MACHINE INTELLIGENCE LABORATORY
Hidden Markov Model

Teaching Assistant
- Pranav Shankar

In this week's experiment, implement the Decoding Sequence Algorithm also known as the Vertibri Algorithm using Pytorch. This finds the most probable sequence of internal states that generate the observation.

You are provided with -
1. SRN.py
2. SampleTest.py

Use the template mentioned in SRN.py, complete the code snippet, and test your output with SampleTest.py.

Background and Deliverables

The Viterbi algorithm (VA) is a recursive optimal solution to the problem of estimating the state sequence of a discrete-time finite-state Markov process observed in memoryless noise. Many problems in areas such as digital communications can be cast in this form. Refer Refer

Initialization:

$$\nu_1(j) = \pi_j b_j(o_1)$$
$$bt_1(j) = 0$$

Recursion:

$$\nu_t(j) = \max_{i=1}^{N} \nu_{t-1}(i)\ a_{ij}\ b_j(o_t)$$
$$bt_1(j) = \operatorname{argmax}_{i=1}^{N} \nu_{t-1}(i)\ a_{ij}\ b_j(o_t)$$

Termination:

$$\text{Best Score: } P* = \max_{i=1}^{N} \nu_T(i)$$
$$\text{Best Path Start: } q_T* = \operatorname{argmax}_{i=1}^{N} \nu_T(i)$$

Viterbi Algorithm

1. You are provided with class HMM. The arguments used are -
   A: State transition matrix
   states: list of states
   emissions: list of observations
   B: Emission probabilities

2. The constructor and make_state_dictonary() methods are already implemented. The make_state_dictonary() does mapping between states and indexes. It creates two dictionaries -
   a. self.states_dict - is a list of iterable containing elements that represent states.
   b. Self.emissions - is a list of elements that represent emissions.

3. Your task is to write the code for the viterbi_algorithm() function. You must return the hidden states which are mostly likely to result in the given observations. The answer must be returned in the form of a Python list.

You can import any other python3.7 libraries and work with helper functions if needed. But ensure that implementation is done using Pytorch.
DO NOT CHANGE THE SKELETON OF THE CODE.

Submission

SampleTest.py -

Execute and check your output with the given test cases. Ensure all edge cases are satisfied too, before final submission.
Re-name the output file as YOUR_SRN.py
Run command -

python3 SampleTest.py --SRN YOUR_SRN