Exercises

0. In a terminal window, move into the directory just above the computational-linguistics subdirectory that in turn contains your solution to Exercise 2 — the (computational-linguistics HMM) library file. Start DrRacket from the command line. In the new window, specify the R7RS language and import the (scheme base) and (computational-linguistics HMM) libraries. Run this (empty) application program to ensure that the libraries can be imported correctly.

1. Given a hidden Markov model and an observation sequence, we’ll want to construct a trellis containing, for each state $s$ of the model and each number $k$ of completed observations from 0 to the length of the observation sequence, the most probable sequence of states leading to state $k$ (given the model and the first $k$ observations) and the inferred probability of that sequence.

   Note that it will be most efficient to maintain the state sequences as reversed lists, in which the most recent state is added at the beginning of the list, and successive list elements indicate states moving backwards in time.

   It would also be a good idea to represent probabilities by their logarithms so as to avoid floating-point underflow, as described in section 9.4.1 of the text.

   Devise and implement a trellis structure to hold all this data.

2. We’ll consider a hidden Markov model that generates as observables the phonemes /t/, /k/, /a/, /o/, and /r/ and a pseudo-phoneme /∅/ that serves to indicate the end of each syllable. The model will have four states: 0 (syllable-initial), 1 (syllable-medial), 2 (syllable-final), and 3 (syllable-complete). Here is the stochastic transition matrix for these states:

   $\begin{array}{ccccc}
   & 0 & 1 & 2 & 3 \\
   0 & 0.00 & 0.35 & 0.65 & 0.00 \\
   1 & 0.00 & 0.00 & 0.70 & 0.30 \\
   2 & 0.00 & 0.00 & 0.20 & 0.80 \\
   3 & 0.90 & 0.10 & 0.00 & 0.00 \\
   \end{array}$

   Here is the matrix of observation probabilities. This is an arc-emission model, so a separate probability for each phoneme is associated with each transition that has a non-zero entry in the stochastic transition matrix.
I don’t care what observation probabilities you assign for other transitions, since they will never occur in practice. Assigning probability 1.0 to /∅/ and 0.0 to all the other phonemes would be as good a choice as any.

The initialization vector for the model should indicate that it starts in state 0 with probability 0.9 and in state 1 with probability 0.1.

Using your library, build and name a hidden Markov model using these parameters.

3. Our goal for today is to determine the most probable sequence of states of the model described in the preceding exercise for the observed sequence of phonemes /ar∅tor∅a∅/, with syllable breaks as indicated by the positions of the ∅ pseudophoneme.

Define a Scheme procedure that takes as arguments a model and a sequence of phonemes and constructs and returns a trellis, using the initialization and induction formulas described on page 332 of our textbook (under the heading “Viterbi algorithm”) to fill in the successive entries. Instead of a backtrace that gives just the preceding state in the most-probable sequence of states, include the entire sequence of states in each trellis entry.

4. Define a Scheme procedure that recovers from the trellis that the procedure in the previous exercise constructed the most probable sequence of states for the given model and the observed phoneme sequence, and the probability of that sequence.

In returning or reporting these results, your procedure should un-reverse the sequence of states so that the caller sees them in chronological order.

What is the most probable sequence of states for the observation sequence /ar∅tor∅a∅/, and how probable is that sequence of states?