MATLAB program used as tool in this modeling

```
%% Creating the Bayesian Network

%=== setup
adj = [0 1 1 0; 0 0 0 0; 0 0 0 1; 0 0 0 0]; %
adjacency matrix
nodeNames = {'S', 'B', 'L', 'X'};           % nodes
S = 1; B = 2; L = 3; X = 4;                 % node
identifiers
n = numel(nodeNames);                       % number
of nodes
t = 1; f  = 2;                              % true
and false
values = cell(1,n);                         % values
assumed by variables
fori = 1:numel(nodeNames)
values{i} = [t f];
end


%=== Conditional Probability Table
CPT{S} = [.2 .8];
CPT{B}(:,t) = [.25 .05] ; CPT{B}(:,f) = 1 -
CPT{B}(:,t);
% CPT{L}(:,t) = [.03 .0005]; CPT{L}(:,f) = 1 -
CPT{L}(:,t);
```

```
CPT{L}(:,t) = [.3 .005]; CPT{L}(:,f) = 1 -
CPT{L}(:,t);
CPT{X}(:,t) = [.6 .02]; CPT{X}(:,f) = 1 -
CPT{X}(:,t);

%=== draw the network
nodeLabels = {'Smoking', 'Bronchitis', 'Lung Cancer',
'Abnormal Xrays'};
bg = biograph(adj, nodeLabels, 'arrowsize', 3);
set(bg.Nodes, 'shape', 'ellipse');
bgInViewer = view(bg);

%=== save as figure
bgFig = figure;
copyobj(bgInViewer.hgAxes,bgFig)

%=== annotate using the CPT
[xp, xn] = find(adj);     % xp = parent id, xn = node
id
pa(xn) = xp;              % parents
pa(1) = 1;                % root is parent of itself

s1 = cell(1,n); s2 = cell(1,n); pos = zeros(n,2);
for i = 2:n
pos(i,:) = bgInViewer.Nodes(i).Position;
    s1{i} = sprintf('P(%s|%s=t) = %f', nodeNames{i},
nodeNames{pa(i)}, CPT{i}(1,t));
    s2{i} = sprintf('P(%s|%s=f) = %f', nodeNames{i},
nodeNames{pa(i)}, CPT{i}(2,t));
```

```
end

pos(1,:) = bgInViewer.Nodes(1).Position; % root
s1{1} = sprintf('P(%s=t) = %f', nodeNames{1},
CPT{1}(1));
s2{1} = ' ';

text(pos(:,1)+2, pos(:,2)-10, s1)
text(pos(:,1)+2, pos(:,2)-15, s2)


%% Initializing the Bayesian Network


root = find(sum(adj,1)==0); % root is any node with
no parent
[nodes, edges] = bnMsgPassCreate(adj, values, CPT);
[nodes, edges] = bnMsgPassInitiate(nodes, edges,
root)

%=== conditional probability given the empty set []
for i = 1:n
disp(['P(' nodeNames{i}, '|[]) = '
num2str(nodes(i).P(1))]);
end

%=== assign relevant info to each node handle
nodeHandles = bgInViewer.Nodes;
for i = 1:n
```

```
nodeHandles(i).UserData.Distribution = [nodes(i).P];
end

%=== draw customized nodes
bgInViewer.ShowTextInNodes = 'none';

set(nodeHandles, 'shape','circle')
bgInViewer.CustomNodeDrawFcn = @(node)
customnodedraw(node);
%bgInViewer.Scale = .7
bgInViewer.dolayout



%=== inference with B = t
evNode = B;
evValue = t;
[n1, e1, A1, a1] = bnMsgPassUpdate(nodes, edges, [],
[], evNode, evValue);

for i = 1:n
disp(['P(' nodeNames{i}, '|B=t) = '
num2str(n1(i).P(1))]);
end

%== plot and compare
figure(); subplot(2,1,1);
x = cat(1,nodes.P);
bar(x, 'stacked'); set(gca, 'xticklabel', nodeNames);
```

```
ylabel('Probability');
title('Initialized network with empty evidence set')
legend({'true', 'false'}, 'location',
'SouthEastOutside')
hold on; subplot(2,1,2);
x1 = cat(1,n1.P);
bar(x1, 'stacked'); set(gca, 'xticklabel',
nodeNames);
ylabel('Probability');
title('Updated network with evidence B=true')
legend({'true', 'false'}, 'location',
'SouthEastOutside')


evNode = X;
evValue = t;
[n2, e2, A2, a2] = bnMsgPassUpdate(nodes, edges, [],
[], evNode, evValue);

for i = 1:n
disp(['P(' nodeNames{i}, '|X=t) = '
num2str(n2(i).P(1))]);
end



evNode = B;
evValue = t;
```

```
[n3, e3, A3, a3] = bnMsgPassUpdate(n2, e2, A2, a2,
evNode, evValue);

for i = 1:n
disp(['P(' nodeNames{i}, '|B=t,X=t) = '
num2str(n3(i).P(1))]);
end



figure(); subplot(3,1,1);
bar(x1,  'stacked'); set(gca, 'xticklabel',
nodeNames);
ylabel('Probability'); title('Bronchitis diagnosis');
legend({'true', 'false'}, 'location',
'SouthEastOutside')
hold on; subplot(3,1,2);
x2 = cat(1,n2.P);
bar(x2,'stacked'); set(gca, 'xticklabel', nodeNames);
ylabel('Probability'); title('Abnormal x-rays');
legend({'true', 'false'}, 'location',
'SouthEastOutside')
hold on; subplot(3,1,3);
x3 = cat(1,n3.P);
bar(x3, 'stacked'); set(gca, 'xticklabel',
nodeNames);
ylabel('Probability'); title('Bronchitis and abnormal
x-ray');
```

```
legend({'true', 'false'}, 'location',
'SouthEastOutside')

evNode = B;
evValue = f;
[n4, e4, A4, a4] = bnMsgPassUpdate(n2, e2, [], [],
evNode, evValue);

for i = 1:n
disp(['P(' nodeNames{i}, '|B=f,X=t) = '
num2str(n4(i).P(1))]);
end

figure();
bar3([n3(S).P(:,t) n4(S).P(:,t); n3(L).P(:,t)
n4(L).P(:,t)]);
colormap(summer); zlabel('Probability');
set(gca,'xticklabel',{'Smoking','Lung
Cancer'},'yticklabel', {'With Bronchitis', 'Without
Bronchitis'});
set(gca,'xticklabel',{'With Bronchitis','Without
Bronchitis'},'yticklabel', {'Smoking','Lung
Cancer'});
title('Conditional probabilities with evidence of
abnormal x-ray results')
view(50,35);

%=== add node D to the network
D = 5;
```

```
CPT{D}(:,:,t) = [.75 .1; .5 .05];
CPT{D}(:,:,f) = 1 - CPT{D}(:,:,t);
values{D} =  [1 2];
adj(end+1,:) = [0 0 0 0];
adj(:,end+1) = [0 1 1 0 0];

%=== draw the updated network
nodeLabels = {'Smoking', 'Bronchitis', 'Lung Cancer',
'Xrays', 'Dyspnea'};
nodeSymbols = {'S', 'B', 'L', 'X', 'D'};
bg = biograph(adj, nodeLabels, 'arrowsize', 4);
nodeHandles= bg.Nodes;
set(nodeHandles, 'shape', 'ellipse');
view(bg)



isAcyclic = graphisdag(sparse(adj | adj'))



%=== combine B and L
adj(B,:) = adj(B,:) | adj(L,:);
adj(:,B) = adj(:,B) | adj(:,L);
adj(L,:) = []; adj(:,L) = [];

%=== update the probability distribution accordingly
b1 = kron(CPT{B}(1,:), CPT{L}(1,:));
b2 = kron(CPT{B}(2,:), CPT{L}(2,:));
x = [CPT{X}(1,:) CPT{X}(1,:)];
```

```
d =  reshape((CPT{D}(:,:,1))', 1, 4);

%=== update the node values
S = 1; BL = 2; X = 3; D = 4;
nodeNames = {'S', 'BL', 'X', 'D'};
tt = 1; tf = 2; ft = 3; ff = 4;
values{BL} = 1:4;
values(L) = [];

%=== create a clustered Conditional Probability Table
cCPT{S} = CPT{S};
cCPT{BL}(t,:) = b1; cCPT{BL}(f,:) = b2;
cCPT{D}(:,t) = d; cCPT{D}(:,f) = 1 - d;
cCPT{X}(:,t) = x; cCPT{X}(:,f) = 1 - x;

%=== create and intiate the net
root = find(sum(adj,1)==0); % root (node with no
parent)
[cNodes, cEdges] = bnMsgPassCreate(adj, values,
cCPT);
[cNodes, cEdges] = bnMsgPassInitiate(cNodes, cEdges,
root);

for i = 1:n
disp(['P(' nodeNames{i}, '|[]) = '
num2str(cNodes(i).P(1))]);
end

%% Drawing the Expanded Network
```

```
%=== draw the network
nodeLabels = {'Smoking', 'Bronchitis or Lung Cancer',
'Abnormal X-rays', 'Dyspnea'};
cbg = biograph(adj, nodeNames, 'arrowsize', 4);
set(cbg.Nodes, 'shape', 'ellipse');
cbgInViewer = view(cbg);

%=== assign relevant info to each node handle
cnodeHandles = cbgInViewer.Nodes;
for i = 1:n
cnodeHandles(i).UserData.Distribution =
[cNodes(i).P];
end

%=== draw customized nodes
set(cnodeHandles, 'shape','circle')
colormap(summer)
cbgInViewer.ShowTextInNodes = 'none';
cbgInViewer.CustomNodeDrawFcn = @(node)
customnodedraw(node);
cbgInViewer.Scale = .7
cbgInViewer.dolayout

[n5, e5, A5, a5] = bnMsgPassUpdate(cNodes, cEdges,
[], [], D, t);

p(1,:) = n5(S).P;
```

```
p(2,:) = [sum(n5(BL).P([tt,tf])), 1-
sum(n5(BL).P([tt,tf]))];     % P(B|evidence)
p(3,:) = [sum(n5(BL).P([tt,ft])), 1-
sum(n5(BL).P([tt,ft]))]; % P(L|evidence)
p(4,:) = n5(X).P;
p(5,:) = n5(D).P;

for i = 1:5
disp(['P(' nodeSymbols{i}, '|D=t) = '
num2str(p(i,1))]);
end

%=== adjust the CPT to reflect B = 1 before
clustering into BL node
B = 2; L = 3;
CPT{B}(:,1) = [1 1] ; CPT{B}(:,2) = 1 - CPT{B}(:,1);
b1 = kron(CPT{B}(1,:), CPT{L}(1,:));
b2 = kron(CPT{B}(2,:), CPT{L}(2,:));

%=== create a clustered Conditional Probability Table
BL = 2;
cCPT{BL}(1,:) = b1; cCPT{BL}(2,:) = b2;

%=== create and intiate the net
root = find(sum(adj,1)==0); % root (node with no
parent)
[cNodes, cEdges] = bnMsgPassCreate(adj, values,
cCPT);
```

```
[cNodes, cEdges] = bnMsgPassInitiate(cNodes, cEdges,
root);

%=== instantiate for F = 1
[n7, e7, A7, a7] = bnMsgPassUpdate(cNodes, cEdges,
[], [], D, t);
w(1,:) = n7(S).P;
w(2,:) = [sum(n7(BL).P([tt,tf])), 1-
sum(n7(BL).P([tt,tf]))];     % P(B|evidence)
w(3,:) = [sum(n7(BL).P([tt,ft])), 1-
sum(n7(BL).P([tt,ft]))]; % P(L|evidence)
w(4,:) = n7(X).P;
w(5,:) = n7(D).P;

for i = 1:5
    disp(['P(' nodeSymbols{i}, '|B=t,D=t) = '
num2str(w(i,1))]);
end

%=== adjust the CPT to reflect L = 1 before
clustering into BL node
B = 2; L = 3;
CPT{B}(:,t) = [.25 .05] ; CPT{B}(:,f) = 1 -
CPT{B}(:,t);
CPT{L}(:,t) = [1 1]; CPT{L}(:,f) = 1 - CPT{L}(:,t);

b1 = kron(CPT{B}(t,:), CPT{L}(t,:));
b2 = kron(CPT{B}(f,:), CPT{L}(f,:));
```

```
BL = 2;
cCPT{BL}(t,:) = b1; cCPT{BL}(f,:) = b2;

%=== create and intiate the net
root = find(sum(adj,1)==0); % root (node with no
parent)
[cNodes, cEdges] = bnMsgPassCreate(adj, values,
cCPT);
[cNodes, cEdges] = bnMsgPassInitiate(cNodes, cEdges,
root);

%=== instantiate for D = 1
[n8, e8, A8, a8] = bnMsgPassUpdate(cNodes, cEdges,
[], [], D, t);
v(1,:) = n8(S).P;
v(2,:) = [sum(n8(BL).P([tt,tf])), 1-
sum(n8(BL).P([tt,tf]))];     % P(B|evidence)
v(3,:) = [sum(n8(BL).P([tt,ft])), 1-
sum(n8(BL).P([tt,ft]))]; % P(L|evidence)
v(4,:) = n8(X).P;
v(5,:) = n8(D).P;

for i = 1:5
    disp(['P(' nodeSymbols{i}, '|L=t,D=t) = '
num2str(v(i,1))]);
end

y = [w(2:3,1)  v(2:3,1)];
figure();
```

```
bar(y);
set(gca, 'xticklabel', {'Bronchitis', 'Lung
Cancer'});
ylabel('Probability'); title('Explaining away with
evidence of dyspnea')
legend('B = t', 'L = t', 'location',
'SouthEastOutside');
colormap(summer)
```