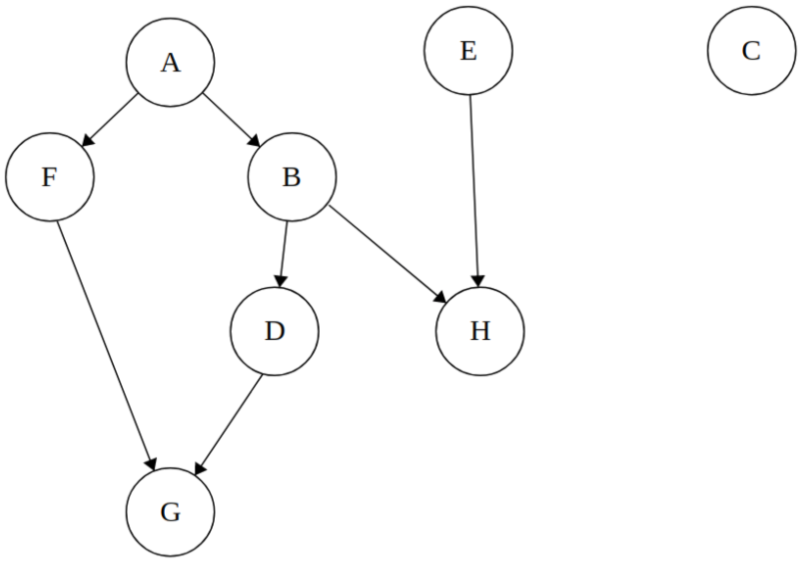
homework 3 solutions

Exercise 1

1.



2.

Let L.M.P. = Local Markov Property that defines a Bayesian Network Graph, i.e.:

each variable is conditionally independent of its non-descendants given its parent variables.

Let HT= Head to Tail, TT = Tail Tail and HH = Head Head,

then

Solution 1

a. $A \perp\!\!\!\perp B$: False: P(A,B) = $\int_{C,D,E,F,G,H} P(A, B, C, D, E, F, G, H)$ = P(B|A) P(A) $\neq$ P(A) P(B)

b. $A \perp\!\!\!\perp C$: True: P(A,C) = P(A) P(C)

c. $A \perp\!\!\!\perp D|\{B, H\}$: True: path from ABD is blocked by B (observed, HT) and path AFGD blocked by unobserved G(HH, no observed descendants)

d. $A \perp\!\!\!\perp E|F$: True : Path AFGDBHE blocked by F(observed, HT), furthermore H, unobserved with no observed descendants, is HH for every path

e. $G \perp\!\!\!\perp E|B$: True: B, observed, HT for GFABHE, furthermore H, unobserved with no observed descendants, is HH for every path.

f. $F \perp\!\!\!\perp C|D$: True: no path from F to C

g. $E \perp\!\!\!\perp D|B$: True: H, unobserved with no observed descendants, is HH for every path

h. $C \perp\!\!\!\perp H|G$: True: no path from C to H

Solution 2

a)False. A is the parent of B, so they are dependent.

b)True. Because C is not linked to the rest of the graph

c)True. $P(A, D|B, H) = \frac{P(A,D,B,H)}{P(B,H)} = \frac{P(A)P(D|B)P(B|A)P(H|B)}{P(B)P(H|B)} = \frac{P(A)P(D|B)P(B|A)}{P(B)} =$ (using Bayes Th.) $P(A|B)P(D|B)$

d)True. $P(A, E|F) = \frac{P(A,E,F)}{P(F)} = \frac{P(A)P(E)P(F|A)}{P(F)} = P(A|F)P(E) = P(A|F)P(E|F)$ because E and F are independent and so $P(E) = P(E|F)$

e)True. $P(G, E|B) = \frac{P(G,E,B)}{P(B)} = \frac{P(G)P(E)P(B)}{P(B)} = P(G)P(E) = P(G|B)P(E|B)$

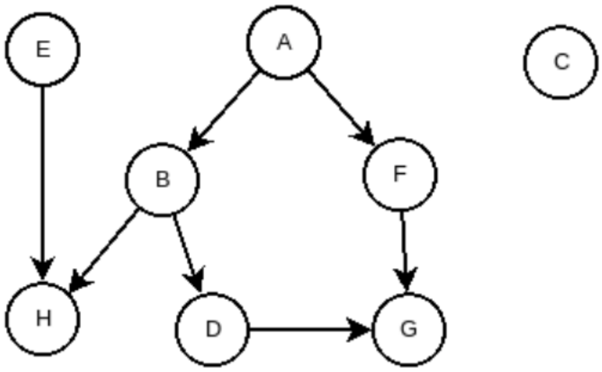
f)True. Because C is not linked to the rest of the graph

g)True.
$P(E, D|B) = \frac{P(E,D,B)}{P(B)} = \frac{P(E)P(D|B)P(B)}{P(B)} = P(E)P(D|B) = P(E|B)P(D|B)$
because E and B are independent if H is not observed

h)True. Because C is not linked to the rest of the graph

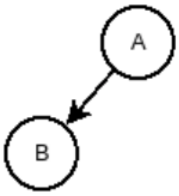
Solution 3

Solution For the solution, I used the algorithm proposed by the following exercise sheet proposed by MIT: http://web.mit.edu/jmn/www/6.034/d-separation.pdf



a) FALSE

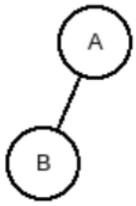
ANCESTRAL GRAPH



MORALISE

Only one parent, thus nothing is changed

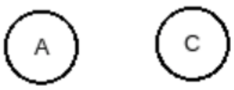
DISORIENT



A and B are connected, therefore they are NOT unconditionally independet

b) TRUE

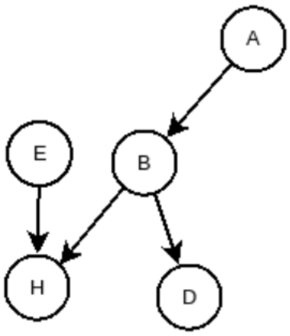
ANCESTRAL GRAPH



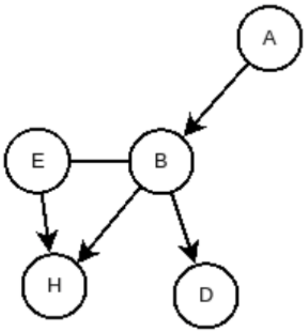
We can immeadiately deduce that A and C are unconditionally independent as they are not connected

c) TRUE

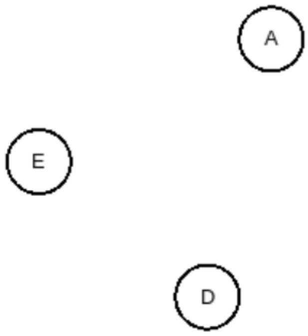
ANCESTRAL GRAPH



MORALISE



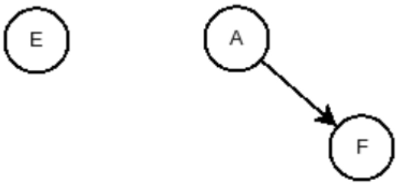
DISORIENT AND
REMOVED OBSERVED



A and D are not connected, thus
they are cond. indep. given {B,H}

d) TRUE

ANCESTRAL G.

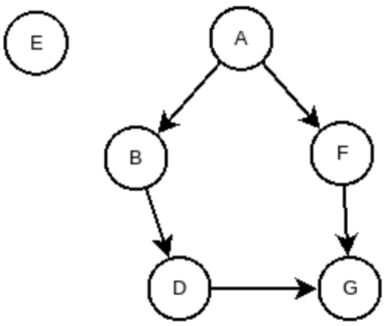


MORALISE: nothing to do
DISORIENT and REMOVE OBSERVED

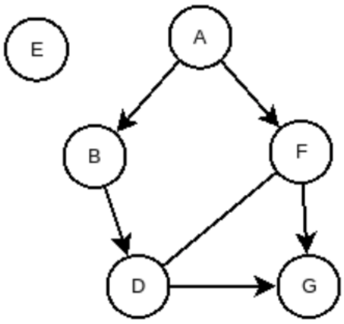
E and A are conditionally indep. given F

e) TRUE

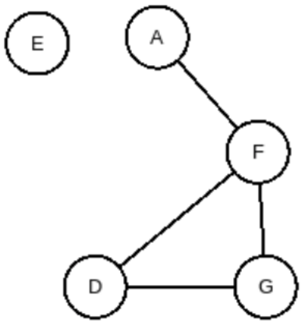
ANCESTRAL GRAPH



MORALISE



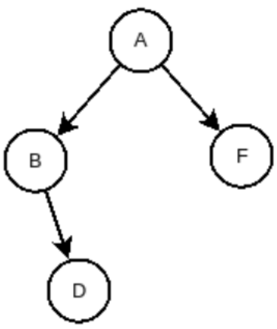
DISORIENT AND
REMOVE OBSERVED



G and E are not connected,
thus they are cond. indep. given B

f) TRUE

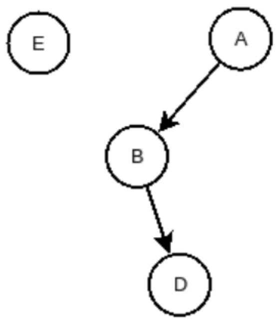
ANCESTRAL GRAPH



We can immediately see that F and C
would be disconnected even after moralise
phase, therefore they are cond. indep. given D

g) TRUE

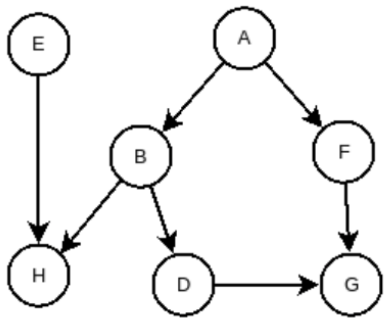
ANCESTRAL GRAPH

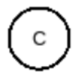


Also here it possible to see that E and D will be disconnected after moralisation phase. Therefore E and D are cond. indep. given B

h) TRUE

ANCESTRAL G.





Same as before, C and H will be disconnected after moralisation of the graph, thus they are cond. indep. given G

Exercise 2

```
# number of componentes
K=2

#parameters
eta=5
alfa=0.8


def model(data):
    #number of observations (index i for z and x)
    N=len(data)

    #conditional independent sample from theta, length=K
    with pyro.plate("components",K):
        theta= pyro.sample("theta", dist.Dirichlet(alfa*torch.ones(K)))

    #conditional independent sample from mu
    with pyro.plate("components", K):
        mu=pyro.sample("mu", dist.Normal(0., eta))

    #sample from a Categorical variable, each component depends on the previous one,
length is the
    # number of the data that we have
    z=np.zeros(N)
    theta_start=theta[random.randint(0,1)] #parameter for z[0]
    z[0]= pyro.sample("z", dist.Categorical(probs=theta_start)) #sample from z[0]
    for i in range (1,N):
        z[i] =pyro.sample("z", dist.Categorical(probs=theta[int(z[i-1])]))

     #conditional independent sample from a normal distribution which mean depends o
n z
    with pyro.plate("data", len(data)):
         x=pyro.sample("x", dist.Normal(mu[z],1), obs=data)


    print("theta=", theta,"\nmu=", mu, "\nz=", z, "\nx=", x)

model(data=[7,0.8,0.1,6,0.5,6.8])
```

```
theta= tensor([[0.3076, 0.6924],
        [0.1218, 0.8782]])
mu= tensor([ 6.1759, -9.5655])
z= [0. 1. 1. 0. 1. 1.]
x= [7, 0.8, 0.1, 6, 0.5, 6.8]
```