

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 E08_YourNumber.pdf, and send it to ai_2020@foxmail.com

4 Codes and Results

"Burglary.py"

```
1  from pomegranate import *
2
3  Burglary = DiscreteDistribution({"B": 0.001, "~B": 0.999})
4  Earthquake = DiscreteDistribution({"E": 0.002, "~E": 0.998})
5
6  Alarm = ConditionalProbabilityTable(
7      [
8          ["B", "E", "A", 0.95],
```

```
            ["B", "E", "~A", 0.05],
            ["B", "~E", "A", 0.94],
            ["B", "~E", "~A", 0.06],
            ["~B", "E", "A", 0.29],
            ["~B", "E", "~A", 0.71],
            ["~B", "~E", "A", 0.001],
            ["~B", "~E", "~A", 0.999]
        ], [Burglary, Earthquake]
)

JohnCalls = ConditionalProbabilityTable(
        [
            ["A", "J", 0.9],
            ["A", "~J", 0.1],
            ["~A", "J", 0.05],
            ["~A", "~J", 0.95]
        ], [Alarm]
)

MaryCalls = ConditionalProbabilityTable(
        [
            ["A", "M", 0.7],
            ["A", "~M", 0.3],
            ["~A", "M", 0.01],
            ["~A", "~M", 0.99]
        ], [Alarm]
)

Burglary_state = State(Burglary, name="Burglary")
Earthquake_state = State(Earthquake, name="Earthquake")
Alarm_state = State(Alarm, name="Alarm")
JohnCalls_state = State(JohnCalls, name="JohnCalls")
MaryCalls_state = State(MaryCalls, name="MaryCalls")
```

```
42
43 model = BayesianNetwork("Burglary")
44 model.add_states(Burglary_state, Earthquake_state, Alarm_state,
45     JohnCalls_state, MaryCalls_state)
46 model.add_transition(Burglary_state, Alarm_state)
47 model.add_transition(Earthquake_state, Alarm_state)
48 model.add_transition(Alarm_state, JohnCalls_state)
49 model.add_transition(Alarm_state, MaryCalls_state)
50 model.bake()
51
52 print("P(A) = ", model.predict_proba({})[2].parameters[0]["A"])
53
54 print("P(J && ~M) = ", model.predict_proba({})[3].parameters[0]["J"] *
55     model.predict_proba({})[4].parameters[0]["~M"])
56
57 print("P(A | J && ~M) = ", model.predict_proba({"JohnCalls": "J",
58     "MaryCalls": "~M"})[2].parameters[0]["A"])
59
60 print("P(B | A) = ", model.predict_proba({"Alarm": "A"})[0].parameters
    [0]["B"])
61
62 print("P(B | J && ~M) = ", model.predict_proba({"JohnCalls": "J",
63     "MaryCalls": "~M"})[0].parameters[0]["B"])
64
65 print("P(J && ~M | ~B) = ", model.predict_proba({"Burglary": "~B"})[3].
    parameters[0]["J"] *
66     model.predict_proba({"Burglary": "~B", "JohnCalls": "J"})[4].
        parameters[0]["~M"])
```

"Diagnosing.py"

```
1 from pomegranate import *
2
3 PatientAge = DiscreteDistribution(
```

```
    {
        "0-30": 0.1,
        "31-65": 0.3,
        "65+": 0.6
    }
)

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],

```

```
91          ["Ischemic Stroke",     "0-30" , "Severe", 0.1],
92          ["Hemmorraghic Stroke", "0-30" , "Severe", 0.1],
93          ["Stroke Mimic",        "0-30" , "Severe", 0.05],
94          ["Ischemic Stroke",     "31-65", "Severe", 0.1],
95          ["Hemmorraghic Stroke", "31-65", "Severe", 0.1],
96          ["Stroke Mimic",        "31-65", "Severe", 0.3],
97          ["Ischemic Stroke",     "65+"  , "Severe", 0.3],
98          ["Hemmorraghic Stroke", "65+"  , "Severe", 0.6],
99          ["Stroke Mimic",        "65+"  , "Severe", 0.8]
100     ], [StrokeType, PatientAge]
101 )
102
103 PatientAge_state = State(PatientAge, name="PatientAge")
104 CTScanResult_state = State(CTScanResult, name="CTScanResult")
105 MRIScanResult_state = State(MRIScanResult, name="MRIScanResult")
106 Anticoagulants_state = State(Anticoagulants, name="Anticoagulants")
107 StrokeType_state = State(StrokeType, name="StrokeType")
108 Mortality_state = State(Mortality, name="Mortality")
109 Disability_state = State(Disability, name="Disability")
110
111 model = BayesianNetwork("Diagnosing")
112 model.add_states(PatientAge_state, CTScanResult_state,
        MRIScanResult_state,
113     Anticoagulants_state, StrokeType_state, Mortality_state,
            Disability_state)
114 model.add_transition(CTScanResult_state, StrokeType_state)
115 model.add_transition(MRIScanResult_state, StrokeType_state)
116 model.add_transition(StrokeType_state, Mortality_state)
117 model.add_transition(Anticoagulants_state, Mortality_state)
118 model.add_transition(StrokeType_state, Disability_state)
119 model.add_transition(PatientAge_state, Disability_state)
120 model.bake()
121
```

```
122 print("P(Mortality=\"True\" | PatientAge=\"31-65\" && CTScanResult=\"
        Ischemic Stroke\") = ",
123     model.predict_proba({"PatientAge": "31-65", "CTScanResult": "
            Ischemic Stroke"})[5].parameters[0]["True"])
124
125 print("P(Disability=\"Moderate\" | PatientAge=\"65+\" && MRIScanResult
        =\"Hemmorraghic Stroke\") = ",
126     model.predict_proba({"PatientAge": "65+", "MRIScanResult": "
            Hemmorraghic Stroke"})[6].parameters[0]["Moderate"])
127
128 print("P(StrokeType=\"Stroke Mimic\" | PatientAge=\"65+\" &&
        CTScanResult=\"Hemmorraghic Stroke\" && MRIScanResult=\"Ischemic
        Stroke\") = ",
129     model.predict_proba({"PatientAge": "65+", "CTScanResult": "
            Hemmorraghic Stroke", "MRIScanResult": "Ischemic Stroke"})[4].
            parameters[0]["Stroke Mimic"])
130
131 print("P(Anticoagulants=\"Not Used\" | PatientAge=\"0-30\") = ",
132     model.predict_proba({"PatientAge": "0-30"})[3].parameters[0]["Not
            Used"])
```

本次实验较为简单，只要仔细地敲好字符串值即可，主要发现的问题是 used 要大写到 Used，运算结果如下图所示。

```
PS C:\Users\YanzuoLu\OneDrive\人工智能\实验\E08_BN\src> python .\Burglary.py
P(A) =  0.0025164420000009344
P(J && ~M) =  0.05152705469428172
P(A | J && ~M) =  0.013573889331311458
P(B | A) =  0.37355122828189946
P(B | J && ~M) =  0.005129858133403523
P(J && ~M | ~B) =  0.049847949000000294
```

```
PS C:\Users\YanzuoLu\OneDrive\人工智能\实验\E08_BN\src> python .\Diagnosing.py
P(Mortality="True" | PatientAge="31-65" && CTScanResult="Ischemic Stroke") =  0.5948499999999999
P(Disability="Moderate" | PatientAge="65+" && MRIScanResult="Hemmorraghic Stroke") =  0.2600000000000001
P(StrokeType="Stroke Mimic" | PatientAge="65+" && CTScanResult="Hemmorraghic Stroke" && MRIScanResult="Ischemic Stroke") =  0.10000000000000042
P(Anticoagulants="Not Used" | PatientAge="0-30") =  0.5
```