Assignment 7 Report

Description:

The main data structures used were numpy arrays. The Hidden Markov Model is first specified with numpy arrays to describe the state transition matrix, the observation probabilities matrix, the observed sequence, and the initial prior probabilities for the v_1 value. Since the domain values were restricted to the 1-10 integers, the state space is implicitly invoked by the index values (since the initial dimensions were that of a size 10 array) which returns the appropriate integer index positions corresponding to the domain values of x after compensation for zero indexing. In the observation probabilities for the edge cases of $v_t = 1$ and $v_t = 10$, these were interpreted in a modular fashion so that in the case of $v_t = 1$, the "noisy measurement" of equal probabilities for incrementing, decrementing or unchanged, were uniformly distributed with each outcome having a 1/3 probability and where the decrement v_t -1 was equal to 10. Similarly, for the other edge case of $v_t = 10$, the noisy measurement increment v_t+1 was interpreted to mean equal to 1. All other observation probabilities for the given state were set to zero since only the decrement, unchanged, or increment were specified as possibilities for the "noisy measurement". In order to optimize the code, the HMM model probabilities were converted to log scale (with an epsilon value of 1e-8), since earlier attempts at finding the maximally likely sequence led to numerical precision errors. In terms of challenges, the preliminary work in modeling the HMM and ensuring the proper indexing and dimensional conformity on the numpy arrays was very time consuming.

Requested Output:

Given the observation sequence:

[8,6,4,6,5,4,5,5,7,9]

The most likely sequence of values $\boldsymbol{v_1, v_2, \ldots, v_{10}}$ is:

[7,6,5,6,5,4,5,6,7,8]