AI
Assignment III

Joseph Pritchard - Adelaide University

June 8, 2020

Exact Inference

To perform exact inference we seek to express our queries in terms of the given conditional probability tables (CPTs). This is achieved noting

$$\begin{aligned} P(a|x_1,...,x_i) &= \frac{P(a,x_1,...,x_i)}{P(x_1,...,x_i)} \\ &= \frac{\sum_{y_i=y_1}^{y_j} P(a,x_1,...,x_i,y_1,...,y_j)}{\sum_{a}\sum_{y_i=y_1}^{y_j} P(a,x_1,...,x_i,y_1,...,y_j)} \end{aligned} \tag{1}$$

where a is the value (True or false) of interest, $x_1,...x_i$ are the i conditions of interest and $y_i,...,y_j$ are the j other variables specified in our CPTs. The joint probability distributions may be broken up in the usual way by observing our Bayesian Network tree diagram. Explicit expressions for the joint probability distributions in terms of our CPTs are implemented using MATLAB software the code for which is attached at the back. The results for each query are then:

Event	**Probability True**
$P(Sick\|L = True, D = True)$	0.10437
$P(Sick\|D = False)$	0.45345
$P(Pub\|L = False)$	0.27508
$P(Pub\|L = False, D = True)$	0.45699

Note Probabilities for false events are the complements to the probability of true events.

Approximate Inference

Approximate inference is performed by generating 5 random true or false values in line with the CPTs given in the Bayesian tree Network provided in the assignment sheet.

Random samples are produced via a uniform random number generator in MATLAB. From the entire set of samples, the subset satisfying the conditions of the query is extracted. Within this subset, the number of true events for the given query is counted and then divided by the total size of the subset. This process is carried out for 100 samples and 200 samples. The relevant code is attached at the back. The results are:

Event	**Sample Size**	**Probability True**
$P(Pub\|L = False)$	100	0.1600
$P(Pub\|L = False)$	200	0.2727
$P(Pub\|L = False, D = True)$	100	0.7000
$P(Pub\|L = False, D = True)$	200	0.5909

Note Probabilities for false events are the complements to the probability of true events.

EXACT INFERENCE METHOD

```
close all
clear all
%Calculate quantities

%Find normalisation
NORM = 0;
for P = 0:1
    for H = 0:1
        for L = 0:1
            for D = 0:1
                for S = 0:1
                    NORM = NORM + P_L_H(L,H) * P_D_H(D,H) *
 P_H_SP(H,S,P) * P_S(S) * P_P(P);
                end
            end
        end
    end
end

%Sick given L=T and D=T
SUM = 0;
S = 1;
L = 1;
D = 1;
%Sum over P
for P = 0:1
    %Sum over H
    for H = 0:1
        SUM = SUM + P_L_H(L,H) *  P_D_H(D,H) * P_H_SP(H,S,P) * P_S(S)
 * P_P(P);
    end
end
S = 0;
NORM = 0;
for P = 0:1
    %Sum over H
    for H = 0:1
        NORM = NORM + P_L_H(L,H) *  P_D_H(D,H) * P_H_SP(H,S,P) *
 P_S(S) * P_P(P);
    end
end

SUM = SUM/(SUM + NORM);
disp("Probability Sick given Lecture and Doctor: " + SUM)

%Sick given D=F
SUM = 0;
S = 1;
D = 0;
%Sum over P
```

```
for P = 0:1
    %Sum over H
    for H = 0:1
        %Sum over L
        for L = 0:1
            SUM = SUM + P_L_H(L,H) * P_D_H(D,H) * P_H_SP(H,S,P) *
 P_S(S) * P_P(P);
        end
    end
end
NORM = 0;
S = 0;
for P = 0:1
    %Sum over H
    for H = 0:1
        %Sum over L
        for L = 0:1
            NORM = NORM + P_L_H(L,H) * P_D_H(D,H) * P_H_SP(H,S,P) *
 P_S(S) * P_P(P);
        end
    end
end
SUM = SUM/(SUM + NORM);
disp("Probability Sick given not Doctor: " + SUM)

%Pub given not Lecture
SUM = 0;
P = 1;
L = 0;
%Sum over P
for S = 0:1
    %Sum over H
    for H = 0:1
        %Sum over L
        for D = 0:1
            SUM = SUM + P_L_H(L,H) * P_D_H(D,H) * P_H_SP(H,S,P) *
 P_S(S) * P_P(P);
        end
    end
end
P = 0;
NORM = 0;
for S = 0:1
    %Sum over H
    for H = 0:1
        %Sum over L
        for D = 0:1
            NORM = NORM + P_L_H(L,H) * P_D_H(D,H) * P_H_SP(H,S,P) *
 P_S(S) * P_P(P);
        end
    end
end
SUM = SUM/(SUM + NORM);
disp("Probability Pub given not Lecture: " + SUM)
```

```
%Pub given L=F and D=T
SUM = 0;
P = 1;
L = 0;
D = 1;
%Sum over P
for S = 0:1
    %Sum over H
    for H = 0:1
        SUM = SUM + P_L_H(L,H) * P_D_H(D,H) * P_H_SP(H,S,P) * P_S(S) *
 P_P(P);
    end
end
P = 0;
NORM = 0;
for S = 0:1
    %Sum over H
    for H = 0:1
        NORM = NORM + P_L_H(L,H) * P_D_H(D,H) * P_H_SP(H,S,P) * P_S(S)
 * P_P(P);
    end
end
SUM = SUM/(SUM + NORM);
disp("Probability Pub given not Lecture and Doctor: " + SUM)


%Define CPTs
function result = P_S(S)
    if(S == 1)
       result = 0.05;
    else
        result = 0.95;
    end
end

function result = P_P(P)
    if(P == 1)
       result = 0.1;
    else
        result = 0.9;
    end
end

function result = P_H_SP(H,S,P)
    if(S == 1 && P == 1)
        if(H == 1)
            result = 0.95;
        else
            result = 0.05;
        end
    elseif(S == 1 && P == 0)
        if(H == 1)
            result = 0.4;
```

```
        else
            result = 0.6;
        end
    elseif(S == 0 && P == 1)
        if(H == 1)
            result = 0.9;
        else
            result = 0.1;
        end
    else
        if(H == 1)
            result = 0.1;
        else
            result = 0.9;
        end
    end
end

function result = P_L_H(L, H)
    if(H == 1)
        if(L == 1)
            result = 0.3;
        else
            result = 0.7;
        end
    else
        if(L == 1)
            result = 0.8;
        else
            result = 0.2;
        end
    end
end

function result = P_D_H(D, H)
    if(H == 1)
        if(D == 1)
            result = 0.6;
        else
            result = 0.4;
        end
    else
        if(D == 1)
            result = 0.01;
        else
            result = 0.6;
        end
    end
end

Probability Sick given Lecture and Doctor: 0.10437
Probability Sick given not Doctor: 0.045345
Probability Pub given not Lecture: 0.27508
Probability Pub given not Lecture and Doctor: 0.45699
```

Published with MATLAB® R2018b

```
%%APPROXIMTE INFERENCE METHOD

clear all
rng shuffle

N = 200; %Set sample number
P = zeros(1,N);
S = zeros(1,N);
H = zeros(1,N);
L = zeros(1,N);
D = zeros(1,N);


pos = 0;
neg = 0;

for i = 1:N
    r = rand(1,1);
    if( r < 0.05 )
        S(i) = 1;
    end
    r = rand(1,1);
    if( r < 0.1 )
        P(i) = 1;
    end

    if( S(i) == 1 && P(i) == 1 )
         r = rand(1,1);
         if( r < 0.95 )
             H(i) = 1;
         end
    elseif( S(i) == 1 && P(i) == 0 )
         r = rand(1,1);
         if( r < 0.4 )
             H(i) = 1;
         end
    elseif( S(i) == 0 && P(i) == 1 )
         r = rand(1,1);
         if( r < 0.9 )
             H(i) = 1;
         end
    elseif( S(i) == 0 && P(i) == 0 )
         r = rand(1,1);
         if( r < 0.1 )
             H(i) = 1;
         end
    end

    if( H(i) == 1 )
         r = rand(1,1);
         if( r < 0.3 )
             L(i) = 1;
```

```
            end
            r = rand(1,1);
            if( r < 0.6 )
                D(i) = 1;
            end
        else
            r = rand(1,1);
            if( r < 0.8 )
                L(i) = 1;
            end
            r = rand(1,1);
            if( r < 0.01 )
                D(i) = 1;
            end
        end

        if( P(i) == 1 && L(i) == 0 && D(i) == 1 )
            pos = pos + 1;
        end
        if( P(i) == 0 && L(i) == 0 && D(i) == 1 )
            neg = neg + 1;
        end
    end

    disp((pos)/(pos + neg))
```

```
    0.4706
```

Published with MATLAB® R2018b