DATA.STAT.840 Statistical Methods for Text Data Analysis

Exercises for Lectures 7-8: Hidden Markov models

✓

Exercise 7.1: The Forward-Backward algorithm.

Consider a HMM model with the following vocabulary of 14 words:
{a, the, over, beside, near, quick, brown, lazy, jumps, runs, walks, fox, dog, cat}
and five states (z=1, z=2, z=3, z=4, z=5), where the distribution of the initial state is uniform and the states have the following emission distributions:

	a	the	over	beside	near	quick	brown	lazy	jumps	runs	walks	fox	dog	cat
$\{\beta_1(w)\}=$	{0.6,	0.4,	0,	0,	0,	0,	0,	0,	0,	0,	0,	0,	0,	0}
$\{\beta_2(w)\}=$	{0,	0,	0.2,	0.4,	0.4,	0,	0,	0,	0,	0,	0,	0,	0,	0}
$\{\beta_3(w)\}=$	{0,	0,	0,	0,	0,	0.5,	0.3,	0.2,	0,	0,	0,	0,	0,	0}
$\{\beta_4(w)\}=$	{0,	0,	0,	0,	0,	0,	0,	0,	0.1,	0.4,	0.5,	0,	0,	0}
$\{\beta_5(w)\}=$	{0,	0,	0,	0,	0,	0,	0,	0,	0,	0,	0,	0.3,	0.4,	0.3}

and the following transition probabilities:

$$\{\theta_{z_t|z_{t-1}}\}= \begin{array}{ccccccc} & & & & z_t & & \\ & & 1 & 2 & 3 & 4 & 5 \\ & 1 & \{\ 0, & 0, & 0.5, & 0, & 0.5, \\ z_{t-1} & 2 & 1, & 0, & 0, & 0, & 0, \\ & 3 & 0, & 0, & 0.3, & 0, & 0.7, \\ & 4 & 0, & 1, & 0, & 0, & 0, \\ & 5 & 0, & 0.5, & 0, & 0.5, & 0,\ \} \end{array} .$$

a) Use the forward-backward algorithm to compute the probability of the sentence "the quick fox jumps over a dog".
Report your computation steps and your answer.

✓

Exercise 7.2: The Viterbi algorithm.

Consider a HMM model with the following vocabulary of 17 words:
{a, the, I, you, can, will, call, own, take, book, round, claim, car, hotel, new, great}
and five states (z=1, z=2, z=3, z=4, z=5), where the distribution of the initial state is uniform and the states have the following emission distributions:

	a	the	I	you	can	will	call	own	take	book	round	claim	car	hotel	new	great
$\{\beta_1(w)\}=$	{0.4,	0.3,	0.2,	0.1,	0,	0,	0,	0,	0,	0,	0,	0,	0,	0,	0,	0}
$\{\beta_2(w)\}=$	{0.3,	0.4,	0,	0.3,	0,	0,	0,	0,	0,	0,	0,	0,	0,	0,	0,	0}
$\{\beta_3(w)\}=$	{0,	0,	0,	0,	0.1,	0.15,	0.15,	0.15,	0.15,	0.1,	0.1,	0.1,	0,	0,	0,	0}
$\{\beta_4(w)\}=$	{0,	0,	0,	0,	0,	0,	0,	0,	0,	0,	0.3,	0,	0,	0,	0.4,	0.3}
$\{\beta_5(w)\}=$	{0,	0,	0,	0.05,	0.05,	0.05,	0.1,	0,	0,	0.15,	0.1,	0.15,	0.2,	0.15,	0,	0}

and the following transition probabilities:

$$\{\theta_{z_t|z_{t-1}}\}= \begin{array}{ccccccc} & & & & z_t & & \\ & & 1 & 2 & 3 & 4 & 5 \\ & 1 & \{\ 0, & 0, & 0.3, & 0.3, & 0.4, \\ z_{t-1} & 2 & 0, & 0, & 0, & 0.4, & 0.6, \\ & 3 & 0, & 0.75, & 0.25, & 0, & 0, \\ & 4 & 0, & 0, & 0, & 0, & 1, \\ & 5 & 0, & 0, & 1, & 0, & 0,\ \} \end{array} .$$

In this HMM model words can act in multiple roles, e.g. 'book' be used as a noun or a verb, so there could be more than one way to parse some sentences.
Use the Viterbi algorithm to compute the most likely state sequence corresponding to the observed word sequence "you claim you can book a round hotel".
Report your computation steps and your answer.

(exercises continue on the next page)

Exercise 7.3: The Baum-Welch algorithm.

Consider the HMM model of exercise 7.1. Suppose the sentence "the quick fox jumps over a dog" is the only training data. Suppose the HMM parameter values listed in exercise 7.1 are initial values to be further optimized. Perform one iteration of the Baum-Welch algorithm to optimize the parameters for the training data.

Report your computation steps and your answer.

Exercise 7.4: HMM modeling of sentences.

The file "hmm_sentences.txt" provided with this exercise pack contains 1000 sentences, one on each line, which have been generated from a relatively simple probabilistic context free grammar with a limited vocabulary. Some example sentences are:

✓

- i will explain .
- they are cautious , however , i am strong and then the cautious robot will feed me , but you will feed a small fox and he is strong , however , is the fox strong ?
- they are wise , but then a insightful strong cat is cautious , however , are you cautious ?
- where will you explain ?
- the bird is insightful .

These sentences contain statements and questions which can be concatenated together, and each statement/question can contain different kinds of subjects and possibly also objects. In this exercise, the idea is to see how well such simplified sentences (which do not represent the full variety of language) can be modeled by a hidden Markov model.

Use the **HMMlearn** Python library, or another library of your choice in your chosen programming language, to learn a hidden Markov model for this set of example sentences. Use either 5 or 10 hidden states. Inspect the resulting emission probabilities of the states, and the transition matrices between the states. Do the states seem to correspond to meaningful properties of the simplified language?

Report your code, the resulting emission and transition probabilities, and your analysis.