hw3

April 29, 2021

1 Homework 3

Deyu Qi

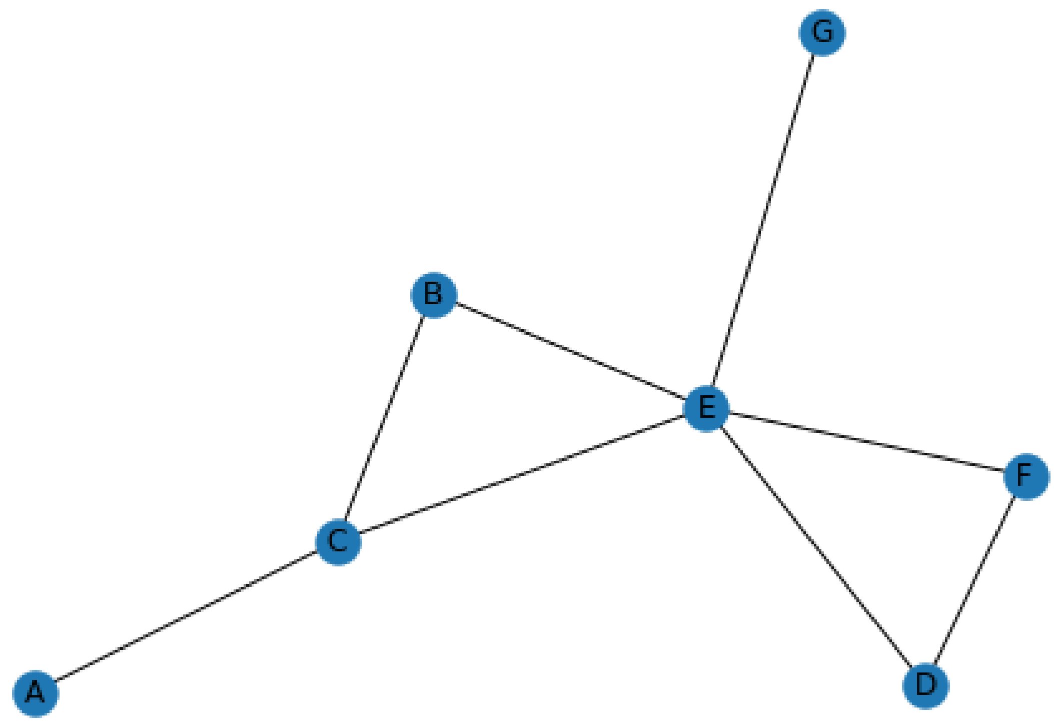
1.1 Problem 1

(a)

[249]:

```
import numpy as np
import matplotlib.pyplot as plt
import networkx
import pyGM as gm

gmv = [gm.Var(i,2) for i in range(0,7)]
names = {0:'A', 1:'B', 2:'C', 3:'D', 4:'E', 5:'F', 6:'G'}
A = gm.Factor([gmv[0]], np.array([0.30, 0.70]))
B = gm.Factor([gmv[1]], np.array([0.60, 0.40]))
D = gm.Factor([gmv[3]], np.array([0.70, 0.30]))
Ca = gm.Factor([gmv[0], gmv[2]], np.array([[0.15,0.85],[0.25,0.75]]))
Ge = gm.Factor([gmv[4], gmv[6]], np.array([[0.10,0.90],[0.30,0.70]]))
Ebc = gm.Factor([gmv[1], gmv[2], gmv[4]], np.array([[[0.40,0.60],
                                                     [0.45,0.55]],
                                                    [[0.60,0.40],
                                                     [0.30,0.70]]]))
Fde = gm.Factor([gmv[3], gmv[4], gmv[5]], np.array([[[0.25,0.75],
                                                     [0.60,0.40]],
                                                    [[0.10,0.90],
                                                     [0.20,0.80]]]))
independency = np.array([A, B, D, Ca, Ge, Ebc, Fde])
graphs = gm.GraphModel(independency)
graphs.drawMarkovGraph(var_labels=names)
plt.show()
```



(b) By tables from the description, we can express the probability for the graph as

$$p(A,B,C,D,E,F,G) = p(A)p(B)p(D)p(C|A)p(G|E)p(E|B,C)p(F|D,E)$$

Therefore, we can express $p(F)$ as

$$\begin{aligned} p(F) &= \sum_{A,B,C,D,E,G} p(A)p(B)p(D)p(C|A)p(G|E)p(E|B,C)p(F|D,E) \\ &= \sum_G \sum_E p(G|E) \sum_D p(D)p(F|D,E) \sum_C \sum_B p(B)p(E|B,C) \sum_A p(A)p(C|A) \end{aligned}$$

$$\sum_A p(A)p(C|A) = p(C)$$

Therefore,

$$p(F) = \sum_G \sum_E p(G|E) \sum_D p(D)p(F|D,E) \sum_C \sum_B p(B)p(E|B,C)p(C)$$

[267]:
```
pAC = A*Ca
pC = pAC.sum([gmv[0]])
print('p(C) =',pC.table)
```

```
p(C) = [0.22 0.78]
0.255
```

$$
\begin{aligned}
\sum_B p(B)p(E|B,C)p(C) &= \sum_B p(B,E|C)p(C) \\
&= p(E|C)p(C) \\
&= p(C,E)
\end{aligned}
$$

Therefore,

$$p(F) = \sum_G \sum_E p(G|E) \sum_D p(D)p(F|D,E) \sum_C p(C,E)$$

[268]:

```
pBEC = Ebc*B
pEC = pBEC.sum([1])
pCE = pEC*pC
print('p(C, E) =\n',pCE.table)
```

```
p(C, E) =
 [[0.1056 0.1144]
 [0.3042 0.4758]]
```

$$\sum_C p(C,E) = p(E)$$

Therefore,

$$p(F) = \sum_G \sum_E p(G|E) \sum_D p(D)p(F|D,E)p(E)$$

[270]:

```
pE = pCE.sum([2])
print('p(E) = ', pE.table)
```

```
p(E) =  [0.4098 0.5902]
```

$$
\begin{aligned}
\sum_D p(D)p(F|D,E)p(E) &= \sum_D p(D,F|E)p(E) \\
&= p(F|E)p(E) \\
&= p(E,F)
\end{aligned}
$$

Therefore,

$$p(F) = \sum_G \sum_E p(G|E)p(E,F)$$

[271]:

```
pDFE = Fde*D
pFE = pDFE.sum([3])
pEF = pFE*pE
print('p(E, F) =\n', pEF.table)
```

```
p(E, F) =
 [[0.084009 0.325791]
 [0.283296 0.306904]]
```

$$\sum_E p(G|E)p(E,F) = p(G)p(F)$$

Therefore,

$$p(F) = \sum_G p(G)p(F)$$

[272]:
```
pG = Ge.sum([4])
pG2 = pG/pG.sum()
pF = pEF.sum([4])
pF2 = pF/pF.sum()
print('p(G) = ', pG2.table, '\np(F) = ', pF2.table)
```

```
p(G) =  [0.2 0.8]
p(F) =  [0.367305 0.632695]
```

And, at the end,

$$\sum_G p(G)p(F) = p(F)$$

Which the table has been shown above. The induced width of this process is 2, by information above.

[273]:
```
GMS = graphs.copy()
GMS.eliminate([0,1,2,3,4,6],'sum')
print(GMS.joint().table)
```

```
[0.367305 0.632695]
```

(c) The induced width is 3, occurs when eliminating node E.

(d)

[12]:
```
graphsDC = graphs.copy()
graphsDC.condition({2:1, 3:0})
graphsDC.drawMarkovGraph(var_labels=names)
plt.show()
```

(e)

$$
\begin{aligned}
p(C,D) &= \sum_{A,B,E,F,G} p(A)p(B)p(D)p(C|A)p(G|E)p(E|B,C)p(F|D,E) \\
&= p(D)\sum_G\sum_F\sum_E p(G|E)p(F|D,E)\sum_B p(B)p(E|B,C)\sum_A p(A)p(C|A) \\
&= p(D)\sum_G\sum_F\sum_E p(G|E)p(F|D,E)\sum_B p(B)p(E|B,C)p(C) \\
&= p(D)\sum_G\sum_F\sum_E p(G|E)p(F|D,E)p(C,E) \\
&= p(D)\sum_G\sum_F p(G)p(F|D)p(C) \\
&= \sum_G\sum_F p(G)p(D,F)p(C) \\
&= \sum_G p(G)p(D)p(C) \\
&= p(D)p(C) \\
&= p(C,D)
\end{aligned}
$$

[277]:
```
pAll = A*B*D*Ca*Ebc*Ge*Fde
pCD = pAll.sum([0,1,4,5,6])
print('p(C = 1, D = 0) = ', pCD[1,0])
```

```
p(C = 1, D = 0) =  0.5459999999999999
```

[282]:
```
GMS2 = graphs.copy()
GMS2.eliminate([0,1,4,5,6],'sum')
GMS2.joint()[1,0]
```

[282]:
```
0.5459999999999999
```

1.2 Problem 2

(a)

[230]:
```
T = np.array([[0.0,0.5,0.5,0.0],[0.0,0.0,1.0,0.0],[0.3,0.0,0.4,0.3],[0.0,0.0,0.
↪3,0.7]])
p0 = np.array([[1.0,0.0,0.0,0.0]])
p2 = p0.dot(T).dot(T)
p2
```

[230]:
```
array([[0.15, 0.  , 0.7 , 0.15]])
```

(b)

[231]:
```
O = np.array([[1,0],[0,1],[1,0],[0,1]])
O
```

[231]:
```
array([[1, 0],
       [0, 1],
       [1, 0],
       [0, 1]])
```

(c)

(1)

[281]:
```
p0.dot(T)
```

[281]:
```
array([[0. , 0.5, 0.5, 0. ]])
```

(2)

[221]:
```
O0 = O[:,0]
O1 = O[:,1]
pY_1X_1 = p0.dot(T)*O0
pY_1X_1 = pY_1X_1/pY_1X_1.sum(axis=1)
print('p(Y1|X1 = 0) = ',pY_1X_1)
```

```
p(Y1|X1 = 0) =  [[0. 0. 1. 0.]]
```

(3)

[222]:
```
print('p(Y2) = ',p0.dot(T).dot(T))
```

```
p(Y2) =  [[0.15 0.   0.7  0.15]]
```

(4)

[223]:
```
pY_2X_1 = pY_1X_1.dot(T)
pY_2X_1X_21 = pY_2X_1*O1
pY_2X_1X_21 = pY_2X_1X_21/pY_2X_1X_21.sum(axis=1)
print('p(Y2|X1 = 0, X2 = 1) = ',pY_2X_1X_21)
```

```
p(Y2|X1 = 0, X2 = 1) =  [[0. 0. 0. 1.]]
```

(5)

[224]:
```
pY_2X_1X_20 = pY_2X_1*O0
pY_2X_1X_20 = pY_2X_1X_20/pY_2X_1X_20.sum(axis=1)
print('p(Y2|X1 = 0, X2 = 0) = ',pY_2X_1X_20)
```

```
p(Y2|X1 = 0, X2 = 0) =  [[0.42857143 0.         0.57142857 0.        ]]
```

(6)

[225]:
```
pY_3X_1X_20 = pY_2X_1X_20.dot(T)
pY_3X_1X_20X_31 = pY_3X_1X_20*O1
pY_3X_1X_20X_31 = pY_3X_1X_20X_31/pY_3X_1X_20X_31.sum(axis=1)
print('p(Y3|X1 = 0, X2 = 0, X3 = 1) = ',pY_3X_1X_20X_31)
```

```
p(Y3|X1 = 0, X2 = 0, X3 = 1) =  [[0.         0.55555556 0.         0.44444444]]
```

(7)

[232]:
```
p100 = p0.copy()
for i in range(0,100):
    p100 = p100.dot(T)
p100
```

[232]:
```
array([[0.12244898, 0.06122449, 0.40816327, 0.40816327]])
```

1.3 Problem 3

(a)

[318]:
```
def HMM(T, O, p0, X):
    r = O[:,X[-1]]
    Rs = np.zeros(shape=[len(X),4])
    Rs[len(X)-1] = r
    for i in range(len(X)-2,-1,-1):
        r = (T*r).max(1)
        r = r*O[:,X[i]]
        Rs[i] = r
    ans = np.empty(len(X))
    ans[0] = int((p0*Rs[0]).argmax())
    for i in range(1,len(X)):
```

```
        ans[i] = int((T*Rs[i])[int(ans[i-1]),:].argmax())
    return ans
print(HMM(T,O,p0,[0,0,1]))
print(HMM(T,O,p0,[0,1,0,0]))
print(HMM(T,O,p0,[0,0,0,0]))
print(HMM(T,O,p0,[0, 1, 0, 0, 0, 1, 0, 0, 1]))
print(HMM(T,O,p0,[0,0,1,1,1,0,0]))
```

```
[0. 2. 3.]
[0. 1. 2. 2.]
[0. 2. 2. 2.]
[0. 1. 2. 2. 0. 1. 2. 0. 1.]
[0. 2. 3. 3. 3. 2. 2.]
```

[319]:
```
print(HMM(T,O,p0,[0,0,0,1]))
print(HMM(T,O,p0,[0,0,0,1,1]))
print(HMM(T,O,p0,[0,0,0,1,0]))
print(HMM(T,O,p0,[0,0,0,1,0,0]))
```

```
[0. 2. 0. 1.]
[0. 2. 2. 3. 3.]
[0. 2. 0. 1. 2.]
[0. 2. 0. 1. 2. 2.]
```