HMM Programming Tasks

1 Remarks

1. In this excercise you will will work on a practical example for the application of HMMs. Specificially, you will work on both the evaluation and the decoding problem, given three already trained models.

2. The three given HMMs model the daily weather conditions for the 31 days of July for the locations Hamburg, Munich, and Brocken, respectively.

3. An observation (for one day) is represented by a 3-tuple, consisting of a value for maximum temperature (degrees Celsius), rainfall (millimeters), and sun hours.

4. There are five states numbered from 0 to 4, respectively: initial, sunny, cloudy, rainy, and final.

5. The transition matrix is of size 5x5 with a layout according to the lecture. However, be aware of the following difference: Since we are looking at a sequence of exactly 31 emissions (deterministic length), the probability for transitioning into the final state needs to be time-dependent. Specifically, for the transitions following each of the first 30 emissions, the probability needs to be zero for all originating states. After the 31st emission, the probability for transitioning into the final state needs to be exactly one. Consequently, you will be given two different transition matrices.

6. The observation probabilities are given as Gaussian mixture models, one for each emitting state. For information on how to use the models look up the functions on the class *sklearn.mixture.GaussianMixture*. The three models are contained in an object list ordered according to the given state numbering.

2 Decoding

1. The decoding problem aims at reconstructing the state sequence from the observed data. For the given context, an estimation of the original weather condition labels (sunny, cloudy, or rainy) for each day is to be performed.

2. Open the notebook *hmm_ex_decode_diy*.

3. Fill in the function *decode*, which implements the Viterbi algorithm (see slide 32) for the given problem.

4. Create a state and observation sequence from one of the given models using the provided function *hmm_sample_month*.

5. Apply the viterbi algorithm to the observation sequence.

6. Evaluate the decoding accuracy by determining how many (in percent) states you estimated correctly.
7. Repeat the previous steps for the other models and other sequences. Also try decoding an observation sequence with a model that was not used for its generation. Discuss your results. Since each sequence is generated by an underlying random process, you may want to look at the average performance.
8. What problem might occur when straightforwardly implementing the equations from the lecture (as done in this exercise)?

3 Evaluation

1. The evaluation problem aims at determining the probability that a given observation sequence $\mathbf{X}$ was generated by the specified model λ. For the given context, the evaluation may be used to match an observation sequence of unknown origin to its generating model, thus, detemining whether the data was recorded in Hamburg, Munich or on top of Brocken.
2. Open the notebook *hmm_ex_evaluate_diy*.
3. Fill in the function *evaluate*, which implements the computation of $p(\mathbf{X}|\lambda)$.
4. Test your implementation with the provided test section.
5. Using this function, try determining which of the recorded observation sequences belongs to which of the three possible locations.
6. Why can it be advantageous to use log-likelihood ratios for comparison?