# University of Missouri-Columbia / College of Engineering

**CS 8750: Artificial Intelligence II**

# Programming Assignment #1

Adil A. Al-Azzawi
Chanmann Lim
Fernando Torre

## 1. Introduction

The ability to detect persons who have been drinking alcohol has been an ongoing struggle since the inception of alcoholic drinks themselves. Despite advances in the 20th century, it is yet to be seen for an application to be able to deduce drunkenness from data alone. In this assignment, we attempt to construct Bayesian networks that can achieve this goal.

There are 5 random variables:

- Pd: drink or not. Domain {+, -}
- Xb: breathing rate. Domain {H, M, L}
- Xh: heart rate. Domain {H, M, L}
- Xt: skin temperature. Domain {H, M, L}
- Xa: ambulation status. Domain {Fast, Slow, Stationary}

Pd is our target variable, which we will attempt to derive under different combinations of the other four variables. To test the soundness of the networks; we will test 10 queries, in which we try to determine Pd from a given set of evidences.

1. Xb=H, Xh=H, Xt=H, Xa=Stationary
2. Xb=H, Xh=M, Xt=M, Xa=Fast
3. Xb=H, Xh=M, Xt=L, Xa=Slow
4. Xb=M, Xh=M, Xt=M
5. Xb=M, Xh=L, Xt=M
6. Xb=H, Xt=L, Xa=Slow
7. Xb=L, Xt=L, Xa=Fast
8. Xb=L, Xt=M
9. Xb=L, Xt=H
10. Xb=M

## 2. BN#1

We start our exploration with a naïve Bayesian network, flowing from Pd as the root cause, to Xb, Xh, and Xt as the effects.

### 2.1. Formula derivations

#### 2.2.1. Queries 1-5

To answer the first five queries with this network, we must calculate $\mathrm{P(Pd = +)}$ for given evidences Xb, Xh, Xt with known values which we'll call A, B, and C respectively

$$\mathrm{P(Pd = +|Xb = A, Xh = B, Xt = C) = \frac{P(Xb = A, Xh = B, Xt = C|Pd = +)P(Pd = +)}{P(Xb = A, Xh = B, Xt = C)}}$$

Notice that Xb, Xh, Xt are independent given Pd, but are not when Pd is unknown. Therefore we can expand $\mathrm{P(Xb = A, Xh = B, Xt = C)}$ into

$$\begin{aligned}
&\mathrm{P(Xb = A, Xh = B, Xt = C)} \\
&\quad = \mathrm{P(Xb = A, Xh = B, Xt = C|Pd = +)P(Pd = +) + P(Xb = A, Xh = B, Xt = C|Pd = -)P(Pd = -)} \\
&\quad = \mathrm{\big(P(Xb = A|Pd = +)P(Xh = B|Pd = +)P(Xt = C|Pd = +)\big)P(Pd = +)} \\
&\quad + \mathrm{\big(P(Xb = A|Pd = -)P(Xh = B|Pd = -)P(Xt = C|Pd = -)\big)P(Pd = -)}
\end{aligned}$$

Finally,

$$\begin{aligned}
&\mathrm{P(Pd = +|Xb = A, Xh = B, Xt = C)} \\
&= \mathrm{\frac{\big(P(Xb = A|Pd = +)P(Xh = B|Pd = +)P(Xt = C|Pd = +)\big)P(Pd = +)}{\big(P(Xb = A|Pd = +)P(Xh = B|Pd = +)P(Xt = C|Pd = +)\big)P(Pd = +) + \big(P(Xb = A|Pd = -)P(Xh = B|Pd = -)P(Xt = C|Pd = -)\big)P(Pd = -)}}
\end{aligned}$$

#### 2.2.2. Queries 6-9

In the next four queries, two nodes of the Bayesian network are known, For this derivation, well use A and B to denote the respective variable holding the query value. Thus, for query 6, $\mathrm{P(B)}$ shall represent$\mathrm{P(Xt = L)}$, while for query 8 it shall represent $\mathrm{P(Xh = M)}$. We apply the same steps as in the previous derivation.

$$\begin{aligned}
\mathrm{P(Pd = +|A, B)} &= \mathrm{\frac{P(A, B|Pd = +)P(Pd = +)}{P(A, B)} = \frac{P(A, B|Pd = +)P(Pd = +)}{P(A, B|Pd = +)P(Pd = +) + P(A, B|Pd = -)P(Pd = -)}} \\
&= \mathrm{\frac{\big(P(A|Pd = +)P(B|Pd = +)\big)P(Pd = +)}{\big(P(A|Pd = +)P(B|Pd = +)\big)P(Pd = +) + \big(P(A|Pd = -)P(B|Pd = -)\big)P(Pd = -)}}
\end{aligned}$$

### 2.2.3. Query 10

The last query is comparatively straightforward:

$$P(Pd=+|Xb=M)=\frac{P(Xb=M|Pd=+)P(Px=+)}{P(Xb=M)}$$

$$=\frac{P(Xb=M|Pd=+)P(Px=+)}{P(Xb=M|Pd=+)P(Px=+)+P(Xb=M|Pd=-)P(Px=-)}$$

## 2.2. Pseudo code

For the implementation of this network, we will store all the conditional probabilities in a series of arrays. Thenceforth, the main focus of the program becomes accessing the correct values to plug into the corresponding formula, and thus, answer the query.

**Step.1:** Setup the arrays values // using Bn1 function
p_pd = [0.13, 0.87];
p_xb_pb_plus = [ 0.64 , 0.22 ,0.14]; //breathing rate. Domain
p_xb_pb_neg = [ 0.09 , 0.42 ,0.49];
p_hx_pd_plus = [0.54 ,0.31,0.15];// heart rate. Domain
p_hx_pd_neg = [0.12 ,0.42, 0.46];
p_xt_pd_plus = [0.73 ,0.18, 0.09];// skin temperature. Domain
p_xt_pd_neg = [0.03 ,0.76,0.21]
**Step 2:** **For** i=1 **to** 10 // find the query from 1 to 10
**Step 2.1:** Get the query features // five random variable values (Xb,Xh,Xt='High or …,Pd='+/-'
**Step 2.2:** For each variable // (Xa,Xh,Xt,Pd) and pd='+'
**Step 2.2.1:** Find the specific symbol for each variable // X_Variable='H' or 'M'or 'L'
**Step 2.2.1.1:** Enter the random variable text features. // using feature function
**Step 2.2.1.2**: Check the input text if it valid for the feature or not.
**IF** input in 'High 'or 'Medium 'or 'Low' …then Flag='TRUE'
E**lse** Flag='False'
**Step 2.2.1.3:** **IF** the Flag='True' then go to **step.2.2**
**Else** go to **Step.2.2.1.1**
**Step 2.2.2:** **IF** pd='+' **then**
pb= pb_plus(1,1) //P(pd='+'))
**Else** pb= pb_plus(1,2) //P(pd='-'))
**Step 2.2.3:** Find the prob. Value for each variable using bn1 function but with pd='+'.
**Step 2.2.3.1:** **For** each variable **=**1 to 4 by using each variable symbol
**Step 2.2.3.2:** **IF** (XB =='-') // the variable is not given
Xb=1; go to **Step 2.2.3.4**
**Else** go to **Step.2.2.3.3**
**Step 2.2.3.3:** Depending on the specific letter of each variable go to **Step.1**
**Step 2.2.3.4:** **IF** (Xh =='-') // the variable is not given

Xh=1; go to **Step 2.2.3.6**
**Else** go to **step.2.2.3.5**
**Step 2.2.3.5:** Depending on the specific letter of each variable go to **Step.1**
**Step 2.2.3.6: IF** (Xt =='-') // the variable is not given
Xt=1; go to **Step 2.2.3.8**
**Else** go to **Step.2.2.3.7**
**Step 2.2.3.7:** Depending on the specific letter of each variable go to **Step.1**
**Step 2.2.3:** Find the prob. Value for each variable using bn1 function but with **pd='-'**.
**Step.2.3:** Find the first and the second part of the equation using Eqs in **sec.2.1**.
**Step.2.4:** Do the printing of the result.
**Step 3:** Next loop.
**Step 4:** End.

## 2.3. Matlab code

What follows is the actual code put into Matlab to run this network. The code is divided into two parts: the main program, containing the user interface (inputting queries, outputting results), and the encoding of the Bayesian network.

### 2.2.4. Main program code:

```
%=========================================================================
% CS 8750 - Artificial Intelligence II...
% Programming Assignment #1 ...
% Adil Al-Azzawi ... ECE
% Chanmann Lim    ... CS
% Fernando Torre ... CS
%=========================================================================
%%
close all; clc; clear;
t=0;
while (t ~= 1)
    clc;
    display('_______________________________________________________');
    display(' ');
    display('               Programming Assignment #1               ');
    display('_______________________________________________________');
    display('_______________________________________________________');
    display(' ');
    display('             1: For Bayesian Networks No.1             ');
    display('             2: For Bayesian Networks No.2             ');
    display('             3: For Exit                               ');
    display('_______________________________________________________');
    display(' ');
    x = input(' Select the Baysian Network that you want to implement : ');
    display(' ');
    if (x ==1)
        clc;
        BayNet_1(x)
        t=0;
    elseif (x==2)
        clc;
        BayNet_2(x)
        t=0;
```

```
        else
            t=1;
        end
    end
```

### 2.2.5. BN#1 function code:

```
function [ ] = BayNet_1(ch)
%% Using Bayesian Network No.1...
%% --------------------- Compute Queries No.1 to 10 ------------------------
for i=1:10
    display('_______________________________________________________');
    display('_______________________________________________________');
display('                                                       ');
display('                  Bayesian Network No.1                 ');
display('_______________________________________________________');
display('_______________________________________________________');
fprintf('Compute Query No: %d\n', i);
display(' ');
    %%------------------------ Bayesia Network #1--------------------
    % Find the query features (five random variables)..
        [xb,xh,xt,xa,pd] = features(ch );
        % From Text to Prob. symbol...
        [c1,~] = pdf2(xb);
        [c2,~] = pdf2(xh);
        [c3,~] = pdf2(xt);
        %[c4,~] = pdf2(xa);
        [c5_post,c5_neg] = pdf2(pd);
        % Find the prob. of given Pd...
            if (c1~='-')
                [p_xb] = bn1(c1,' ',' ',' ',c5_post)
            else
                p_xb=1;
            end
            if (c1~='-')
                [p_xh] = bn1(' ',c2,' ',' ',c5_post)
            else
                p_xh=1;
            end
            if (c1~='-')
                [p_xt] = bn1(' ',' ',c3,' ',c5_post)
             else
                p_xt=1;
            end
        %[p_xa] = bn1(' ',' ',' ',c4,c5_post)
        [pd_plus] = bn1(' ',' ',' ',' ',c5_post)
        % Find the prob. of not given Pd...
        p_xb_not_pd= bn1(c1,' ',' ',' ',c5_neg)
        p_xh_not_pd= bn1(' ',c2,' ',' ',c5_neg)
        p_xt_not_pd= bn1(' ',' ',c3,' ',c5_neg)
        pd_neg_pd= bn1(' ',' ',' ',' ',c5_neg)
        %
        %-------------------- P(xb,xh,xt/pd(+))P(pd(+) -------------
        p_xbxhxt_pd=p_xb*p_xh*p_xt*pd_plus
        %
        %---------------------- p(xb,xh,xt) ------------------------
p_xbxhxt=p_xbxhxt_pd+(p_xb_not_pd*p_xh_not_pd*p_xt_not_pd*pd_neg_pd)
        %
        %------------------Final Result P(Pd/Xb,Xh,Xt) -------------
        result=p_xbxhxt_pd/ p_xbxhxt
        Print(i,c1,c2,c3,'-',result,p_xb,p_xh,p_xt,0)
        input('Press enter to continue...','s');
```

```
        close all;clc;
end
end

function [ xb,xh,xt,xa,pd] = features(ch )
fla=0;clc;
    while (fla ~= 1)
        clc;
        display('___________________________________');
        display('                                   ');
        display('       Breathing Rate Domain       ');
        display('___________________________________');
        display('___________________________________');
        display('          1: H for High...');
        display('          2: M for Medium...');
        display('          3: L for Low...');
        display('          4: X for Non...');
        display('___________________________________');
        display('___________________________________');
        xb = input(' Enter the breathing rate domain (xb):','s');
        display(' ');
        [fla] = check('b',xb);
        if (fla == 1)
            break;
        else
           msgbox('Invalid Value', 'Error','error');
        end
    end
    fla=0;clc;
    while (fla ~=1)
        clc;
        fprintf('  Baysian Network No.(%d%s\n', ch,')');
        display('___________________________________');
        display('                                   ');
        display('         Heart Rate Domain         ');
        display('___________________________________');
        display('___________________________________');
        display('         1: H for High...');
        display('         2: M for Medium...');
        display('         3: L for Low...');
        display('         4: X for Non...');
        display('___________________________________');
        display('___________________________________');
        display('                                   ');
        xh = input(' Enter Heart rate domain (xh) : ','s');
        display(' ');
        [fla] = check('h',xh);
        if (fla == 1)
            break;
        else
           msgbox('Invalid Value', 'Error','error');
           fla=0;
        end
    end
    fla=0;clc;
    while (fla ~= 1)
        clc;
        fprintf('  Baysian Network No.(%d%s\n', ch,')');
        display('___________________________________');
        display('                                   ');
        display('      Skin Temperature Domain      ');
        display('___________________________________');
        display('___________________________________');
```

```
    display('          1: H for High...          ');
    display('          2: M for Medium...        ');
    display('          3: L for Low...           ');
    display('          4: X for Non...           ');
    display('___________________________________');
    display('___________________________________');
    display('                                   ');
    xt = input(' Enter skin temperature domain (xt): ','s');
    display(' ');
    [fla] = check('t',xt);
    if (fla == 1)
        break;
    else
       fla=0;
       msgbox('Invalid Value', 'Error','error');
    end
end
fla=0;clc;
while fla ~=1
    if (ch==1)
        xa='-';
        break;
    else
        clc;
        fprintf('  Baysian Network No.(%d%s\n', ch,')');
        display('___________________________________');
        display('                                   ');
        display('      Ambulation Status Domain     ');
        display('___________________________________');
        display('___________________________________');
        display('          1: F for Fast...');
        display('          2: S for Slow...');
        display('          3: St for Stationart...');
        display('          4: X for Non...');
        display('___________________________________');
        display('___________________________________');
        display('                                   ');
        xa = input(' Enter ambulation status domain (xa) : ','s');
        display(' ');
        [fla] = check('a',xa);
        if (fla == 1)
            break;
        else
           fla=0;
           msgbox('Invalid Value', 'Error','error');
        end
    end
end
fla=0;clc;
while (fla ~=1)
    clc;
    fprintf('  Bayesian Network No.(%d%s\n', ch,')');
    display('___________________________________');
    display('                                   ');
    display('       Personal Information        ');
    display('___________________________________');
    display('___________________________________');
    display('          1: + for Positive...');
    display('          2: - for Negative...');
    display('___________________________________');
    display('___________________________________');
    display('                                   ');
    pd = input(' Does the person drink or not (pd): ','s');
```

```
            display(' ');
            [fla] = check('d',pd);
            if (fla == 1)
                break;
            else
               fla=0;
               msgbox('Invalid Value', 'Error','error');
            end
        end
    end

function [net] = bn1(xb,xh,xt,xa,pd)
%%
% the prob. of the drink or not. Domain {+, -}
     p_pd = [0.13, 0.87];
     % the prob. of the  breathing rate. Domain
     p_xb_pb_plus = [ 0.64 , 0.22 ,0.14];
     p_xb_pb_neg = [ 0.09 , 0.42 ,0.49];
     % the Prob. of the plus heart rate. Domain
     p_hx_pd_plus = [0.54 ,0.31,0.15];
     p_hx_pd_neg = [0.12 ,0.42, 0.46];
     % the Prob. of the skin temperature. Domain
     p_xt_pd_plus = [0.73 ,0.18, 0.09];
     p_xt_pd_neg = [0.03 ,0.76,0.21];
     % the Prob. Xa: ambulation status. Domain {Fast, Slow, Stationary}
     p_xa = [0.21 ,0.22, 0.57];
%% Using BN#1....
% Pd prob...
    if pd == 'p'
      c = p_pd(1);
    else
      c = p_pd(2);
    end
% prob. of Xb: breathing rate. Domain {H, M, L}
    if (pd == 'p')&& (xb == 'h')
         c = p_xb_pb_plus(1);
    end
    if (pd == 'p')&&(xb == 'm')
         c = p_xb_pb_plus(2);
    end
    if (pd == 'p')&&(xb == 'l')
         c = p_xb_pb_plus(3);
    end
    if (pd == 'n')&& (xb == 'h')
         c =  p_xb_pb_neg(1);
    end
    if (pd == 'n')&&(xb == 'm')
         c =  p_xb_pb_neg(2);
    end
    if (pd == 'n')&&(xb == 'l')
         c =  p_xb_pb_neg(3);
    end
% prob. of Xh: heart rate. Domain {H, M, L}
    if (pd == 'p')&&(xh == 'h')
         c = p_hx_pd_plus(1);
    end
    if (pd == 'p')&&(xh == 'm')
         c =p_hx_pd_plus(2);
    end
    if (pd == 'p')&&(xh == 'l')
         c = p_hx_pd_plus(3);
    end
```

```
    if (pd == 'n')&&(xh == 'h')
        c = p_hx_pd_neg(1);
    end
    if (pd == 'n')&&(xh == 'm')
        c = p_hx_pd_neg(2);
    end
    if (pd == 'n')&&(xh == 'l')
        c = p_hx_pd_neg(3);
    end
 % prob. of Xt: skin temperature. Domain {H, M, L}
    if (pd == 'p')&&(xt == 'h')
        c = p_xt_pd_plus(1);
    end
    if (pd == 'p')&&(xt == 'm')
        c =p_xt_pd_plus(2);
    end
    if (pd == 'p')&&(xt == 'l')
        c = p_xt_pd_plus(3);
    end
    if (pd == 'n')&&(xt == 'h')
        c = p_xt_pd_neg(1);
    end
    if (pd == 'n')&&(xt == 'm')
        c = p_xt_pd_neg(2);
    end
    if (pd == 'n')&&(xt == 'l')
        c = p_xt_pd_neg(3);
    end
 % Prob. of Xa: ambulation status. Domain {Fast, Slow, Stationary}
 if xa == 'x'
     c=0;
 else
     if xa == 'f'
         c = p_xa(1);
     end
     if xa == 'w'
         c =p_xa(2);
     end
     if xa == 't'
         c =p_xa(3);
     end
 end

net = c;
end

function [O1,O2] = pdf2( I )
%% Pd: drink or not. Domain {+, -}
    if  strcmp(I,'+')
        O1='p';O2='n';
    elseif strcmp(I,'-')
        O1='p';O2='p';
    end
%% Xb: breathing rate. Domain {H, M, L}
 % Xh: heart rate. Domain {H, M, L}
 % Xt: skin temperature. Domain {H, M, L}
    if  strcmp(I,'H') || strcmp(I,'h')
        O1='h';O2=' ';
    elseif  strcmp(I,'M') || strcmp(I,'m')
        O1='m';O2=' ';
    elseif  strcmp(I,'L') || strcmp(I,'l')
        O1='l';O2=' ';
```

```
    elseif strcmp(I,'X')|| strcmp(I,'x')
        O1='-';O2=' ';
    end
%% Xa: ambulation status. Domain { Fast, Slow, Stationary}
    if  strcmp(I,'F') || strcmp(I,'f')||strcmp(I,'Fast') ||
strcmp(I,'fast')||strcmp(I,'FAST')
        O1='f';O2=' ';
    elseif  strcmp(I,'s')|| strcmp(I,'S')||strcmp(I,'SLOW') || strcmp(I,'slow')||
strcmp(I,'Slow')
        O1='w';O2=' ';
    elseif  strcmp(I,'St') || strcmp(I,'st')|| strcmp(I,'ST')
        O1='t';O2=' ';
    elseif strcmp(I,'n')|| strcmp(I,'N')
        O1='-';O2=' ';
    end
 end

function [flag] = check(x,I)
%UNTITLED Summary of this function goes here
%   Detailed explanation goes here
   % check the Xb vaild values...
   if (x =='b')||(x =='h')||(x =='t')
       if  strcmp(I,'H') || strcmp(I,'h')
            flag=1;
       elseif  strcmp(I,'M') || strcmp(I,'m')
            flag=1;
       elseif  strcmp(I,'L') || strcmp(I,'l')
            flag=1;
       elseif strcmp(I,'x')
            flag=1;
        else
            flag=0;
       end
   end
   % check the Pd vaild values...
   if (x =='d')
       if  strcmp(I,'+')
            flag=1;
       elseif  strcmp(I,'-')
            flag=1;
        else
            flag=0;
       end
   end
   % check the Xa vaild values...
   if (x =='a')
       if  strcmp(I,'F') || strcmp(I,'f')||strcmp(I,'Fast') ||
strcmp(I,'fast')||strcmp(I,'FAST')
            flag=1;
       elseif  strcmp(I,'s')|| strcmp(I,'S')||strcmp(I,'SLOW') || strcmp(I,'slow')||
strcmp(I,'Slow')
            flag=1;
       elseif  strcmp(I,'St') || strcmp(I,'st')|| strcmp(I,'ST')
            flag=1;
       elseif strcmp(I,'X') || strcmp(I,'x')
            flag=1;
       else
            flag=0;
       end
    end
end
```

## 2.4. Query execution results

The Bayesian Network 1 queries results are shown in the next tables

1- Query number 1: $P(Pd = +| Xb = H, Xh = H, Xt = H)$

| | ID | P(Xb/Pd) | P(Xh/Pd) | P(Xh/Pd) | P(Xa) | Pd(p=+) |
|---|---|---|---|---|---|---|
| Prob. | 1 | 0.6400 | 0.5400 | 0.7300 | 0 | 0.9915 |

2- Query number 1: $P(Pd = +| Xb = H, Xh = M, Xt = M)$

| | ID | P(Xb/Pd) | P(Xh/Pd) | P(Xh/Pd) | P(Xa) | Pd(p=+) |
|---|---|---|---|---|---|---|
| Prob. | 2 | 0.6400 | 0.3100 | 0.1800 | 0 | 0.1567 |

3- Query number 1: $P(Pd = +| Xb = H, Xh = M, Xt = L)$

| | ID | P(Xb/Pd) | P(Xh/Pd) | P(Xh/Pd) | P(Xa) | Pd(p=+) |
|---|---|---|---|---|---|---|
| Prob. | 3 | 0.6400 | 0.3100 | 0.0900 | 0 | 0.2516 |

4- Query number 1: $P(Pd = +| Xb = M, Xh = M, Xt = M)$

| | ID | P(Xb/Pd) | P(Xh/Pd) | P(Xh/Pd) | P(Xa) | Pd(p=+) |
|---|---|---|---|---|---|---|
| Prob. | 4 | 0.2200 | 0.3100 | 0.1800 | 0 | 0.0135 |

5- Query number 1: $P(Pd = +| Xb = M, Xh = L, Xt = M)$

| | ID | P(Xb/Pd) | P(Xh/Pd) | P(Xh/Pd) | P(Xa) | Pd(p=+) |
|---|---|---|---|---|---|---|
| Prob. | 5 | 0.2200 | 0.1500 | 0.1800 | 0 | 0.0060 |

6- Query number 1: $P(Pd = +| Xb = M, Xt = M)$

| | ID | P(Xb/Pd) | P(Xh/Pd) | P(Xh/Pd) | P(Xa) | Pd(p=+) |
|---|---|---|---|---|---|---|
| Prob. | 6 | 0.2200 | 1 | 0.0900 | 0 | 0.0325 |

7- Query number 1: $P(Pd = +| Xb = L, Xt = M)$

| | ID | P(Xb/Pd) | P(Xh/Pd) | P(Xh/Pd) | P(Xa) | Pd(p=+) |
|---|---|---|---|---|---|---|
| Prob. | 7 | 0.1400 | 1 | 0.0900 | 0 | 0.0180 |

8- Query number 1: $P(Pd = +| Xb = L, Xh = M)$

| | ID | P(Xb/Pd) | P(Xh/Pd) | P(Xh/Pd) | P(Xa) | Pd(p=+) |
|---|---|---|---|---|---|---|
| Prob. | 8 | 0.1400 | 0.3100 | 1 | 0 | 0.0305 |

9- Query number 1: $P(Pd = +| Xb = L, Xh = H)$

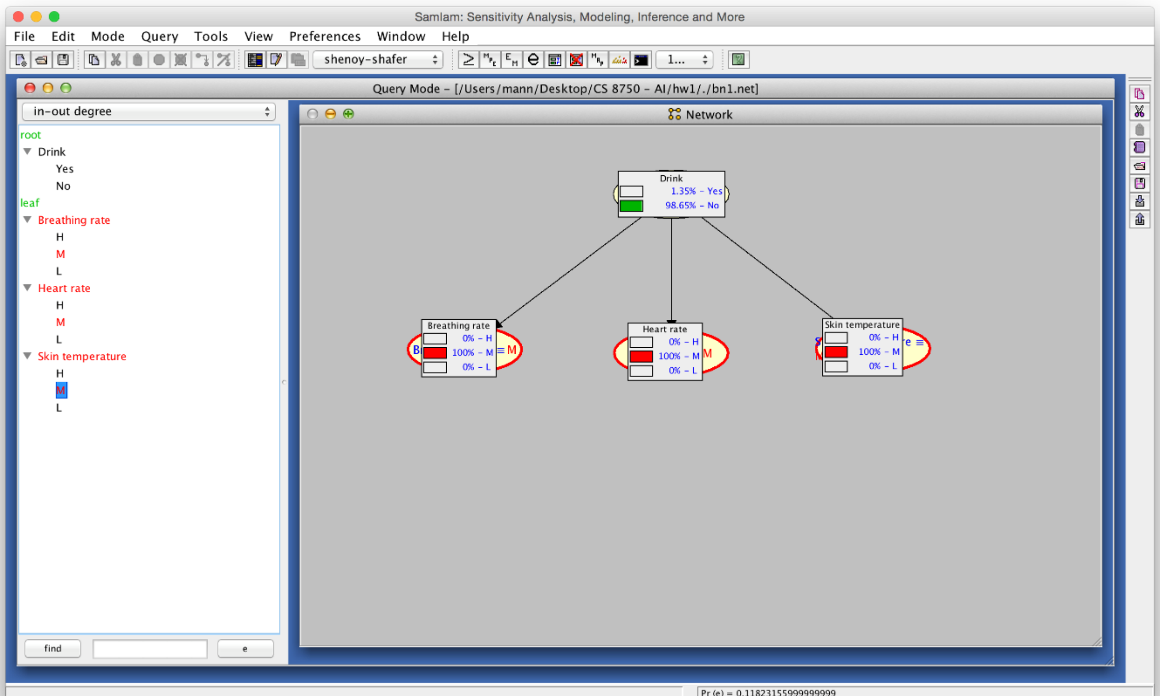
| | ID | P(Xb/Pd) | P(Xh/Pd) | P(Xh/Pd) | P(Xa) | Pd(p=+) |
|---|---|---|---|---|---|---|
| Prob. | 9 | 0.1400 | 0.5400 | 1 | 0 | 0.1612 |

10- Query number 1: $\mathrm{P(Pd = \ + \mid Xb = M)}$

| | ID | P(Xb/Pd) | P(Xh/Pd) | P(Xh/Pd) | P(Xa) | Pd(p=+) |
|---|---|---|---|---|---|---|
| Prob. | 10 | 0.2200 | 1 | 1 | 0 | 0.0726 |

## 2.5. Samlam implementation

The naïve Bayesian network was implemented explicitly in Samlam. The results were verified against the Matlab program and matched. The graph is shown below. As an example, we've selected the evidence to answer query number 4: $\mathrm{P(Pd = \ + \mid Xb = M, Xh = M, Xt = M)}$




The results of running the query using junction tree algorithm Shenoy-Shafer: $\mathrm{P(Pd = \ + \mid Xb = M, Xh = M, Xt = M) = 1.35\%}$

# 3. BN#2

We build a second Bayesian network that considers Xa. Xa is posited as an alternative cause for Xb, Xh, and Xt. It is hoped that if Xa can explain away Xb, Xh and Xt better than Pd, Pd will be use as the explanation less often.

## 3.1.Formula derivations

### 3.2.1. Queries 1-3

To answer the first three queries with this network, we must calculate $P(Pd = +)$ for Xa, Xb, Xh, X with known values. In the interest of legibility, we will denote $Pd = +$ as A, $Pd = -$ as $\neg A$, $Xa = y_1$ (Where $y_i$ is the desired value for the query) as $B$, $Xb = y_2$ as $C$, $Xh = y_3$ as $D$, and $Xt = 4$ as $E$. Thus, in each of the three queries, we are seeking

$$P(Pd = +|Xa = y_1, Xb = y_2, Xh = y_3, Xt = 4) = P(A|B,C,D,E) = \frac{P(A,B,C,D,E)}{P(B,C,D,E)} = \frac{P(C,D,E|A,B)\,P(A,B)}{P(C,D,E|B)P(B)}$$

$$= \frac{P(C,D,E|A,B)\,P(A,B)}{[P(C,D,E|B,A)P(A) + P(C,D,E|B,\neg A)P(\neg A)]P(B)}$$

$$= \frac{\big(P(C|A,B)P(D|A,B)P(E|A,B)\big)P(A)P(B)}{\big[\big(P(C|B,A)P(D|B,A)P(E|B,A)\big)P(A) + \big(P(C|B,\neg A)P(D|B,\neg A)P(E|B,\neg A)\big)P(\neg A)\big]P(B)}$$

$$= \frac{\big(P(C|A,B)P(D|A,B)P(E|A,B)\big)P(A)}{\big(P(C|B,A)P(D|B,A)P(E|B,A)\big)P(A) + \big(P(C|B,\neg A)P(D|B,\neg A)P(E|B,\neg A)\big)P(\neg A)}$$

### 3.2.2. Queries 4-5

We use the same procedure and notation as to derive the previous expression, however, we add the following: $B_f$ to denote $Xb =$ fast, $B_{sw}$ to denote $Xb =$ slow, and $B_{st}$ to denote $Xb =$ stationary. We therefore have:

$$P(Pd = +|Xb = y_2, Xh = y_3, Xt = y_4) = P(A|C,D,E) = \frac{P(A,C,D,E)}{P(C,D,E)}$$

$$= \frac{P(C,D,E|A)P(A)}{P(C,D,E|A)P(A) + P(C,D,E|\neg A)P(\neg A)}$$

We derive separately $P(C,D,E|A)$ and $P(C,D,E|\neg A)$ and later plug them into the expression

$$P(C,D,E|A) = P\big(C,D,E\big|A,B_f\big)P\big(B_f\big) + P(C,D,E|A,B_{sw})P(B_{sw}) + P(C,D,E|A,B_{st})P(B_{sw})$$

$$= \sum_{i\in\{f,sw,st\}} P(C,D,E|A,B_i)P(B_i) = \sum_{i\in\{f,sw,st\}} \big(P(C|A,B_i)P(D|A,B_i)P(E|A,B_i)\big)P(B_i)$$

$$= \sum_{i\in\{f,sw,st\}} \prod_{j\in\{A,D,E\}} P(j|A,B_i)\,P(B_i)$$

$$P(C,D,E|\neg A) = P(C,D,E|\neg A,B_f)P(B_f) + P(C,D,E|\neg A,B_{sw})P(B_{sw}) + P(C,D,E|\neg A,B_{st})P(B_{sw})$$
$$= \sum_{i\in\{f,sw,st\}} P(C,D,E|\neg A,B_i) = \sum_{i\in\{f,sw,st\}} \big(P(C|\neg A,B_i)P(D|\neg A,B_i)P(E|\neg A,B_i)\big)P(B_i)$$
$$= \sum_{i\in\{f,sw,st\}} \prod_{j\in\{A,D,E\}} P(j|\neg A,B_i)\,P(B_i)$$

Finally,

$$P(A|C,D,E)$$
$$= \frac{\left(\sum_{i\in\{f,sw,st\}}\prod_{j\in\{A,D,E\}} P(j|A,B_i)\,P(B_i)\right)P(A)}{\left(\sum_{i\in\{f,sw,st\}}\prod_{j\in\{A,D,E\}} P(j|A,B_i)\,P(B_i)\right)P(A) + \left(\sum_{i\in\{f,sw,st\}}\prod_{j\in\{A,D,E\}} P(j|\neg A,B_i)\,P(B_i)\right)P(\neg A)}$$

### 3.2.3. Queries 6-7

We use the same procedure and notation as before.

$$P(Pd=+|Xa=y_1,Xb=y_2,Xt=4) = P(A|B,C,E) = \frac{P(A,B,C,E)}{P(B,C,E)} = \frac{P(C,E|A,B)P(A,B)}{P(C,E|B)P(B)}$$
$$= \frac{P(C,E|A,B)P(A,B)}{[P(C,E|B,A)P(A) + P(C,E|B,\neg A)P(\neg A)]P(B)}$$
$$= \frac{\big(P(C|A,B)P(E|A,B)\big)P(A)P(B)}{\big[\big(P(C|B,A)P(E|B,A)\big)P(A) + \big(P(C|B,\neg A)P(E|B,\neg A)\big)P(\neg A)\big]P(B)}$$

### 3.2.4. Queries 8-9

Once again we use the same notation

$$P(Pd=+|Xb=y_2,Xh=y_3) = \frac{P(A,C,D)}{P(C,D)} = \frac{P(A,C,D,B_f) + P(A,C,D,B_{sw}) + P(A,C,D,B_{st})}{P(C,D)}$$

Let us expand $P(C,D)$ separately

$$P(C,D)$$
$$= P(C,D,A,B_f) + P(C,D,A,B_{sw}) + P(C,D,A,B_{st})$$
$$+P(C,D,\neg A,B_f) + P(C,D,\neg A,B_{sw}) + P(C,D,\neg A,B_{st})$$
$$= P(C,D|A,B_f)P(A,B_f) + P(C,D|A,B_{sw})P(A,B_{sw}) + P(C,D|A,B_{st})$$
$$+P(C,D|\neg A,B_f)P(\neg A,B_f) + P(C,D|\neg A,B_{sw})P(\neg A,B_{sw}) + P(C,D|\neg A,B_{st})P(\neg A,B_{st})$$

$$= \sum_{i \in \{f,sw,st\}} \big(P(C,D|A,B_i)P(A,B_i) + P(C,D|\neg A,B_i)P(\neg A,B_i)\big) = \sum_{i \in \{f,sw,st\}} \sum_{j \in \{A,\neg A\}} P(C,D|j,B_i)P(j,B_i)$$

$$= \sum_{i \in \{f,sw,st\}} \sum_{j \in \{A,\neg A\}} \big(P(C|j,B_i)P(D|j,B_i)\big)P(j)P(B_i)$$

We note that

$$P\big(C,D,A,B_f\big) + P(C,D,A,B_{sw}) + P(C,D,A,B_{st}) = \sum_{i \in \{f,sw,st\}} P(C,D|A,B_i)P(A,B_i)$$

$$= \sum_{i \in \{f,sw,st\}} \big(P(C|A,B_i)P(D|A,B_i)\big)P(A)P(B_i)$$

Therefore

$$\frac{P(A,C,D)}{P(C,D)} = \frac{\sum_{i \in \{f,sw,st\}} \big(P(C|A,B_{\mathrm{i}})P(D|A,B_i)\big)P(A)P(B_i)}{\sum_{i \in \{f,sw,st\}} \sum_{j \in \{A,\neg A\}} \big(P(C|j,B_i)P(D|j,B_i)\big)P(j)P(B_i)}$$

### 3.2.5. Query 10

Once more, the same notation is used.

$$P(A|C) = \frac{P(A,C)}{P(C)} = \frac{P\big(A,C,B_f\big) + P(A,C,B_{sw}) + P(A,C,B_{st})}{P(C)}$$

We calculate $P(C)$ separately

$$P(C) = P\big(C,A,B_f\big) + P(C,A,B_{sw}) + P(C,A,B_{st}) + P\big(C,\neg A,B_f\big) + P(C,\neg A,B_{sw}) + P(C,\neg A,B_{st})$$

$$= P\big(C\big|A,B_f\big)P(A)P\big(B_f\big) + P(C|A,B_{sw})P(A)P(B_{sw}) + P(C|A,B_{st})P(A)P(S_{sw})$$

$$+ P\big(C\big|\neg A,B_f\big)P(\neg A)P\big(B_f\big) + P(C|\neg A,B_{sw})P(\neg A)P(B_{sw}) + P(C|\neg A,B_{st})P(\neg A)P(B_{st})$$

$$= \sum_{i \in \{f,sw,st\}} \big(P(C|A,B_i)P(A)P(B_i) + P(C|\neg A,B_i)P(\neg A)P(B_i)\big)$$

Therefore:

$$P(A|C) = \frac{P\big(C\big|A,B_f\big)P(A)P\big(B_f\big) + P(C|A,B_{sw})P(A)P(B_{sw}) + P(C|A,B_{st})P(A)P(S_{sw})}{\sum_{i \in \{f,sw,st\}} \big(P(C|A,B_i)P(A)P(B_i) + P(C|\neg A,B_i)P(\neg A)P(B_i)\big)}$$

### 3.2.Pseudocode

As with the previous network, we will store all the conditional and prior probabilities in a series of arrays. Thenceforth, the main focus of the program becomes accessing the correct values to plug into the corresponding formula to be able to answer the query.

**Step.1:** Setup the arrays values for BN#1 // using Bn1 function
p_pd = [0.13, 0.87]
p_xb_pb_plus = [ 0.64 , 0.22 ,0.14]; //breathing rate. Domain
p_xb_pb_neg = [ 0.09 , 0.42 ,0.49];
p_hx_pd_plus = [0.54 ,0.31,0.15];//heart rate. Domain
p_hx_pd_neg = [0.12 ,0.42, 0.46];
p_xt_pd_plus = [0.73 ,0.18, 0.09];//skin temperature. Domain
p_xt_pd_neg = [0.03 ,0.76,0.21]
**Step.2:** Setup the arrays values for BN#2 // using Bn1 function
p_xb_pdxa_plus=[0.95,0.03,0.02 ;0.77,0.19,0.04; 0.71,0.2,0.09]; //P(Xh|Pd, Xa)
p_xb_pdxa_neg=[0.87,0.11,0.02 ;0.14,0.74,0.12; 0.03,0.16,0.81];
p_xh_pdxa_plus=[0.97,0.02,0.01 ;0.76,0.2,0.04; 0.63,0.23,0.14]; //P(Xh|Pd, Xa)
p_xh_pdxa_neg=[0.92,0.07,0.01 ;0.11,0.82,0.07; 0.07,0.08,0.85];
p_xt_pdxa_plus=[0.91,0.06,0.03 ;0.54,0.36,0.1; 0.49,0.38,0.13]; //(Xt|Pd, Xa)
p_xt_pdxa_neg=[0.74,0.18,0.08 ;0.21,0.47,0.32; 0.11,0.62,0.27];
**Step 3: For** i=1 **to** 10 // find the query from 1 to 10
**Step 3.1:** Get the query features // five random variable values (Xb,Xh,Xt='High or …,Pd='+/-'
**Step 3.2:** For each variable // (Xa,Xh,Xt,Pd) and pd='+'
**Step 3.2.1:** Find the specific symbol for each variable // X_Variable='H' or 'M'or 'L'
**Step 3.2.1.1:** Enter the random variable text features. // using feature function
**Step 3.2.1.2**: Check the input text if it valid for the feature or not.
**IF** input in 'High 'or 'Medium 'or 'Low' …then Flag='TRUE'
E**lse** Flag='False'
**Step 3.2.1.3: IF** the Flag='True' then go to **step.2.2**
**Else** go to **Step.3.2.1.1**
**Step 3.3:** Find the prob. Value for each variable using bn1 function but with **pd='-'**and **pd='-'**.
**Step 3.4: For** i=1 **to** 10 computes the queries
**Step 3.4.1: IF** pd='+' **then**
pb= pb_plus(1,1) //P(pd='+'))
**Else** pb= pb_plus(1,2) //P(pd='-'))
**Step 3.4.2:** Find the prob. of given Pd='+'
[p_xb1],[p_xh1], [p_xt1], [pd_plus],[p_xa]
**Step 3.4.3:** P(variables/P(pd(+))^P(xa)
**Step 3.4.4:** Find the prob. of not given Pd='-'
[p_xb2],[p_xh2], [p_xt2], [pd_neg]
**Step 3.4.5:** P(variables/P(~pd(+))^P(xa)

**Step 3.4.6:** Find the queries from 1 to 4

**Step.3.5:** Do the printing of the result.

**Step 3:** Next loop.

**Step 4:** End.

## 3.3.Matlab code

What follows is the actual code put into matlab to run this network. The main program code is the same as for the first Bayesian network.

```
function [ ] = BayNet_2(ch)
%% using Bayesian Netwrok No.2 ...
% --------------------- Compute Queries No.1 to 3 ------------------------
 for i=1:10
    display('_______________________________________________________');
    display('_______________________________________________________');
    display('                                                       ');
    display('                  Bayesian Network No.2                ');
    display('_______________________________________________________');
    display('_______________________________________________________');
    display(' ')
    fprintf('Compute Query No: %d\n', i);
    display(' ');
    % Find the query#1 ,#2 & #3 results...
        [xb,xh,xt,xa,pd ] = features(ch);
         % From Text to Prob. values...
           [c1,~] = pdf2(xb)
           [c2,~] = pdf2(xh)
           [c3,~] = pdf2(xt)
           [c4,~] = pdf2(xa)
           [c5_post,c5_neg] = pdf2(pd)
           [pd_plus] = bn1(' ',' ',' ',' ',c5_post)
           [pd_neg] = bn1(' ',' ',' ',' ',c5_neg)
       %% Find the query #4&5 results...
       if (c4 =='-')&&(c3~='-')&&(c2~='-')
           % Find the prob. of given Pd...
           % Postive P(Pd(+))...
           c4='f';
            [p_xa_p1] = bn1(' ',' ',' ',c4,' ')
            [p_xb_p1] = bn2(c1,' ',' ',c4,c5_post)
            [p_xh_p1] = bn2(' ',c2,' ',c4,c5_post)
            [p_xt_p1] = bn2(' ',' ',c3,c4,c5_post)
           c4='w';
            [p_xa_p2] = bn1(' ',' ',' ',c4,' ')
            [p_xb_p2] = bn2(c1,' ',' ',c4,c5_post)
            [p_xh_p2] = bn2(' ',c2,' ',c4,c5_post)
            [p_xt_p2] = bn2(' ',' ',c3,c4,c5_post)
           c4='t';
            [p_xa_p3] = bn1(' ',' ',' ',c4,' ')
            [p_xb_p3] = bn2(c1,' ',' ',c4,c5_post)
            [p_xh_p3] = bn2(' ',c2,' ',c4,c5_post)
            [p_xt_p3] = bn2(' ',' ',c3,c4,c5_post)
           % Negative P(Pd(+))...
           c4='f';
            [p_xa_n1] = bn1(' ',' ',' ',c4,' ')
            [p_xb_n1] = bn2(c1,' ',' ',c4,c5_neg)
            [p_xh_n1] = bn2(' ',c2,' ',c4,c5_neg)
            [p_xt_n1] = bn2(' ',' ',c3,c4,c5_neg)
           c4='w';
            [p_xa_n2] = bn1(' ',' ',' ',c4,' ')
```

```
        [p_xb_n2] = bn2(c1,' ',' ',c4,c5_neg)
        [p_xh_n2] = bn2(' ',c2,' ',c4,c5_neg)
        [p_xt_n2] = bn2(' ',' ',c3,c4,c5_neg)
       c4='t';
        [p_xa_n3] = bn1(' ',' ',' ',c4,' ')
        [p_xb_n3] = bn2(c1,' ',' ',c4,c5_neg)
        [p_xh_n3] = bn2(' ',c2,' ',c4,c5_neg)
        [p_xt_n3] = bn2(' ',' ',c3,c4,c5_neg)
       % --------------------- P(xhxhxt/P(pd(+))^P(xa) -----------------------
        p_A = (pd_plus*p_xb_p1*p_xh_p1*p_xt_p1*p_xa_p1) +
              (p_xb_p2*p_xh_p2*p_xt_p2*p_xa_p2*pd_plus) +
              (p_xb_p3*p_xh_p3*p_xt_p3*p_xa_p3*pd_plus)
       % --------------------- P(xhxhxt/P(pd(+))^~P(xa) -----------------------
        p_B = p_A + (pd_neg*p_xb_n1*p_xh_n1*p_xt_n1*p_xa_n1) +
              (p_xb_n2*p_xh_n2*p_xt_n2*p_xa_n2*pd_neg) +
              (p_xb_n3*p_xh_n3*p_xt_n3*p_xa_n3*pd_neg)
        result=p_A/ p_B
       % for printing...
        p_xb1=p_xb_p1*p_xb_p2*p_xb_p3*pd_plus
        p_xh1=p_xh_p1*p_xh_p2*p_xh_p3*pd_plus
        p_xt1=p_xt_p1*p_xt_p2*p_xt_p3*pd_plus
        p_xa1=p_xa_p1*p_xa_p2*p_xa_p3*pd_plus
        Print(i,c1,c2,c3,'-',result,p_xb1,p_xh1,p_xt1,p_xa1)
        input('Press enter to continue...','s');
        close all;clc;
   %% Find the query #6&7 results...
   elseif (c2 =='-')&&(c3~='-')&&(c4~='-')
     % Find the prob. of given Pd...
       [p_xa] = bn1(' ',' ',' ',c4,' ')
       [p_xb1] = bn2(c1,' ',' ',c4,c5_post)
       [p_xt1] = bn2(' ',' ',c3,c4,c5_post)
       p_A=pd_plus*p_xa*p_xb1*p_xt1
     % Find the prob. of not given Pd...
       [p_xb2] = bn2(c1,' ',' ',c4,c5_neg)
       [p_xt2] = bn2(' ',' ',c3,c4,c5_neg)
       p_B=p_A+(pd_neg*p_xa*p_xb2*p_xt2)
     % Final Result...
       result=p_A/ p_B
     % For printing...
       p_xb=p_xb1*pd_plus
       p_xt=p_xt1*pd_plus
       Print(i,c1,'-',c3,c4,result,p_xb,0,p_xt,p_xa)
       input('Press enter to continue...','s');
       close all;clc;
  %% Find the query #8&9 results...
     elseif (c2~='-')&&(c3 =='-')&&(c4=='-')
        % Find the prob. of given Pd...
          c4='f';
           [p_xa_p1] = bn1(' ',' ',' ',c4,' ')
           [p_xb_p1] = bn2(c1,' ',' ',c4,c5_post)
           [p_xh_p1] = bn2(' ',c2,' ',c4,c5_post)
          c4='w';
           [p_xa_p2] = bn1(' ',' ',' ',c4,' ')
           [p_xb_p2] = bn2(c1,' ',' ',c4,c5_post)
           [p_xh_p2] = bn2(' ',c2,' ',c4,c5_post)
          c4='t';
           [p_xa_p3] = bn1(' ',' ',' ',c4,' ')
           [p_xb_p3] = bn2(c1,' ',' ',c4,c5_post)
           [p_xh_p3] = bn2(' ',c2,' ',c4,c5_post)
         % Negative P(Pd(+))...
          c4='f';
           [p_xa_n1] = bn1(' ',' ',' ',c4,' ')
           [p_xb_n1] = bn2(c1,' ',' ',c4,c5_neg)
```

```
          [p_xh_n1] = bn2(' ',c2,' ',c4,c5_neg)
         c4='w';
          [p_xa_n2] = bn1(' ',' ',' ',c4,' ')
          [p_xb_n2] = bn2(c1,' ',' ',c4,c5_neg)
          [p_xh_n2] = bn2(' ',c2,' ',c4,c5_neg)
         c4='t';
          [p_xa_n3] = bn1(' ',' ',' ',c4,' ')
          [p_xb_n3] = bn2(c1,' ',' ',c4,c5_neg)
          [p_xh_n3] = bn2(' ',c2,' ',c4,c5_neg)
      % --------------------- P(xhxhxt/P(pd(+))^P(xa) -----------------------
        p_A = (pd_plus*p_xb_p1*p_xh_p1*p_xa_p1) + (p_xb_p2*p_xh_p2*p_xa_p2*pd_plus)
            + (p_xb_p3*p_xh_p3*p_xa_p3*pd_plus)
      % --------------------- P(xhxhxt/P(pd(+))^P(xa) -----------------------
       p_B = p_A + (pd_neg*p_xb_n1*p_xh_n1*p_xa_n1) +
             (p_xb_n2*p_xh_n2*p_xa_n2*pd_neg) + (p_xb_n3*p_xh_n3*p_xa_n3*pd_neg)
        result = p_A / p_B
      % For printing...
        p_xb1=p_xb_p1*p_xb_p2*p_xb_p3*pd_plus
        p_xh1=p_xh_p1*p_xh_p2*p_xh_p3*pd_plus
        p_xa1=p_xa_p1*p_xa_p2*p_xa_p3*pd_plus
        Print(i,c1,c2,'-','-',result,p_xb1,p_xh1,0,p_xa1)
    %% Find the query #10 result...
    elseif (c2=='-')&&(c3 =='-')&&(c4=='-')&&(c1~='-')
        c4='f';
         [p_xa_p1] = bn1(' ',' ',' ',c4,' ')
         [p_xb_p1] = bn2(c1,' ',' ',c4,c5_post)
        c4='w';
         [p_xa_p2] = bn1(' ',' ',' ',c4,' ')
         [p_xb_p2] = bn2(c1,' ',' ',c4,c5_post)
        c4='t';
         [p_xa_p3] = bn1(' ',' ',' ',c4,' ')
         [p_xb_p3] = bn2(c1,' ',' ',c4,c5_post)
        % Negative P(Pd(+))...
        c4='f';
         [p_xa_n1] = bn1(' ',' ',' ',c4,' ')
         [p_xb_n1] = bn2(c1,' ',' ',c4,c5_neg)
        c4='w';
         [p_xa_n2] = bn1(' ',' ',' ',c4,' ')
         [p_xb_n2] = bn2(c1,' ',' ',c4,c5_neg)
        c4='t';
         [p_xa_n3] = bn1(' ',' ',' ',c4,' ')
         [p_xb_n3] = bn2(c1,' ',' ',c4,c5_neg)
      % --------------------- P(xhxhxt/P(pd(+))^P(xa) -----------------------
        p_A=(pd_plus*p_xb_p1*p_xa_p1)+(p_xb_p2*p_xa_p2*pd_plus)+(p_xb_p3*p_xa_p3*pd_plus)
      % --------------------- P(xhxhxt/P(pd(+))^P(xa) -----------------------
        p_B=p_A+(pd_neg*p_xb_n1*p_xa_n1)+(p_xb_n2*p_xa_n2*pd_neg)+(p_xb_n3*p_xa_n3*pd_neg)
        result=p_A/ p_B
      % For printing...
        p_xb1=p_xb_p1*p_xb_p2*p_xb_p3*pd_plus
        p_xa1=p_xa_p1*p_xa_p2*p_xa_p3*pd_plus
        Print(i,c1,'-','-','-',result,p_xb1,0,0,0)
    else
      % Find the prob. of given Pd...
        [p_xb1] = bn2(c1,' ',' ',c4,c5_post)
        [p_xh1] = bn2(' ',c2,' ',c4,c5_post)
        [p_xt1] = bn2(' ',' ',c3,c4,c5_post)
        [pd_plus] = bn1(' ',' ',' ',' ',c5_post)
        [p_xa] = bn1(' ',' ',' ',c4,' ')
     % --------------------- P(xhxhxt/P(pd(+))^P(xa) -----------------------
       p_A=p_xb1*p_xh1*p_xt1*pd_plus
     % Find the prob. of not given Pd...
       [p_xb2] = bn2(c1,' ',' ',c4,c5_neg)
       [p_xh2] = bn2(' ',c2,' ',c4,c5_neg)
```

```
            [p_xt2] = bn2(' ',' ',c3,c4,c5_neg)
            [pd_neg] = bn1(' ',' ',' ',' ',c5_neg)
        % -------------------- P(xhxhxt/P(~pd(+))^P(xa) -----------------------
             p_B=p_A+(p_xb2*p_xh2*p_xt2*pd_neg)
             result=p_A/ p_B
             Print(i,c1,c2,c3,c4,result,p_xb1,p_xh1,p_xt1,p_xa)
             input('Press enter to continue...','s');
             close all;clc;
        end
end
```

### 3.4. Query execution results

1. Query number 1: $P(Pd = +\,|\, Xb = H, Xh = H, Xt = H)$

| | ID | P(Xb/Pd) | P(Xh/Pd) | P(Xh/Pd) | P(Xa) | Pd(p=+) |
|---|---|---|---|---|---|---|
| Prob. | 1 | 0.7100 | 0.6300 | 0.4900 | 0.5700 | 0.9930 |

2. Query number 1: $P(Pd = +\,|\, Xb = H, Xh = M, Xt = M)$

| | ID | P(Xb/Pd) | P(Xh/Pd) | P(Xh/Pd) | P(Xa) | Pd(p=+) |
|---|---|---|---|---|---|---|
| Prob. | 2 | 0.9500 | 0.0200 | 0.0600 | 0.2100 | 0.0153 |

3. Query number 1: $P(Pd = +\,|\, Xb = H, Xh = M, Xt = L)$

| | ID | P(Xb/Pd) | P(Xh/Pd) | P(Xh/Pd) | P(Xa) | Pd(p=+) |
|---|---|---|---|---|---|---|
| Prob. | 3 | 0.7700 | 0.2000 | 0.1000 | 0.2200 | 0.0589 |

4. Query number 1: $P(Pd = +\,|\, Xb = M, Xh = M, Xt = M)$

| | ID | P(Xb/Pd) | P(Xh/Pd) | P(Xh/Pd) | P(Xa) | Pd(p=+) |
|---|---|---|---|---|---|---|
| Prob. | 4 | 0.7700 | 0.2000 | 0.1000 | 0 | 0.0279 |

5. Query number 1: $P(Pd = +\,|\, Xb = M, Xh = L, Xt = M)$

| | ID | P(Xb/Pd) | P(Xh/Pd) | P(Xh/Pd) | P(Xa) | Pd(p=+) |
|---|---|---|---|---|---|---|
| Prob. | 5 | 0.7700 | 0.2000 | 0.1000 | 0 | 0.0183 |

6. Query number 1: $P(Pd = +\,|\, Xb = M, Xt = M)$

| | ID | P(Xb/Pd) | P(Xh/Pd) | P(Xh/Pd) | P(Xa) | Pd(p=+) |
|---|---|---|---|---|---|---|
| Prob. | 6 | 0.1900 | 0 | 0.1000 | 0.2200 | 0.0118 |

7. Query number 1: $P(Pd = +\,|\, Xb = L, Xt = M)$

| | ID | P(Xb/Pd) | P(Xh/Pd) | P(Xh/Pd) | P(Xa) | Pd(p=+) |
|---|---|---|---|---|---|---|
| Prob. | 7 | 0.0200 | 0 | 0.0300 | 0.2100 | 0.0531 |

8. Query number 1: $P(Pd = +\,|\, Xb = L, Xh = M)$

| | ID | P(Xb/Pd) | P(Xh/Pd) | P(Xh/Pd) | P(Xa) | Pd(p=+) |
|---|---|---|---|---|---|---|
| Prob. | 8 | 0.0200 | 0.2000 | 0 | 0 | 0.0335 |

9. Query number 1: $P(Pd = +\mid Xb = L, Xh = H)$

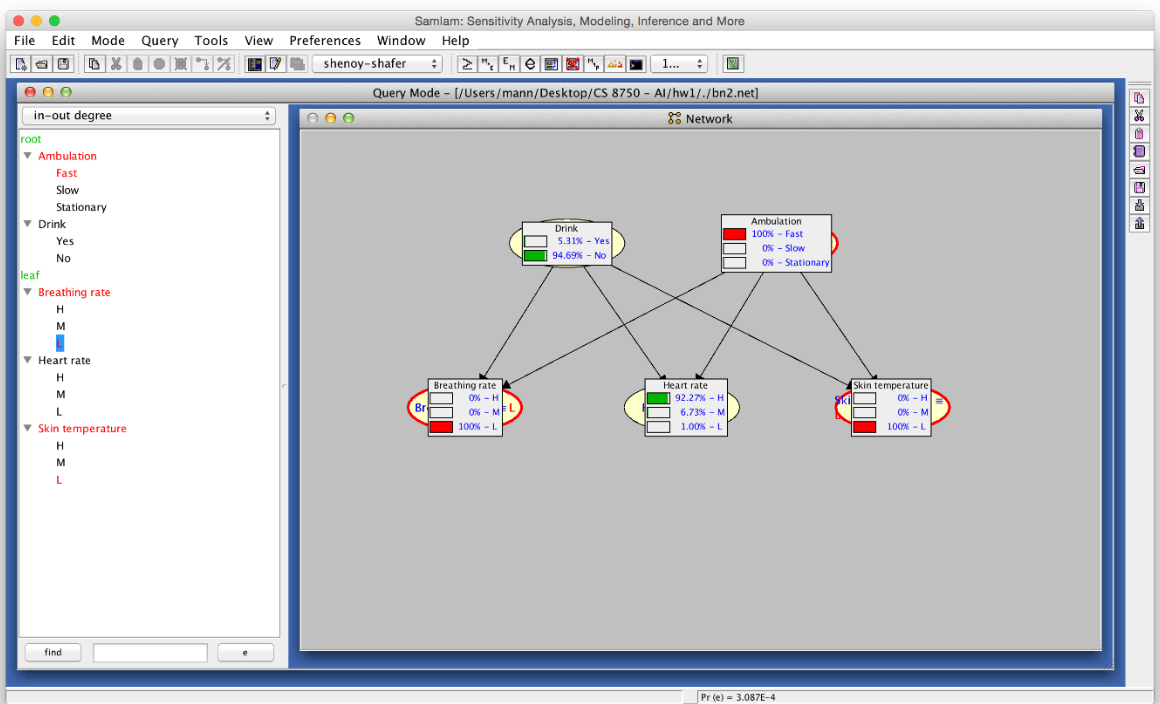
| | ID | P(Xb/Pd) | P(Xh/Pd) | P(Xh/Pd) | P(Xa) | Pd(p=+) |
|---|---|---|---|---|---|---|
| Prob. | 9 | 0.0200 | 0.2000 | 0 | 0 | 0.1414 |

10. Query number 1: $P(Pd = +\mid Xb = M)$

| | ID | P(Xb/Pd) | P(Xh/Pd) | P(Xh/Pd) | P(Xa) | Pd(p=+) |
|---|---|---|---|---|---|---|
| Prob. | 10 | 0.0200 | 0 | 0 | 0 | 0.0804 |

## 3.5.Samlam implementation

The complex Bayesian network was implemented explicitly in Samlam. The results were verified against the Matlab program and matched. The graph is shown below. As an example, we've selected the evidence to The Bayesian Network 2 graph constructed using Samlam to answer query number 7 : $P(Pd = +\mid Xa = Fast, Xb = L, Xt = L)$




The results of running the query using junction tree algorithm Shenoy-Shafer: $P(Pd = +\mid Xa = Fast, Xb = L, Xt = L) = 5.31\%$

## 4. Conclusions

We implemented successfully two Bayesian networks: one naïve and one complex. Using the equations derived in sections 2.2 and 3.2, these implementations cover all possible cases for missing values, as tested by the 10 queries set out in the introduction of this assignment. We also modeled the Bayesian networks in SamIam and were able to get results that matched the output of our implementations.

When comparing the same queries in both networks, we observe that Pr decreases in the second network. This is expected to be because of the introduction of an alternate cause to explain away Xb Xh and Xt. The new probabilities are more in line with the prior probabilities of Pd, suggesting a more accurate model.