

**ABIN Assignment 4**  
**Dr. Debarka Sengupta**  
**26/10/23**

**Guidelines**

1. Deadline: 2<sup>nd</sup> November 2023, 11:59 pm (Midnight).
2. Late submission: 3<sup>rd</sup> November 2023, 11:59 pm (Midnight) (50% of the obtained marks will be deducted).
3. Further late submissions after 3<sup>rd</sup> November will be awarded zero.
4. All coding assignments must be submitted as .ipynb files with proper comments.
5. For every question you should write a function with appropriate name and define the test case outside the function. The test case should be passed in the function as argument, and the function should return the result. **If any group does not follow it, they will be awarded zero without further evaluation.**

For example in Q1:

def Q1(<test case arguments>):

...code block.....

.....

return <output>

output = Q1(sample test case)

6. Standard IIIT-D plagiarism policy applies.
7. The assignments have to be submitted in the following manner:
  - (a) Create a single Jupyter notebook with proper demarcation of questions and responses (text, code, command, explanation, output, graphs). The accepted language is Python.
  - (b) A PDF file comprising the entire Jupyter Notebook and a separate Jupyter Notebook file needs to be uploaded (zip format) at the assignment link on Google Classroom before the deadline.
  - (c) One group should submit only from the registered submission ID during group formation. Rest of the members must turn in the assignments with private comments mentioning the name, roll, and submission id. Any violation of above will disqualify your submission from evaluation.
  - (d) The name of the PDF and jupyter notebook file should be a combination of the group number and assignment number. For example, group1 1.pdf, where 'group1' is the group number and '1' is the assignment number.
8. No shift of the deadline is allowed.

### Question 1 (5 Points)

Implement the Viterbi Algorithm for the following input-:

observations = ("normal", "cold", "dizzy")

states = ("Healthy", "Fever")

start\_p = {"Healthy": 0.6, "Fever": 0.4}

trans\_p = {

"Healthy": {"Healthy": 0.7, "Fever": 0.3},

"Fever": {"Healthy": 0.4, "Fever": 0.6},

}

emit\_p = {

"Healthy": {"normal": 0.5, "cold": 0.4, "dizzy": 0.1},

"Fever": {"normal": 0.1, "cold": 0.3, "dizzy": 0.6},

}

The code should output the following-:

- Probability of Hidden States ('Healthy' & 'Fever') for all three steps.
- Highest probability at step 3 from where we start backtracking.
- Most probable set of hidden states for the above-given input.

### Question 2 (5 Points)

Compute the probability of a string emitted by an HMM

Given: A string  $x$ , followed by the alphabet  $\Sigma$  from which  $x$  was constructed, followed by the states  $States$ , transition matrix  $Transition$ , and emission matrix  $Emission$  of an HMM ( $\Sigma$ ,  $States$ ,  $Transition$ ,  $Emission$ ).

Return: The probability  $Pr(x)$  that the HMM emits  $x$ .

Sample Input –

**xzyyzzzyzyy**

-----

**x y z**

-----

**A B**

-----

**A B**

**A 0.303 0.697**

**B 0.831 0.169**

-----

**x y z**

**A 0.533 0.065 0.402**

**B 0.342 0.334 0.324**

Sample Output –

1.1005510319694847e-06

### Question 3 (5 Points)

Perform a Multiple Sequence Alignment with a Profile HMM

Given: A string Text, a multiple alignment Alignment, a threshold  $\theta$ , and a pseudocount  $\sigma$ .

Return: An optimal hidden path emitting Text in  $HMM(Alignment, \theta, \sigma)$ .

Sample Input –

**AEFDFDC**

-----

**0.4 0.01**

-----

**A B C D E F**

-----

**ACDEFACADF**

**AFDA---CCF**

**A--EFD-FDC**

**ACAEF--A-C**

**ADDEFAAADF**

Sample Output –

M1 D2 D3 M4 M5 I5 M6 M7 M8