Caden Kroonenberg
Problem Set 8
EECS 649
4-3-22

1

P(Red) = 0.3
P(B1) = 0.333 P(Red | B1) = 0.182
P(B2) = 0.333 P(Red | B2) = 0.444
P(B3) = 0.333 P(Red | B3) = 0.3

P(B1 | Red) = P(Red | B1)*P(B1)/P(Red)
= 0.182*0.333/0.3
= 0.202

P(B2 | Red) = P(Red | B2)*P(B2)/P(Red)
= 0.444*0.333/0.3
= 0.493

P(B3 | Red) = P(Red | B1)*P(B3)/P(Red)
= 0.3*0.333/0.3
= 0.333

We can say P(Red) = 0.3 because 9 of the 30 total balls are red. We can then calculate P(Box = B1, B2, B3) as 1/3 for each box because the problem states a box is selected at random. From there, we can calculate P(Red | Box) as (# of red balls in the given box)/(# of balls in the box). Using Bayes' Rule, we can calculate the probability of the box chosen given the color of the ball.

2

$P(A \mid +d) = \frac{P(A,+d)}{P(+d)}$

$P(+a \mid +d) = \frac{P(+a)\sum_B \sum_C P(+a,B,C,+d)}{P(+d)}$

$P(+a \mid +d) = \frac{P(+a)\sum_B \sum_C P(B|+a)P(C|+a)P(+d|B,C)}{P(+d)}$

= P(+a)*(P(+b|+a)P(+c|+a)P(+d|+b,+c) + P(+b|+a)P(-c|+a)P(+d|+b,-c) + P(-b|+a)P(+c|+a)P(+d|-b,+c) + P(-b|+a)P(-c|+a)P(+d|-b,-c))/P(+d)
= 0.5*(1*1*1 + 1*0*0.5 + 0*1*0.5 + 0*0*0)/P(+d)
= α*0.5

$P(-a \mid +d) = \frac{P(-a)\sum_B \sum_C P(-a,B,C,+d)}{P(+d)}$

$P(-a \mid +d) = \frac{P(-a)\sum_B \sum_C P(B|-a)P(C|-a)P(+d|B,C)}{P(+d)}$

= P(-a)*(P(+b|-a)P(+c|-a)P(+d|+b,+c) + P(+b|-a)P(-c|-a)P(+d|+b,-c) + P(-b|-a)P(+c|-a)P(+d|-b,+c) + P(-b|-a)P(-c|-a)P(+d|-b,-c))/P(+d)
= 0.5*(0.5*0.5*1 + 0.5*0.5*0.5 + 0.5*0.5*0.5 + 0.5*0.5*0)/P(+d)
= α*0.25

P(A | +d) = ⟨α*0.5, α*0.25⟩ = ⟨0.667, 0.333⟩
P(+a | +d) = 0.667

3

$P(-j, -m, +b, +e) = P(-j \mid A)*P(-m \mid A)*P(A \mid +b, +e)*P(+b)*P(+e)$

$= P(+b)*P(+e) *\sum_A P(A \mid +b, +e)P(-j \mid A)P(-m \mid A)$

$= 0.000002 *\sum_A P(A \mid +b, +e)P(-j \mid A)P(-m \mid A)$

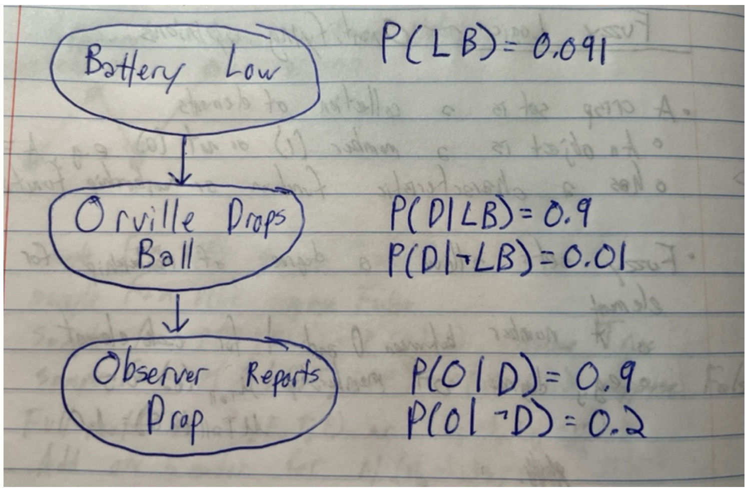
$= 0.000002*(P(+a \mid +b, +e)P(-j \mid +a)P(-m \mid +a) + P(-a \mid +b, +e)P(-j \mid -a)P(-m \mid -a))$

$= 0.000002*(0.98*0.05*0.3 + 0.02*0.99*0.99)$

$= 0.000000068604$

$= 6.8604*10^{-8}$

4



$\mathbf{P}(LB \mid +o) = \langle P(+lb \mid +o), P(-lb \mid +o)\rangle$

$P(+lb \mid +o) = P(+lb, +o)/P(+o)$

$= \alpha*P(+lb)\sum_D P(+o|D)P(D \mid +lb)$

$= \alpha*P(+lb)(P(+o|+d)P(+d \mid +lb) + P(+o|-d)P(-d \mid +lb)$

$= \alpha*0.091(0.9*0.9+0.2*0.01)$

$= \alpha*0.073892$

$P(-lb \mid +o) = P(-lb, +o)/P(+o)$

$= \alpha*P(-lb)\sum_D P(+o|D)P(D \mid -lb)$

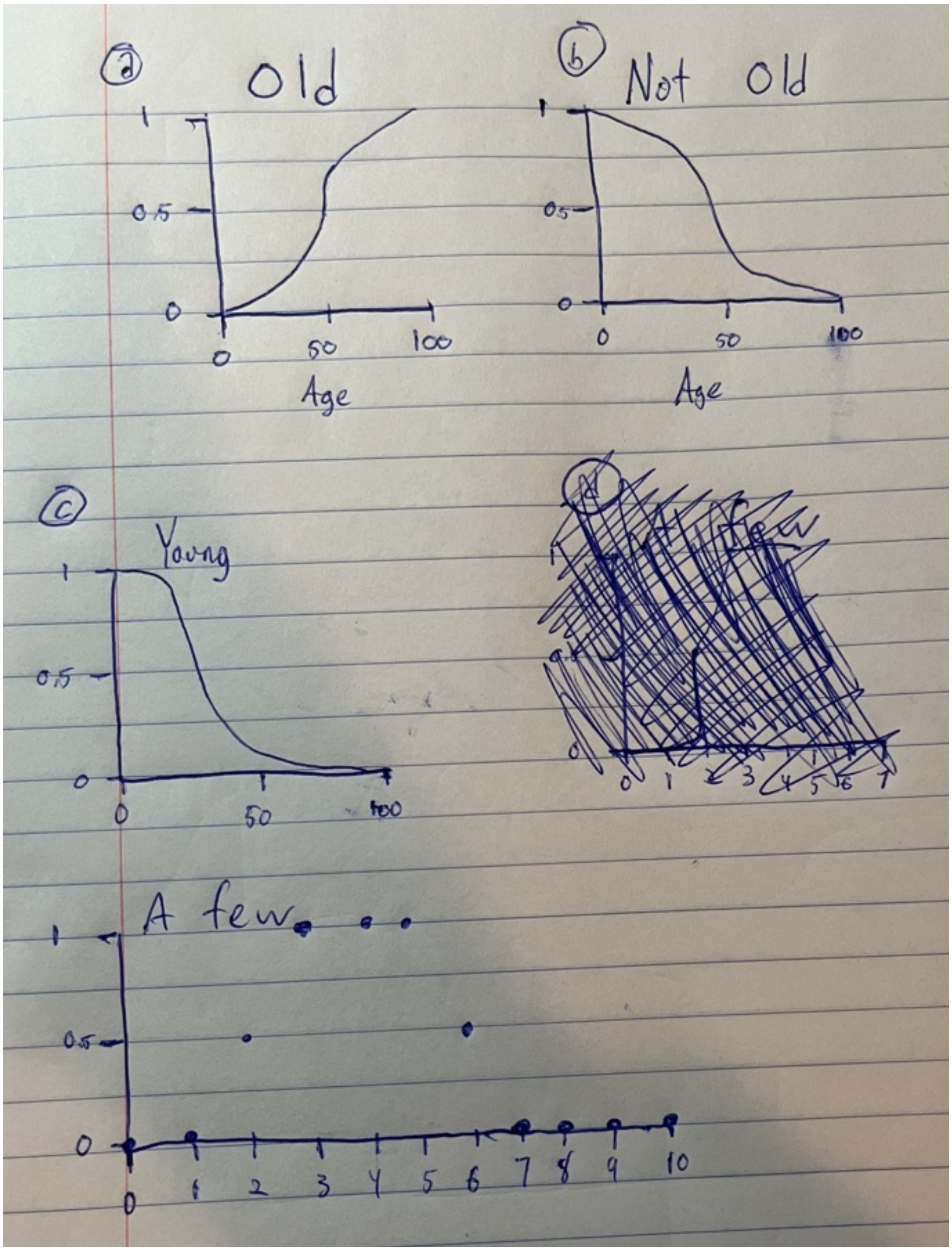
$= \alpha*P(-lb)(P(+o|+d)P(+d \mid -lb) + P(+o|-d)P(-d \mid -lb)$

$= \alpha*0.909(0.9*0.01+0.2*0.99)$

$= \alpha*0.188163$

$\mathbf{P}(+lb \mid O) = \langle \alpha*0.073892, \alpha*0.188163 \rangle = \langle 0.282, 0.718\rangle$

5



e. “Young” and “Not Old” are not necessarily the same categories (although my fuzzy set graphs look similar for the two categories). Most people would consider an 18 year old person definitely young but also somewhat old. If Young and Not Old were the same category, then Old and Young would need to be exact opposites. An 18 year old could only be considered definitely young if they were considered not old at all, but this is not the case.

6

```python
from random import random

TRIALS = 10**6

evidence = 0
query = 0
pall = []
pdenomall = []

for trial in range(1,TRIALS+1):
    randA, randB, randC, randD = ( random() for i in range(4) )
    A = (randA < 0.5)
    if A:
        B = (randB < 1)
        C = (randC < 1)
    else:
        B = (randB < 0.5)
        C = (randC < 0.5)
    if B and C:  # +B and +C
        D = (randD < 1)
    elif B and not C:  # +B and -C
        D = (randD < 0.5)
    elif not B and C:  # -B and +C
        D = (randD < 0.5)
    else:   # -B and -C
        D = (randD < 0)

    # REJECTION SAMPLING: ONLY PROCESS IF EVIDENCE True
    if D:
        evidence += 1
        if A:
            query += 1
        p = query/evidence
        pall.append(p)
    pdenom = evidence/trial
    pdenomall.append(pdenom)

print("P(A|D) ~=", pall[-1])
print("P(D) ~=", pdenomall[-1])
print("P(A,D) ~=", pall[-1]*pdenomall[-1])
```

```
➜  ps8 git:(main) ✗ python3 monte_carlo.py
P(A|D) ~= 0.6672006514614497
P(D) ~= 0.749085
P(A,D) ~= 0.49979000000000007
```

The Monte Carlo simulation verifies my results from 8.2 that P(+a|+d) = 0.667 and P(+a, +d) = 0.5. I used normalization and did not calculate P(+d) initially, but it can be found as P(+d) = P(+a,+d)/P(+a|+d). Using this we find P(+d) = 0.75. The simulation verifies this value as well.

7

It is impractical to attempt to receive a result from Monte Carlo simulation of the result for $P(-j, -m, b, e)$ because the event is very rare. Thus we will use likelihood weighting to calculate $P(-j \mid -m, +b, +e)$ and $P(-m \mid +b, +e)$ so we may calculate $P(-j, -m, +b, +e)$ as $P(-j, -m, +b, +e) = P(-j \mid -m, +b, +e) * P(-m \mid +b, +e) * P(+b) * P(+e)$. We don't need to estimate $P(+b)$ and $P(+e)$ because they are given.

```python
# likelihood weighting in Bayes Networks
# Caden Kroonenberg, 2022-04-03
# See: https://artint.info/2e/html/ArtInt2e.Ch8.S6.SS4.html
from random import random

TRIALS = 10**6

vars = {
    'B': [0.001],             # [P(B)]
    'E': [0.002],             # [P(E)]
    'A': [[0.001, 0.29],      # [[P(A | -B, -E), P(A | -B, +E)],
          [0.95, 0.98]],      #  [P(A | +B, -E), P(A | +B, +E)]]
    'J': [0.01, 0.95],        # [P(J | -A), P(J | +A)]
    'M': [0.01, 0.70]         # [P(M | -A), P(M | +A)]
}

def likelihood_weighting(B, e, Q, n):
    counts = [0,0] # [False count, True count]
    for trial in range(1,n+1):
        sample = {}
        weight = 1
        for x in B:
            if x in e:
                v = e[x]
                sample[x] = v
                if x == 'B' or x == 'E':
                    weight *= (B[x][0]) if sample[x] else (1 - B[x][0])
                elif x == 'A':
                    weight *= (B[x][sample['B']][sample['E']]) if sample[x] else (1 - B[x][sample['B']][sample['E']])
                elif x == 'J' or x == 'M':
                    weight *= (B[x][sample['A']]) if sample[x] else (1 - B[x][sample['A']])
            else:
                if x == 'A':
                    sample[x] = (random() < B[x][sample['B']][sample['E']])
                elif x == 'J' or x == 'M':
                    sample[x] = (random() < B[x][sample['A']])
                elif x == 'B' or x == 'E':
                    sample[x] = (random() < B[x][0])
        v = sample[Q]
        counts[v] += weight
    return [round(x / sum(counts), 3) for x in counts]
```

```python
# P(-J, -M, B, E) = P(-J | -M, B, E)P(-M | B, E)P(B)P(E)

# P(B)
p_b = vars['B'][0]

# P(E)
p_e = vars['E'][0]

# P(-J | -M, B, E)
e = {
    'M': False,
    'B': True,
    'E': True
}
P_j = likelihood_weighting(vars, e, 'J', TRIALS)
print('P(+J | -M, B, E): {}'.format(P_j[0]))
p_j = P_j[0]

# P(-M | B, E, A)
e = {
    'B': True,
    'E': True
}
P_m = likelihood_weighting(vars, e, 'M', TRIALS)
print('P(+M | B, E): {}'.format(P_m[0]))
p_m = P_m[0]

joint_p = p_j*p_m*p_b*p_e

print('P(-J, -M, B, E) = {}'.format(joint_p))
```

Output:

```
➜  ps8 git:(main) ✗ python3 likelihood_weighting.py
P(+J | -M, B, E): 0.109
P(+M | B, E): 0.314
P(-J, -M, B, E) = 6.8452e-08
```