Homework 4 **Name: ____________________**
ECE/CS 498 DS Spring 2019 **NetID: ____________________**
Issued: 04/17/19
Due: 04/24/19

HMM Forward-Backward Algorithm

You will implement the forward-backward algorithm for HMMs.

Part 1

Files:

HMM.py

HMM_example.py

What to submit:

A modified HMM.py with your implementation.

You will need to fill in the missing code in HMM.py:

- ***def forward_algorithm*:** calculate $P(S_t|E_1, E_2, \ldots, E_t)$, the probability of the hidden state at time t given the observation(s) up to time t
- ***def backward_algorithm*:** calculate $P(E_{t+1}, \ldots, E_n|S_t)$, the probability of the future observation(s) given the hidden state at time t
- ***def forward_backward:*** calculate $P(S_t|E_1, E_2, \ldots, E_n)$, the probability of the hidden state at time t given all the observations

In HMM_example.py, we provide the security example you solved in ICA4. You can use it to test your implementation.

forward

```python
    def forward_algorithm(self, seq):
        """
        Apply forward algorithm to calculate probabilities of seq

        :param seq: Observed sequence to calculate probabilities upon
        :return: Alpha matrix with 1 row per time step
        """

        T = len(seq)

        # Initialize forward probabilities matrix Alpha
        Alpha = np.zeros((T, self.n_states))

        # Your implementation here
        hidden_state_idx = seq[0]
        Alpha[0] = self.pi0 * self.B[:, hidden_state_idx]
        Z = np.sum(Alpha[0])
        Alpha[0] = Alpha[0] / Z
        for i in range(1, T):
            hidden_state_idx = seq[i]
            Alpha[i] = (Alpha[i - 1].dot(self.A)) * (self.B[:, hidden_state_idx])
            Z = np.sum(Alpha[i])
            Alpha[i] = Alpha[i] / Z

        return Alpha
```

backward

```python
    def backward_algorithm(self, seq):
        """
        Apply backward algorithm to calculate probabilities of seq

        :param seq: Observed sequence to calculate probabilities upon
        :return: Beta matrix with 1 row per timestep
        """

        T = len(seq)

        # Initialize backward probabilities matrix Beta
        Beta = np.zeros((T, self.n_states))

        # Your implementation here
        Beta = np.zeros((T, self.n_states))
        Beta[T - 1] = [1] * self.n_states
        for i in range(T - 2, -1, -1):
            hidden_state_idx = seq[i + 1]
            temp = self.B[:, hidden_state_idx] * Beta[i + 1]
            Beta[i] = self.A.dot(temp)

        return Beta
```

forward-backward

```python
    def forward_backward(self, seq):
        """
        Applies forward-backward algorithm to seq

        :param seq: Observed sequence to calculate probabilities upon
        :return: Gamma matrix containing state probabilities for each timestamp
        :raises: ValueError on bad sequence
        """

        # Convert sequence to integers
        if all(isinstance(i, str) for i in seq):
            seq = [self.emissions.index(i) for i in seq]

        # Infer time steps
        T = len(seq)

        # Calculate forward probabilities matrix Alpha
        Alpha = self.forward_algorithm(seq)
        # Initialize backward probabilities matrix Beta
        Beta = self.backward_algorithm(seq)

        # Initialize Gamma matrix
        Gamma = np.zeros((T, self.n_states))

        # Your implementation here
        for i in range(T):
            Gamma[i] = Alpha[i] * Beta[i]
            Z = np.sum(Gamma[i])
            Gamma[i] = Gamma[i] / Z

        return Alpha, Beta, Gamma

    def hidden_state_predict(self, seq, Gamma):
        T = len(seq)

        hidden_state = []
        for i in range(T):
            idx = np.where(Gamma[i, :] == np.max(Gamma[i, :]))[0][0]
            hidden_state.append(self.states[idx])

        return hidden_state
```

Name: ____________________
NetID: ____________________

Part 2

Based on the forward-backward algorithm you implemented in Part 1, provide the most likely sequence of the hidden states for the HMM given below. For partial credit, please also provide $P(S_t|E_1, E_2, \ldots, E_t)$ (Alpha), $P(E_{t+1}, \ldots, E_n|S_t)$ (Beta), and $P(S_t|E_1, E_2, \ldots, E_n)$ (Gamma).

Transition probability matrix A:

	A	B	C	D
A	0.1	0.3	0.3	0.3
B	0.7	0.1	0.1	0.1
C	0.25	0.25	0.25	0.25
D	0.1	0.4	0.4	0.1

Observation matrix B:

	e0	e1	e2	e3	e4
A	0.6	0.1	0.1	0.1	0.1
B	0.1	0.6	0.1	0.1	0.1
C	0.2	0.2	0.2	0.2	0.2
D	0	0	0	0.5	0.5

The initial distribution of hidden states π:

A	B	C	D
0.25	0.25	0.25	0.25

Observations:

t=1	t=2	t=3	t=4	t=5	t=6	t=7	t=8	t=9
e4	e3	e2	e2	e0	e1	e0	e0	e4

Alpha:

Alpha:

	A	B	C	D
t=1	0.111111	0.111111	0.222222	0.555556
t=2	0.102857	0.165714	0.331429	0.4
t=3	0.222449	0.259184	0.518367	0
t=4	0.333265	0.222245	0.44449	0
t=5	0.72002	0.0933268	0.186654	0
t=6	0.0779633	0.691528	0.230509	0
t=7	0.879783	0.0400722	0.0801443	0
t=8	0.485852	0.171383	0.342765	0
t=9	0.113354	0.110831	0.221662	0.554154

Beta:

Beta:

	A	B	C	D
t=1	2.4352e-06	1.35343e-06	2.16476e-06	1.64303e-06
t=2	8.12546e-06	9.14392e-06	8.38007e-06	1.05122e-05
t=3	9.65315e-05	6.25828e-05	8.80443e-05	0.000127188
t=4	0.000456084	0.00158771	0.00073899	0.000536328
t=5	0.0035892	0.0017604	0.003132	0.0047574
t=6	0.00846	0.01602	0.01035	0.01062
t=7	0.033	0.111	0.0525	0.039
t=8	0.25	0.15	0.225	0.18
t=9	1	1	1	1

Gamma:

Gamma:

	A	B	C	D
t=1	0.149094	0.0828634	0.265073	0.502969
t=2	0.0895461	0.162352	0.297579	0.450523
t=3	0.257681	0.194646	0.547673	0
t=4	0.182397	0.423433	0.39417	0
t=5	0.775323	0.0492899	0.175387	0
t=6	0.0466998	0.78438	0.168921	0
t=7	0.770338	0.118021	0.111641	0
t=8	0.541538	0.114615	0.343846	0
t=9	0.113354	0.110831	0.221662	0.554154

hidden-state:

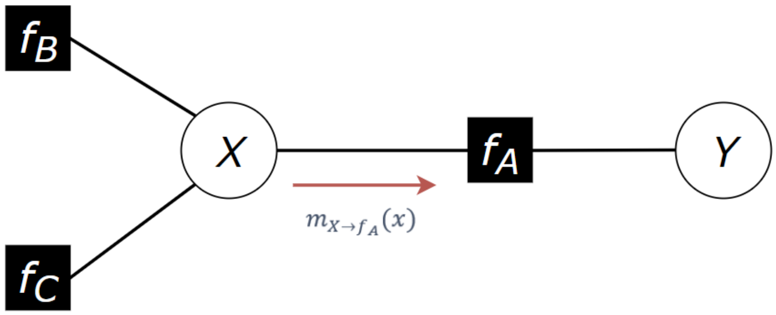
```
predicted hidden state:
['D', 'D', 'C', 'B', 'A', 'B', 'A', 'A', 'D']
```

Homework 4
ECE/CS 498 DS Spring 2019
Issued: 04/17/19
Due: 04/24/19

Name: ____________________
NetID: ____________________

Factor Graphs & Belief Propagation

Problem 1. Consider the following factor graph.



Factor Graph (1)

Factor function values for f_A, f_B, and f_C are:

X	Y	f_A
0	0	0.3
0	1	0.1
1	0	0.4
1	1	0.2

X	f_B
0	0.4
1	0.6

X	f_C
0	0.3
1	0.7

1. Write the expression for $P(Y)$ in terms of $m_{X \to f_A}(x)$ and f_A.

$P(Y) = \sum_{x} m_{x \to f_A}(x)\, f_A(X, Y)$

Homework 4
ECE/CS 498 DS Spring 2019
Issued: 04/17/19
Due: 04/24/19

Name: ____________________
NetID: ____________________

2. First, calculate the message $m_{X\to f_A}(x)$ based on the tables provided, then calculate the value of $P(Y)$. Show all steps of your work.

$$m_{X\to f_A}(x) = \begin{array}{c c} & x \\ \begin{array}{c} 0 \\ 1 \end{array} & \begin{bmatrix} 0.4\times 0.3 = 0.12 \\ 0.6\times 0.7 = 0.42 \end{bmatrix} \end{array}$$

$P(Y) =$

$m_{x\to f_A}(x) = m_{f_B(x)\to x}\ m_{f_C(x)\to x}$

$P(Y=0) = \sum_x m_{x\to f_A}(x) f_A(x,Y)$

$= 0.12\times 0.3 + 0.42\times 0.4 = 0.204$

$P(Y=1) = \sum_x m_{x\to f_A}(x) f_A(x,Y)$

$= 0.12\times 0.1 + 0.42\times 0.2 = 0.096$

$Z = 0.204 + 0.096 = 0.3$

$\Rightarrow P(Y=0) = 0.204/0.3 = 0.68$

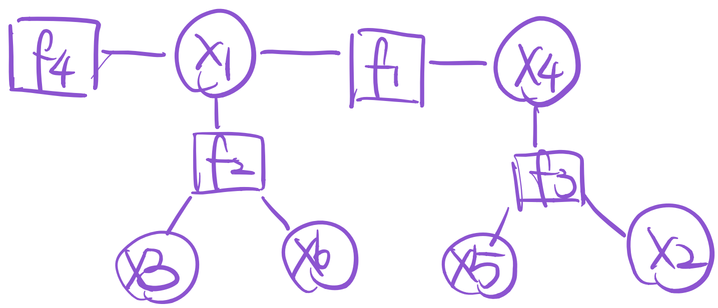
$P(Y=1) = 0.096/0.3 = 0.32$

Homework 4 **Name:** ____________________
ECE/CS 498 DS Spring 2019 **NetID:** ____________________
Issued: 04/17/19
Due: 04/24/19

Problem 2. Given a function that can be factorized as follows.

$$f(X_1, X_2, X_3, X_4, X_5, X_6) = f_1(X_1, X_4) f_2(X_1, X_3, X_6) f_3(X_2, X_4, X_5) f_4(X_1)$$

1. Draw the corresponding FG.



2. Assume that X_1 is the hidden state that we are interested in. Write the formula for computing the marginal of X_1.

$$\mu_{f_4 \to x_1} = f_4(x_1)$$

$$\mu_{f_2 \to x_1} = \sum_{x_3, x_6} f_2(x_1, x_3, x_6)$$

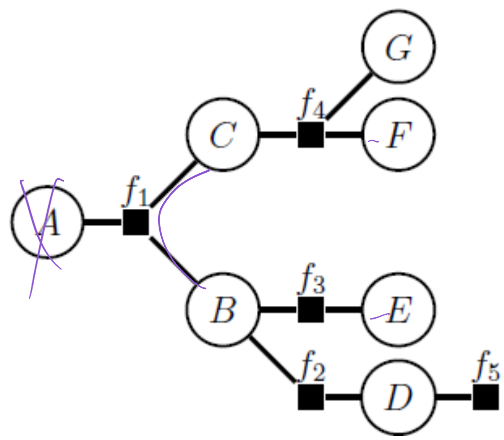
$$\mu_{f_4 \to x_1} = \sum_{x_4} \left[f_1(x_1, x_4) \sum_{x_2, x_5} f_3(x_2, x_4, x_5) \right]$$

$$P(x_1) = \mu_{f_4 \to x_1} \mu_{f_2 \to x_1} \mu_{f_4 \to x_1}$$

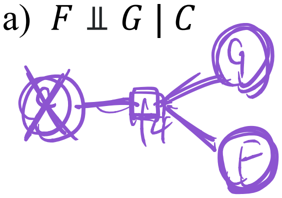
$$= f_4(x_1) \cdot \sum_{x_3, x_6} f_2(x_1, x_3, x_6) \cdot \sum_{x_4} \left[f_1(x_1, x_4) \sum_{x_2, x_5} f_3(x_2, x_4, x_5) \right]$$

Homework 4 | **Name: ____________________**
ECE/CS 498 DS Spring 2019 | **NetID: ____________________**
Issued: 04/17/19
Due: 04/24/19

Problem 3. For the Factor Graph given below, which of the following conditional independence relations is true? Justify your answer.



a) $F \perp\!\!\!\perp G \mid C$



No. G and F are not conditional independent since they are still connected.

b) $A \perp\!\!\!\perp G \mid C$

Yes. The path from A to G has node C which is observed.

c) $E \perp\!\!\!\perp F \mid B, A, D$

Yes. The path is E-B-A-C-F. Since B is observed. Then it's conditionally indepent.