Onur Poyraz
2010401036
06 Mayıs 2016

Homework III

1) I can not implement the Loopy Belief Propogation Algorithm rightly. I put the code that can work correctly is given below.

```
function LoopyBP(Edges,Factors)

len = length(Edges(:,1));
num_of_nodes = max(Edges(:));

for i=1:num_of_nodes
    F(i) = -i;
end

FactorGraph = [0 0];
i = 1;
for j=1:len
    if any(Edges(j,i) == FactorGraph(:,2)) == 0
        FactorGraph = [FactorGraph;F(Edges(j,i)),Edges(j,i)];
    end
end
i = 2;
for j=1:len
    g2 = FactorGraph(:,2);
    if any(Edges(j,i) == FactorGraph(:,2)) == 0
        FactorGraph =
[FactorGraph;F(Edges(j,i)),Edges(j,i);Edges(j,i-1),F(Edges(j,i))];
    else
        FactorGraph = [FactorGraph;Edges(j,i-1),F(Edges(j,i))];
    end
end

FactorGraph(1,:) = [];
MirrorGraph = rot90(FactorGraph,2);
FactorGraph = [FactorGraph;MirrorGraph];

for i=1:length(FactorGraph(:,1))
    if FactorGraph(i,1) > 0
        Messages{i} = ones(1,length(Factors{FactorGraph(i,1)}));
    else
        Messages{i} = ones(1,length(Factors{FactorGraph(i,2)}));
    end
end
```

I can not implement the message updates. Therefore my LoopyBP algorithm is not working.

2) In BruteForce Algorithm I take the inputs as a two parts. One part is edge matrix which identificates the edges of the network. I took the edges as a N*2 dimension and first column implies the parents where seconds are implies the child. Additionally edges should be well ordered such that the parents always should come first. I took the Factors Matrix as a cell and it is second input for my algorithm and every cell is for one nodes. Every cell holds the CPT of corresponding node.

```
function margBruteForce(Edges,Factors)

len = length(Edges(:,1));
num_of_nodes = max(Edges(:));

for i=1:num_of_nodes
    if any(Edges(:,2)==i) == 0
        margs{i} = Factors{i};
    else
        fact = Factors{i};
        [row,column] = size(fact);
        coef = 1;
        for j = 1:len
            if i == Edges(j,2)
                marg = margs{Edges(j,1)}';
                marg_len = length(marg);
                marg_matrix = [];
                for k = 1:marg_len
                    marg_matrix = [marg_matrix;repmat(marg(k),coef,1)];
                end
                fact = fact.*repmat(marg_matrix,row/
length(marg_matrix),column);
                coef = coef*marg_len;
            end
        end
        sum = 0;
        for l = 1:row
            sum = sum + fact(l,:);
        end
        margs{i} = sum;
    end
end

for i=1:num_of_nodes
    margs{i}
end
```

3) In this part I just use the bruteForce Algorithm because I can not take an output from LoopyBP. I wrote the inputs and corresponding output for bruteForceAlgorithm.

```
Edges = [1,3;2,3;2,4;3,5];
Factors{1} = [0.6,0.4];
Factors{2} = [0.7,0.3];
Factors{3} = [0.3 0.4 0.3; 0.05 0.25 0.7; 0.9 0.08 0.02; 0.5 0.3 0.2];
Factors{4} = [0.95,0.05;0.2,0.8];
Factors{5} = [0.1,0.9;0.4,0.6;0.99,0.01];
```

```
margBruteForce(Edges,Factors)
```

The marginal probabilities that algorithms found is that;
P(X1) = [0.6, 0.4]
P(X2) = [0.7, 0.3]
P(X3) = [0.3620, 0.2884,0.3496]
P(X4) = [0.7250, 0.2750]
P(X5) = [0.4977, 0.5023]

4) In this part again I just use the margBruteForce Algorithm. Procedure is completely same with previous question. I wrote the inputs and corresponding outputs below.

```
Edges = [1,3;2,3;2,4;3,5;4,5];
Factors{1} = [0.6,0.4];
Factors{2} = [0.7,0.3];
Factors{3} = [0.3 0.4 0.3; 0.05 0.25 0.7; 0.9 0.08 0.02; 0.5 0.3 0.2];
Factors{4} = [0.95,0.05;0.2,0.8];
Factors{5} = [0.4,0.6;0.6,0.4;0.99,0.01;0.01,0.99;0.4,0.6;0.8,0.2];
margBruteForce(Edges,Factors)
```

The marginal probabilities that algorithms found is that;
P(X1) = [0.6, 0.4]
P(X2) = [0.7, 0.3]
P(X3) = [0.3620, 0.2884,0.3496]
P(X4) = [0.7250, 0.2750]
P(X5) = [0.5910, 0.4090]