

Task 7: Training the HMM

Your aim is to calculate the parameters of the HMM which you will use to predict when the dice is switched between fair and loaded. The prediction itself (using the Viterbi algorithm) will be carried out in the following session. In the third session, you will use the same approach (and much the same code) to predict properties of real biological sequence data.

1. Download and examine the dice dataset. This dataset contains a number of files corresponding to sequences of states. The first line in each is the observed sequence, which corresponds to the numbers shown on the dice (`DiceRoll.java`), and the second line is the hidden sequence, corresponding to whether the fair or loaded dice was used (`DieType.java`). This can be used as labelled training data for an HMM. `Exercise7Tester.java` picks out a training set from the data set for you. In the next task you will use cross-validation.
2. Your task in this session is to read the sequences from the files and use them to estimate the transition probabilities, emission probabilities and initial state probabilities of the HMM that generated them. Note that, although in this task there are only two states for the dice (fair and loaded), in the biological task, there will be more states. So you may want to write your code to be general enough to cope with more than two states now. You can use the `HMMDataStore` class to store your sequence pairs and load the files.
3. Your code should return a `HiddenMarkovModel<DieRoll, DieType>` object constructed from the `HiddenMarkovModel.java` file provided to you. Its constructor takes three matrices: transition matrix (A), emission matrix (B) and initial probabilities matrix (C).
4. The transition matrix (A) consists of the estimates of transition probabilities.

Example: What is the probability that a roll with a fair dice will be followed by a roll with the loaded dice?

$$P(F \rightarrow L) = \frac{\text{count}(F \rightarrow L)}{\text{count}(\text{all transitions from F})}$$

For this dataset there are two states, fair and loaded. You therefore need the probabilities of the following state sequences:

-
- a. fair fair
 - b. fair loaded
 - c. loaded fair
 - d. loaded loaded

-
4. The emission probabilities (B) measure how probable it is to obtain a particular roll given the dice type. You can calculate it by counting the number of times each dice roll appears for the given state (fair or loaded) and dividing by the count of that state.
 5. The initial probabilities for the matrix C can be calculated by counting how often the fair state begins the sequence and how often the loaded one.
 6. Once you have calculated these probabilities, make sure that the numbers are plausible.

Note: *die* and *dice* are both acceptable singular forms of the word *dice* in English. As computational linguists, we would be, however, quite happy if *die* died out in this usage.