1.

a) The loss function now is

b) The update rule is

c) In fact, a neural network, which computes the output as,  can be just treated as a processor of **Logistic regression**, which is **linear regression** essentially. So it can be solved analytically such as linear regression.

d)

Depending on the hint, the solution to calculate the vector  that minimizes



for a coefficient matrix  and a bias vector  is



In the current scene, the destination loss function is



To calculate the vector  that minimizes the loss function base on Eq.(1) and Eq.(3), we can know





 is the unit matrix, and  is . So the solution is:



2.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Iteration |  |  |  |  |  |  |  |  |
| 1 |  | 0 | 3 |  | 1 | 2 | 2 | 2 |
| 2 |  | 1 | 11/3 |  | 1 | 1 | 2 | 2 |
| 3 |  | 1.5 | 4.5 |  | 1 | 1 | 2 | 2 |
| 4 |  | 1.5 | 4.5 |  | 1 | 1 | 2 | 2 |

a)

b) Let , then we initialize the parameters and then iterate the following two steps:

E-step:

Compute , which is the probability that data point I is generated by Gaussian j.

　 let

M-step:

Update the parameters:

(Where N is the total number of data points)

If we fix we just don’t update in the M-step above.

c)

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Iter |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 1 | 0.5 | 0 | 3 | 1 | 1 | 0.8176 | 0.1824 | 0.0006 | 0.0000 | 0.1824 | 0.8176 | 0.9994 | 1.0000 |
| 2 | 0.25 | 1.1841 | 3.6058 | 0.1538 | 1.8157 | 0.8694 | 0.2111 | 0.0000 | 0.0000 | 0.1306 | 0.7889 | 1.0000 | 1.0000 |
| 3 | 0.27 | 1.1954 | 3.6679 | 0.1572 | 1.7157 | 0.8960 | 0.2596 | 0.0000 | 0.0000 | 0.1040 | 0.7404 | 1.0000 | 1.0000 |
| 4 | 0.29 | 1.2246 | 3.7213 | 0.1742 | 1.6442 | 0.9117 | 0.3548 | 0.0000 | 0.0000 | 0.0883 | 0.6452 | 1.0000 | 1.0000 |

3.

a) No.

The Bayes Network (a) and (b) both denotes so the joint distributions that can be represented by (a) and (b) are the same.

For (a) by defining CPTs of p(a) p(b|a) p(c|b), we know p(a,b,c)= p(a)p(b|a)p(c|b)

Same for (b), we get

So both Bayes Networks can represent p(a,b,c), p(a,b), p(b,c), and cannot represent p(a,c).

There is no joint distribution that can be represented using network in (a) that cannot be represented using network in (b).

b) Yes.

(a) denotes while (b) denotes

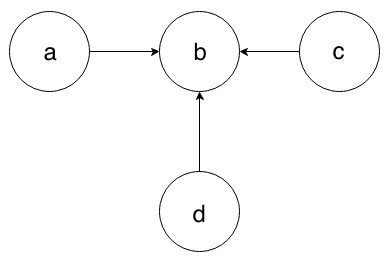
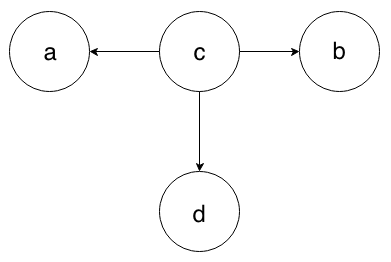
Same as in 3.a) (a) network can represent p(a,b,c), p(a,b), p(b,c); while (b) network can represent p(a,b,c), p(a,b), p(a,c).

So there is a joint distribution (p(b,c)) that can be represented using network in (a) that cannot be represented using network in (b).

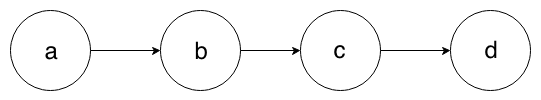
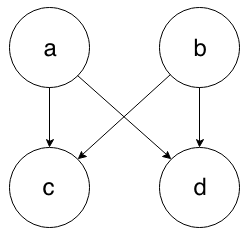
c) The joint distribution is

d)



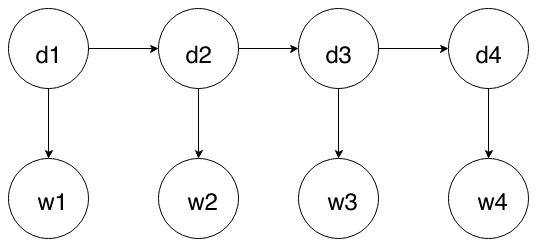






4.

a) The HMM Bayes network is:



dn for whether he dopped in year n; wn for whether he win in year n.

b)

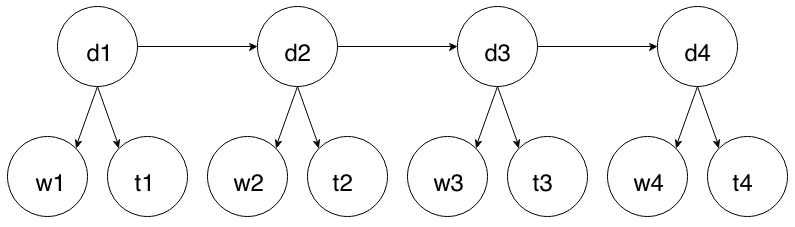
c)

First we compute backwards messages:

Then we combine these with the forwards messages computed previously and normalize:

d)

Given that drug tests are administered in each of the four years, we are adding new evidence to the HMM. Use tn to denote the test result of year n, we have:

5.

a) When in state a it can take action a->b or a->e. In other states, however, it can take only one action. So there are two distinct policies in this MDP:

: Whenever it gets state a, take action a->b

: Whenever it gets state a, take action a->e

b)

n denotes the nth time the MDP reach state a.

for

Since

For this array, we can calculate the nth element as:

for

Since

For this array, we can calculate the nth element as:

c)

Since

If we want the optimal policy take the agent through d. We need to have the

Which is：

It is always true for

So, for all , the optimal policy takes the agent through d.

6．

a) The variables are , which denotes the row number of the Queen in column k.

can pick values in domain: {1,2,3,4,5}

|  |  |
| --- | --- |
| remaining variables | possible values |
|  | {3,4,5} |
|  | {2,4,5} |
|  | {2,3,5} |
|  | {2,3,4} |

The constraints is that are not attacking each other.

b)

c) Since all remaining variables have 3 remaining values, we assign next.

|  |  |
| --- | --- |
| remaining variables | possible values |
|  | {5} |
|  | {2} |
|  | {2,4} |

After that:

After that:

|  |  |
| --- | --- |
| remaining variables | possible values |
|  | {2} |
|  | {2,4} |

Then:

|  |  |
| --- | --- |
| remaining variables | possible values |
|  | {4} |

.

7.

a)

The state is a -tuple, and  is the number of candidate positions (in this problem, there are 19 candidate positions among all 25 positions, so n is 19). Each element in the 19-tuple is a position (i,j) (i=1, 2, 3, 4; j=1, 2, 3, 4, 5, 6).

Since each element has 19 positions as candidate values, and each state contains 19 positions, so there are states in sum. In fact, it’s just a theoretical upper bound because most of states are unreachable from the start state.

The start state is a tuple contains all the candidate start positions, i.e.

The successor function where , and is a successor function for a position.

The goal state ((1,6) is the goal position, the upper right corner.).

b)

Use the heuristic function

M(x,y) is the function of Manhattan distance. GOAL is (1,6) in this case. So the heuristic function is the sum of distances between each component position in current state and the goal position.

c)

Suppose  is the inverse function of  for the variable  ( is treated as a constant value temporarily here). Then

