COMS 4701: Artificial Intelligence

Homework 4 Sample Solutions and Feedback

Problem 1

First, we must write down the transition matrix, which is given by

$$T = \begin{pmatrix} 0.5 & 0.25 & 0.25 & 0 & 0 & 0 \\ 0.25 & 0.5 & 0 & 0.25 & 0 & 0 \\ 0.25 & 0 & 0.5 & 0.25 & 0 & 0 \\ 0 & 0.25 & 0.25 & 0 & 0.25 & 0.25 \\ 0 & 0 & 0 & 0.25 & 0.75 & 0 \\ 0 & 0 & 0 & 0.25 & 0 & 0.75 \end{pmatrix}$$

1. The normalized eigenvector of T corresponding to eigenvalue 1 is $(\frac{1}{6}, \frac{1}{6}, \frac{1}{6}, \frac{1}{6}, \frac{1}{6}, \frac{1}{6})$.

2. The observation matrix for both computations is $O = diag(0, 0.25, 0.25, 0, 0.5, 0.5)$.

   $\Pr(X_1 \mid e_1) = (0, \frac{1}{6}, \frac{1}{6}, 0, \frac{1}{3}, \frac{1}{3})$, obtained by multiplying $O \cdot T \cdot \Pr(X_0)$ and normalizing.

   $\Pr(X_2 \mid e_1, e_2) = (0, \frac{1}{14}, \frac{1}{14}, 0, \frac{3}{7}, \frac{3}{7})$, obtained by multiplying $O \cdot T \cdot \Pr(X_1|e_1)$ and normalizing.

3. There are four state sequences in which the robot can twice observe #: (B,B), (C,C), (E,E), (F,F). The joint distribution is proportional to the values of

   $\Pr(x_1, x_2, e_1, e_2) = \Pr(x_1) \Pr(e_1|x_1) \Pr(x_2|x_1) \Pr(e_2|x_2) \propto \Pr(x_1|e_1) \Pr(x_2|x_1) \Pr(e_2|x_2)$.

   $\Pr(B, B \mid e_{1:2}) = \frac{1}{14}$, $\Pr(C, C \mid e_1, e_2) = \frac{1}{14}$, $\Pr(E, E \mid e_1, e_2) = \frac{3}{7}$, $\Pr(F, F \mid e_1, e_2) = \frac{3}{7}$

4. X_1 and X_2 are not independent, since $\Pr(X_2 \mid X_1, e_1, e_2) \neq \Pr(X_2 \mid e_1, e_2)$ in general. For example, the RHS is nonzero for X_2 being B, C, E, or F, but the LHS is zero if X_1 takes a different value from X_2. Intuitively, the robot's first state location restricts where the robot can be after one transition. The most likely state sequences are either (E,E) or (F,F).

POS Tagging

Coding

1. Make sure to handle unseen words correctly by returning the most likely tag of '#UNSEEN' for those cases.

2. The elapse time step of Viterbi returns both a belief state (using max) as well as a list of prior tags (using argmax). Both should be vectors with the same number of components as the number of states.

3. No major comments here.

4. The backward pass over the list of prior tag listss was the trickest. Make sure that you are starting with the last list and moving backward. The first list can be disregarded, since that points to X_0, which we are not interested in. Finally, as you move backward, make sure that you update each successive index to the value extracted from the previous iteration. A concise way of doing so would be something like the line `tag = (pointers[i])[tag]`.

5. As we loop over obs_counts, we should only consider entries that have a total count of 1 (this is the definition of hapax legomena). The corresponding tag count in hapax is incremented only when that condition is met.

Responses

1. 0.4% of all unseen words in the training data have a POS tag of SYM.

2. The word ”round“ has multiple parts of speech, and the context of the word sequences that it appears in (both prior to and after its appearance) helps determines the most likely one.

3. Nouns and proper nouns occur most often because they exhibit great diversity in the English language. It makes sense that many nouns only occur once, and the same reasoning can also be applied to proper nouns, many of which are names of people and places. Conversely, particles and conjunctions occur least often, as most are used fairly often and there are very few rare words that have these POS tags.