Margot Wagner
A53279875
CSE 250a HW3
Due: 10/27/20

3.1 Inference in a chain

(a) Prove $P(X_{t+1}=j \mid X_1=i) = [A^t]_{ij}$

given $A_{ij} = P(X_{t+1}=j \mid X_t=i)$

$$P(X_{t+1}=j \mid X_1=i) = \sum_k P(X_{t+1}=j, X_t=k \mid X_1=i) \quad \text{marginalization}$$

$$= \sum_k P(X_t=k \mid X_1=i)P(X_{t+1}=j \mid X_t=k, X_1=i) \quad \text{product rule}$$

$$= \sum_k P(X_t=k \mid X_1=i)P(X_{t+1}=j \mid X_t=k) \quad \text{cond. ind}$$

$$[A^t]_{ij} = P(X_{t+1}=j \mid X_1=i)$$

$\therefore$

$$[A^{t-1}]_{ij} = P(X_t=j \mid X_1=i)$$

$$= \sum_k [A^{t-1}]_{ik} P(X_{t+1}=j \mid X_t=k) \quad \text{subst.}$$

$$= \sum_k [A^{t-1}]_{ik} A_{kj} \quad \text{subst } A_{ij} = P(X_{t+1}=j \mid X_t=i)$$

$$= [A^t]_{ij}$$

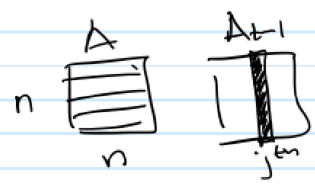
(b) Algo that scales $O(n^2 t)$

Want $[A^t]_{ij}$

$$[A^t]_{ij} = \sum_k A_{kj}[A^{t-1}]_{ik}$$

For each time step:
1. Multiply entire A matrix by j^{th} column (n multiplications)
2. Sum across row to get values (n summations)

↳ $O(n^2 t)$



(c) $O(n^3 \log_2 t)$

We can compute matrices at later time steps by using previous time steps squared.
For example, A^8 can be obtained by

$$A \longrightarrow A^2 \longrightarrow A^4 \longrightarrow A^8$$

where $\rightarrow$ is a squaring operation. Each squaring operation is a matrix, matrix multiplication of $O(n^3)$. The number of times it needs to be squared is $\log_2 t$ times $(\log_2 8 = 3)$

If the number is not a power of two, you can find the next largest power of 2 and multiply it by smaller powers of 2.

$$A^{15} \rightarrow A^8 \cdot A^4 \cdot A^2 \cdot A$$

$O(\log_2 t n^3)$ $\qquad$ $O(\log_2 t \cdot n^3)$

Overall, this gives $O(\log_2 t \cdot n^3)$

matrix mult

# of matrix mult

(d) For the multiplication in part (b), we would still have to multiply n-elements but only need to sum over the m nonzero elements

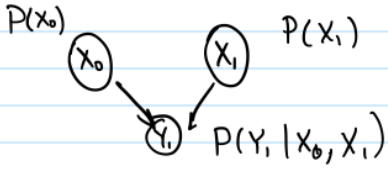
$O(n^2 t) \rightarrow O(mnt)$ where $m << n$

(e) $P(X_1 = i \mid X_T = j) = \dfrac{P(X_T = j \mid X_1 = i) P(X_1 = i)}{P(X_T = j)}$ Bayes rule

$$= \frac{P(X_T = j \mid X_1 = i) P(X_1 = i)}{\sum_k P(X_1 = k) P(X_T = j \mid X_1 = k)}$$ marginalization

$$= \frac{[A^{T-1}]_{ij} P(X_1 = j)}{\sum_k [A^{T-1}]_{jk} P(X_1 = k)}$$ subst. $P(X_T \mid X_1 = i) = [A^{T-1}]_{ij}$

3.2 More inference in a chain



a

$$P(Y_1 | X_1) = \sum_{x_0} P(Y_1, X_0 = x_0 | X_1) \quad \text{marginalization}$$

$$= \sum_{x_0} P(Y_1 | X_0 = x_0, X_1) P(X_0 = x_0 | X_1) \quad \text{product rule}$$

$$= \sum_{x_i} P(Y_1 | X_0 = x_i, X_1) P(X_0 = x_i) \quad \text{ind}$$

b

$$P(Y_1) = \sum_{x_i} P(Y_1, X_1 = x_i) \quad \text{marginalization}$$

$$= \sum_{x_i} \underbrace{P(X_1 = x_i)}_{\text{CPT}} \underbrace{P(Y_1 | X_1 = x_i)}_{\text{part (a)}} \quad \text{product rule}$$

$$= \sum_{x_i} \sum_{x_j} P(X_1 = x_i) P(X_0 = x_j) P(Y_1 | X_0 = x_j, X_1 = x_i)$$

(c) $P(X_n | Y_1, Y_2, \dots Y_{n-1}) = P(X_n)$ by rule #3

(d)

$$P(Y_n | X_n, Y_1, Y_2, \dots Y_{n-1}) = \sum_x P(Y_n, X_{n-1} = x | X_n, Y_1 \dots Y_{n-1}) \quad \text{marginalization}$$

$$= \sum_x P(X_{n-1} = x | \cancel{X_n}, Y_1 \dots Y_{n-1}) P(Y_n | X_{n-1} = x, X_n, \cancel{Y_1, \dots Y_{n-1}}) \quad \text{product rule}$$

cond ind   cond. ind

$$= \sum_x \underbrace{P(X_{n-1} = x | Y_1, \dots Y_{n-1})}_{\text{known}} \underbrace{P(Y_n | X_{n-1} = x, X_n)}_{\text{CPT}}$$

(e)

$$P(Y_n | Y_1, Y_2, \dots Y_{n-1}) = \sum_x \sum_{x'} P(Y_n, X_n = x, X_{n-1} = x' | Y_1, \dots Y_{n-1}) \quad \text{marginalization}$$

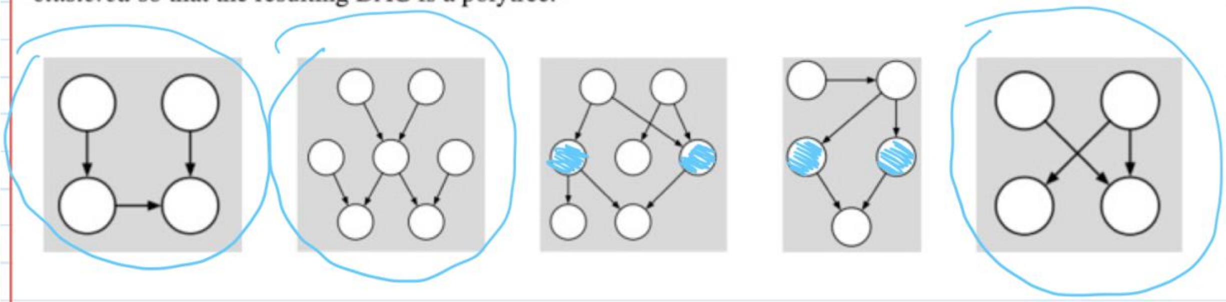
$$= \sum_x \sum_{x'} P(X_n = x | \cancel{Y_1, \dots Y_{n-1}}) P(X_{n-1} = x' | \cancel{X_n = x}, Y_1 \dots Y_{n-1}) P(Y_n | X_n = x, X_{n-1} = x', \cancel{Y_1, \dots Y_{n-1}}) \quad \text{product rule}$$

cond ind rule #3   cond ind   cond ind

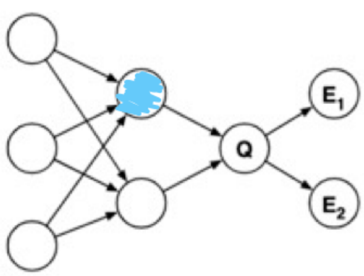
$$= \sum_x \sum_{x'} P(X_n = x) P(X_{n-1} = x' | Y_1 \dots Y_{n-1}) P(Y_n | X_n = x, X_{n-1} = x')$$

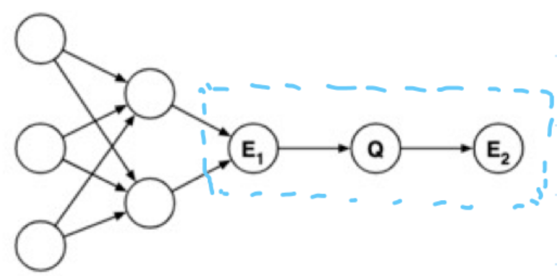
3.3 Node clustering + polytrees

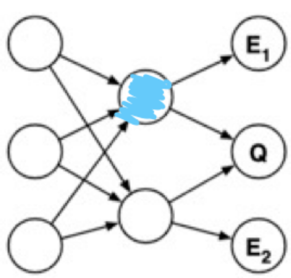
In the figure below, *circle* the DAGs that are polytrees. In the other DAGs, shade **two** nodes that could be *clustered* so that the resulting DAG is a polytree.

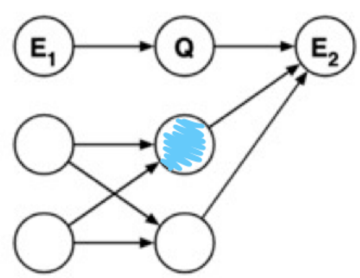


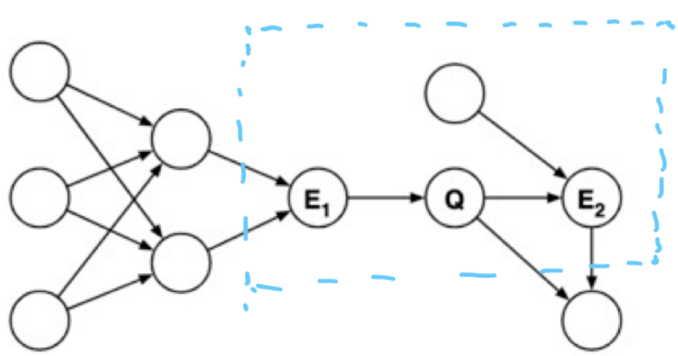
3.4 Cutsets + Polytrees











3.5 Node Clustering

Y_1	Y_2	Y_3	Y	$P(Y\|X=0)$	$P(Y\|X=1)$	$P(Z_1=1\|Y)$	$P(Z_2=1\|Y)$
0	0	0	1	0.504	0.048	0.2	0.9
1	0	0	2	0.056	0.192	0.3	0.8
0	1	0	3	0.126	0.072	0.4	0.7
0	0	1	4	0.216	0.032	0.5	0.6
1	1	0	5	0.014	0.288	0.6	0.5
1	0	1	6	0.024	0.128	0.7	0.4
0	1	1	7	0.054	0.048	0.8	0.3
1	1	1	8	0.006	0.192	0.9	0.2

$P(Y|X) = P(Y_1, Y_2, Y_3 | X) = P(Y_1|X)P(Y_2|X)P(Y_3|X)$

ex: $P(Y_1=0 | X=0) = P(Y_1=0 | X=0)\,P(Y_2=0|X=0)\,P(Y_3=0|X=0)$

$= (1-0.1)(1-0.2)(1-0.3)$

$= 0.504$

$P(Z_1|Y) = P(Z_1 | Y_1, Y_2, Y_3)$

$P(Z_1=1 | Y=1) = P(Z_1=1 | Y_1=0, Y_2=0, Y_3=0)$

$= 0.2$

3.6 Stochastic Simulation

(a) $\sum_z P(Z=z \mid B_1, B_2, \ldots B_n) = \underline{1}$

$$= \sum_z \left(\frac{1-\alpha}{1+\alpha}\right)\alpha^{|z-f(B)|}$$

$$= \sum_{z=-\infty}^{\infty}\left(\frac{1-\alpha}{1+\alpha}\right)\alpha^{|z-f(B)|}$$

$$= \left(\frac{1-\alpha}{1+\alpha}\right)\sum_{z=-\infty}^{+\infty}\alpha^{|z-f(B)|}$$ remove const.

$$= \left(\frac{1-\alpha}{1+\alpha}\right)\sum_{t=-\infty}^{\infty}\alpha^{|t|}$$ define $t = z - f(B)$, $t \in (-\infty, \infty)$

$$= \left(\frac{1-\alpha}{1+\alpha}\right)\left(\alpha^0 + 2\sum_{t=1}^{\infty}\alpha^t\right)$$ expand series (symmetric over all values except 0)

$$= \left(\frac{1-\alpha}{1+\alpha}\right)\left(\alpha^0 + 2\left(\frac{\alpha}{1-\alpha}\right)\right)$$ subst series expansion $\frac{\alpha}{1-\alpha} = \sum_{z=1}^{\infty}\alpha^z$

$$= \left(\frac{1-\alpha}{1+\alpha}\right)\left(1 + \frac{2\alpha}{1-\alpha}\right)$$

$$= \left(\frac{1-\alpha}{1+\alpha}\right)\left(\frac{1-\alpha}{1-\alpha} + \frac{2\alpha}{1-\alpha}\right)$$

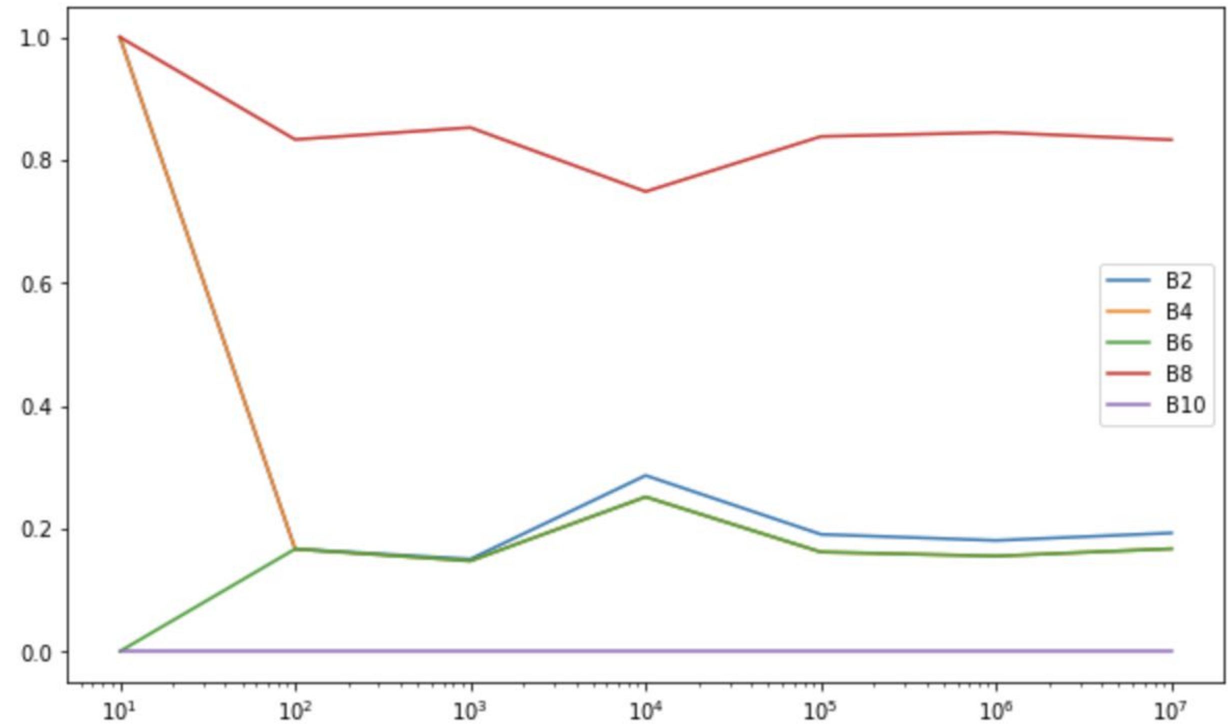
$$= \left(\frac{1-\alpha}{1+\alpha}\right)\left(\frac{1-\alpha+2\alpha}{1-\alpha}\right)$$

$$= \left(\frac{1-\alpha}{1+\alpha}\right)\left(\frac{1+\alpha}{1-\alpha}\right) = 1$$

(b) / (c)

i	$P(B_i = 1 \mid Z = 128)$
2	0.192
4	0.167
6	0.167
8	0.833
10	3.32×10^{-269}

(c)



(d) Source code

```
In [2]: import random
        import matplotlib.pyplot as plt
        import numpy as np
```

```
In [3]: random.seed(4)
        def randn_bits(n):
            '''
            Generate n binary random variables {0,1}; In order B1B2...Bn
            '''
            bits = []
            for i in range(n):
                bits.append(random.randint(0, 1))

            return bits
```

```
In [4]: def fb(bits):
            '''
            Nonnegative integer whose binary representation is given by
            BnBn-1...B2B1
            '''
            sum = 0
            for i in range(len(bits)):
                sum += bits[i]*(2**(i))

            return sum
```

```
In [5]: def p_z_given_bs(alpha, z, bits):
            '''
            Prior probability of P(Z|B1, B2,..., Bn)
            '''
            return (((1 - alpha)/(1 + alpha))*(alpha**abs(z - fb(bits))))
```

In [6]:
```
def indicator(desired, actual):
    '''
    Indicator function
    '''
    if desired == actual:
        return 1
    else:
        return 0
```

In [7]:
```
def p_bi_z(n_samples):
    alpha = 0.2
    z = 128
    n_rvs = 10

    num = [0]*5
    denom = 0
    for i in range(n_samples):
        bits = randn_bits(n_rvs)
        for j in range(5):
            num[j] += indicator(bits[2*j + 1], 1)*p_z_given_bs(alpha, z, bits)
        denom += p_z_given_bs(alpha, z, bits)

    probs = [x / denom for x in num]
    return probs
```

In [8]:
```
n_samples = [10**i for i in range(1, 8)]
probs = []
for n in n_samples:
    probs.append(p_bi_z(n))
```

In [9]:
```
plt.figure(figsize=(10,6))
lineObjects = plt.plot(n_samples, probs)
plt.legend(iter(lineObjects), ("B2", "B4", "B6", "B8", "B10"))
plt.xscale("log")
```

In [15]:
```
print(probs[-1])
```

```
[0.1922881447326238, 0.16672044933143992, 0.16671873260430486, 0.8332812673973782, 3.32453904
5155545e-269]
```

3.7 Even more inference

(a) Markov blanket

$$P(B \mid A,C,D) = \frac{P(D \mid \overset{\text{cond ind}}{\cancel{A}},B,C)\,P(B \mid A,\overset{\text{marg ind}}{\cancel{C}})}{P(D \mid A,C)} \qquad \text{Bayes' rule}$$

$$= \frac{P(D \mid B,C)\,P(B \mid A)}{\sum_b P(D,B=b \mid A,C)} \qquad \text{marginalization}$$

$$= \frac{P(D \mid B,C)\,P(B \mid A)}{\sum_b P(B=b \mid A,\underset{\text{marg ind}}{\cancel{C}})\,P(D \mid B=b,\underset{\text{cond ind}}{\cancel{A}},C)} \qquad \text{product rule}$$

$$= \frac{P(D \mid B,C)\,P(B \mid A)}{\sum_b P(B=b \mid A)\,P(D \mid B=b,C)}$$

(b) Conditional Independence

$$P(B \mid A,C,D,\cancel{E},\cancel{F}) = \overset{\text{part a}}{P(B \mid A,C,D)}$$

cond ind by rule #3

$$\overset{\text{CPTs}}{=} \frac{P(D \mid B,C)\,P(B \mid A)}{\sum_b P(B=b \mid A)\,P(D \mid B=b,C)}$$

(c) More conditional independence

$$P(B,E,F \mid A,C,D) = P(B \mid A,C,D)\,P(E \mid \cancel{A},\cancel{B},C,\cancel{D})\,P(F \mid A,\cancel{B},\cancel{C},\cancel{D},\cancel{E})$$

cond ind   cond ind

$$= \underbrace{P(B \mid A,C,D)}_{\text{part (a)}}\,\underbrace{P(E \mid C)}_{\text{CPT}}\,\underbrace{P(F \mid A)}_{\text{CPT}}$$

3.8 More liklihood weighting

→ in CPTs, given P(node | parents (node))

(a) $P(Q=q \mid E=e)$

$$P(Q=q \mid E=e) \approx \frac{\sum_{t=1}^{T} I(Q, q_t)\, P(E=e \mid Y=y_t, Z=z_t)}{\sum_{t=1}^{T} P(E=e \mid Y=y_t, Z=z_t)}$$

(b) $P(Q_1=q_1, Q_2=q_2 \mid E_1=e_1, E_2=e_2)$

$$P(Q_1=q_1, Q_2=q_2 \mid E_1=e_1, E_2=e_2)$$

$$\approx \frac{\sum_{t=1}^{T} I(Q_1, q_{1t})\, I(Q_2, q_{2t})\, P(E_1=e_1 \mid Q_1=q_{1t}, X=x_t)\, P(E_2=e_2 \mid E_1=e_1, Z=z_t)}{\sum_{T} P(E_1=e_1 \mid Q_1=q_1, X=x_t)\, P(E_2=e_2 \mid E_1=e_1, Z=z_t)}$$