CSE 527 Problem Set 1

Yingxin Cao (1623230)

1.

a)

The chances that you actually have the disease is denoted by, (D=Disease, P=Possitive)

$$
\begin{aligned}
P(D|P) &= \frac{P(PD)}{P(P)} = \frac{P(P|D) \times P(D)}{P(\bar{D}) \cdot P(P|\bar{D}) + P(D) \cdot P(P|D)} = \frac{P(P|D) \times P(D)}{(1-P(D)) \cdot (1-P(\bar{P}|\bar{D})) + P(D) \cdot P(P|D)} \\
&= \frac{l \cdot P(D)}{k + m \cdot P(D)}
\end{aligned}
$$

Where, l = P(P|D), k = 1 - $P(\bar{P}|\bar{D})$, m = $P(P|D) + P(\bar{P}|\bar{D}) - 1$. Thus, k, l, m are all positive numbers.

Take derivative,

$$
\frac{dP(D|P)}{dP(D)} = \frac{l \cdot (k + m \cdot P(D)) - m \cdot l \cdot P(D)}{(k + m \cdot P(D))^2} = \frac{k \cdot l}{(k + m \cdot P(D))^2}
$$

The derivative is poitive, so as P(D) decrease, P(D|P) decrease, Thus the disease is rare is a good news since P(D|P) is lower.

Where, $P(P|D) = 0.99$, $P(\bar{P}|\bar{D}) = 0.99$, $P(D) = 1/10,000$

$$
P(D|P) = 0.0098
$$

b)

According to the Bayes' rule:

$$
\begin{aligned}
P(A|B,E)P(B|E) &= \frac{P(A,B,E)}{P(B,E)} \times \frac{P(B,E)}{P(E)} \\
&= \frac{P(A,B,E)}{P(E)} \\
&= P(A,B|E)
\end{aligned}
$$

$$
\begin{aligned}
\frac{P(B|A,E)P(A|E)}{P(B|E)} &= \frac{\frac{P(A,B,E)}{P(A,E)} \frac{P(A,E)}{P(E)}}{\frac{P(B,E)}{P(E)}} \\
&= \frac{P(A,B,E)}{P(B,E)} \\
&= P(A|B,E)
\end{aligned}
$$

2.

a)

Denote 1 as up, 0 as down.

x2, x3	x1 up	x1 down
1, 1	$\theta_{X1\|1,1}$	1 - $\theta_{X1\|1,1}$
1, 0	$\theta_{X1\|1,0}$	1 - $\theta_{X1\|1,0}$
0, 1	$\theta_{X1\|0,1}$	1 - $\theta_{X1\|0,1}$
0, 0	$\theta_{X1\|0,0}$	1 - $\theta_{X1\|0,0}$

2.

b).

$$\mathcal{L}(\theta : D) = P(D|\theta)$$

$$= \prod_i P(D_i|\theta)$$

$$= \prod_i P(X_{1i}, X_{2i}, X_{3i}; \theta)$$

$$= \prod_i \begin{pmatrix} P(X_2[i] : \theta) \\ \times \\ P(X_3[i] : \theta) \\ \times \\ P(X_1[i] \mid X_2[i], X_3[i] : \theta) \end{pmatrix}$$

$$= \prod_i P(X_1[i] \mid X_2[i] X_3[i]) \prod_i P(X_2[i]) \prod_i P(X_3[i])$$

c). $\arg\max_\theta \mathcal{L}(\theta : D) = \arg\max_\theta \ln[\mathcal{L}(\theta : D)] = \arg\max_\theta l(\theta)$.

Since X_k is binary, $\therefore P(X_k) = \begin{cases} 1, & \theta_{X_k|pa_k} \\ 0, & 1-\theta_{X_k|pa_k} \end{cases}$

$\therefore l(\theta) = \sum_k [\log \theta_{k|pa_k}^{M_{X_k|pa_k}} (1-\theta_{k|pa_k})^{M_{pa_k} - M_{X_k|pa_k}}]\ (k = 1, 2, \dots, N) = \sum_k [M_{X_k|pa_k} \log \theta_{k|pa_k} + (M_{pa_k} - M_{X_k|pa_k}) \log(1-\theta_{k|pa_k})]$

In order to argmax $l(\theta)$, we take derivative w.r.t. each $\vec{\theta}_{X_k|pa_k}$ and set to zero.

$\frac{d\, l(\theta)}{d\, \theta_{X|pa_k}} = 0$, we have, $\frac{M_{X_k|pa_k}}{\hat{\theta}_{k|pa_k}} - \frac{M_{pa_k} - M_{X_k|pa_k}}{1-\hat{\theta}_{k|pa_k}} = 0$

So, $\hat{\theta}_{X_k|pa_k} = \frac{M[x_k, pa_k]}{M[pa_k]}$.

3.

a). Denote Gal80 : A, Gal4 : B, Gal2 : C,

In model 1: $\hat{\theta}_A$, $\hat{\theta}_{B|A=1}$, $\hat{\theta}_{B|A=0}$, $\hat{\theta}_{C|B=1}$, $\hat{\theta}_{C|B=0}$.

In model 2: $\hat{\theta}_A$, $\hat{\theta}_B$, $\hat{\theta}_{C|A=1,B=1}$, $\hat{\theta}_{C|A=1,B=0}$, $\hat{\theta}_{C|A=0,B=1}$, $\hat{\theta}_{C|A=0,B=0}$.

b). For model 1:

$$\mathcal{L}(\theta : D)_1 = \prod_i P(A_i) P(B_i|A_i) P(C_i|B_i)$$

$$= \theta_A^{M_{A=1}} (1-\theta_A)^{M-M_A} \cdot \theta_{B|A=1}^{M_{B|A=1}} (1-\theta_{B|A=1})^{M_{A=1}-M_{B|A=1}}\, \theta_{B|A=0}^{M_{B|A=0}} (1-\theta_{B|A=0})^{M_{A=0}-M_{B|A=0}} \cdot \theta_{C|B=1}^{M_{C|B=1}} (1-\theta_{C|B=1})^{M_{B=1}-M_{C|B=1}} \cdot \theta_{C|B=0}^{M_{C|B=0}} (1-\theta_{C|B=0})^{M_{B=0}-M_{C|B}}$$

For model 2:

$$\mathcal{L}(\theta : D)_2 = \prod_i P(A_i) P(B_i) P(C_i|A_i B_i)$$

$$= \theta_A^{M_A} (1-\theta_A)^{M-M_A} \cdot \theta_B^{M_B} (1-\theta_B)^{M-M_B} \times$$

$$\times\, \theta_{C|A=1,B=1}^{M_{C|A=1,B=1}} (1-\theta_{C|A=1,B=1})^{M_{A=1,B=1}-M_{C|A=1,B=1}} \times \theta_{C|A=1,B=0}^{M_{C|A=1,B=0}} (1-\theta_{A=1,B=0})^{M_{A=1,B=0}-M_{C|A=1,B=0}} \times \theta_{C|A=0,B=1}^{M_{C|A=0,B=1}} (1-\theta_{C|A=0,B=1})^{M_{A=0,B=1}-M_{C|A=0,B=1}} \times \theta_{C|A=0,B=0}^{M_{C|A=0,B=0}} (1-\theta_{C|A=0,B=0})^{M_{A=0,B=0}-M_{A=0,B=0}}$$

c. Mathematically we should take derivative w.r.t. each θ and set to zero, Computationally, we can compute the cpd for each node to get all the θ.

d.

Codes are attached in the end.

CPDs for Model 1

Gal80(0)	**0.5179**
Gal80(1)	0.4821

Gal80	**0**	**1**
Gal4(0)	0.3103	0.6667
Gal4(1)	0.6897	0.3333

Gal4	**0**	**1**
Gal2(0)	0.6296	0.2931
Gal2(11)	0.3704	0.7069

CPDs for Model 2

Gal80(0)	**0.5179**
Gal80(1)	0.4821

Gal4(0)	**0.4821**
Gal4(1)	0.5179

Gal4	**0**	**0**	**1**	**1**
Gal80	0	1	0	1
Gal2(0)	0.6667	0.6111	0.2	0.5
Gal2(1)	0.3333	0.3889	0.8	0.5

Likelihood for two models.

Model	Likelihood Score
1	-218.535478586
2	-223.129027335

e.

Model 1 has higher score, so it's selected.

Code for 3.d

```
from __future__ import division, absolute_import, print_function
import numpy as np
import pandas as pd
from pgmpy.models import BayesianModel
from pgmpy.factors.discrete import TabularCPD
from pgmpy.estimators import K2Score
#%%
#Method 1

# Read data
data = pd.read_table('disc-gal80-gal4-gal2.txt', index_col=0).values

# Get params for Model 1
M = len(data[0])
Ma1 = np.count_nonzero(data[0])
Ma0 = M - Ma1
Mb1 = np.count_nonzero(data[1])
Mb0 = M - Mb1
Mb_a1 = np.count_nonzero(data[1][np.where(data[0]==1)])
Mb_a0 = np.count_nonzero(data[1][np.where(data[0]==0)])
Mc_b1 = np.count_nonzero(data[2][np.where(data[1]==1)])
Mc_b0 = np.count_nonzero(data[2][np.where(data[1]==0)])

theta_a = Ma1 / M
theta_b_1 = Mb_a1 / Ma1
theta_b_0 = Mb_a0 / Ma0
theta_c_1 = Mc_b1 / Mb1
theta_c_0 = Mc_b0 / Mb0

#Compute log Likelihood
def local_l(theta, M, M_):

    return M_ * np.log(theta) + (M - M_) * np.log(1 - theta)

L1 = np.sum(local_l(np.array([theta_a, theta_b_1, theta_b_0, theta_c_1, theta_c_0]),
                        np.array([M, Ma1, Ma0, Mb1, Mb0]),
                        np.array([Ma1, Mb_a1, Mb_a0, Mc_b1, Mc_b0])))
print ('log likelilhood for model 1 is ' + str(L1))
#%%
# Get params for Model 2
M = len(data[0])
Ma1 = np.count_nonzero(data[0])
#Ma0 = M - Ma1
Mb1 = np.count_nonzero(data[1])
#Mb0 = M - Mb1
M11 = len(np.intersect1d(np.where(data[0]==1)[0], np.where(data[1]==1)[0]))
M10 = len(np.intersect1d(np.where(data[0]==1)[0], np.where(data[1]==0)[0]))
M01 = len(np.intersect1d(np.where(data[0]==0)[0], np.where(data[1]==1)[0]))
M00 = len(np.intersect1d(np.where(data[0]==0)[0], np.where(data[1]==0)[0]))
Mc_11 = np.count_nonzero(data[2][np.intersect1d(np.where(data[0]==1)[0], np.where(data[1]==1)
[0])])


Mc_10 = np.count_nonzero(data[2][np.intersect1d(np.where(data[0]==1)[0], np.where(data[1]==0)
```

```
[0])])
Mc_01 = np.count_nonzero(data[2][np.intersect1d(np.where(data[0]==0)[0], np.where(data[1]==1)
[0])])
Mc_00 = np.count_nonzero(data[2][np.intersect1d(np.where(data[0]==0)[0], np.where(data[1]==0)
[0])])

theta_a = Ma1 / M
theta_b = Mb1 / M
theta_c_11 = Mc_11 / M11
theta_c_10 = Mc_10 / M10
theta_c_01 = Mc_01 / M01
theta_c_00 = Mc_00 / M00

#Compute Likelihood
L2 = np.sum(local_l(np.array([theta_a, theta_b, theta_c_11, theta_c_10, theta_c_01,
theta_c_00]),
                       np.array([M, M, M11, M10, M01, M00]),
                       np.array([Ma1, Mb1, Mc_11, Mc_10, Mc_01, Mc_00])))
print ('log likelilhood for model 2 is ' + str(L2))
#%%
#Method 2, Using pgmpy

# Read Data
train_data = pd.DataFrame(pd.read_table('disc-gal80-gal4-gal2.txt', index_col=0).values.T,
columns=['Gal80', 'Gal4', 'Gal2'])

# Define Model
model1 = BayesianModel([('Gal80', 'Gal4'), ('Gal4', 'Gal2')])
model2 = BayesianModel([('Gal80', 'Gal2'), ('Gal4', 'Gal2')])

# Fit the data
model1.fit(train_data)
model2.fit(train_data)

# Get CPDs
print('For Model 1')
print(model1.get_cpds('Gal80'))
print(model1.get_cpds('Gal4'))
print(model1.get_cpds('Gal2'))

print('For Model 2')
print(model2.get_cpds('Gal80'))
print(model2.get_cpds('Gal4'))
print(model2.get_cpds('Gal2'))

#Calculate K2 Score
print('K2Score of Model1 and 2')
print(K2Score(train_data).score(model1))
print(K2Score(train_data).score(model2))
```