Exercises – Algorithms for Genome Research

Universität Bielefeld, WS 2014, Dr. Pedro Feijao

http://wiki.techfak.uni-bielefeld.de/gi/Teaching/2014winter/AlgoGR

Exercise List 2 — 31.10.2014

Discussion of exercises on: 07.11.2014

Exercise 1 For a given HMM the probability of observing a sequence x is $P(x) = P(x_1) \prod_{i=2}^{L} a_{x_{i-1}x_i}$. The sum of the probabilities of all possible sequences of length L can be written as

$$\sum_{\{x\}} P(x) = \sum_{x_1} \sum_{x_2} \cdots \sum_{x_L} P(x_1) \prod_{i=2}^{L} a_{x_{i-1}x_i}$$

Show that this is equal to 1.

Exercise 2 Download the archive **ExerciseList2.zip** containing Java code for this exercise sheet. Implement the

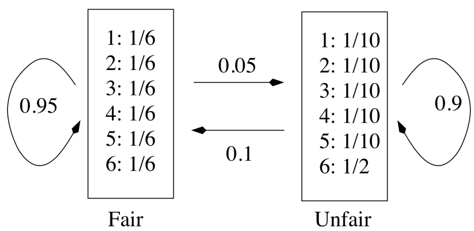
(a) *Viterbi*,

(b) *Forward*, and

(c) *Backward* Algorithm

in the corresponding classes by following the instructions.

Exercise 3 *The occasionally dishonest casino.* A casino uses a fair die most of the time, but occasionally switches to an unfair die. The unfair die has probability 0.5 of a six and probability 0.1 for the numbers one to five. Assume that the casino can be modeled as a hidden Markov process, in which switching from a fair to an unfair die happens with probability 0.05 before each roll, and that the probability of switching back is 0.1, as illustrated by the following figure:



(a) What is the probability $P(x)$ of emitting the sequence $x = 5\ 1\ 6\ 5\ 4\ 5\ 5\ 4\ 2$?

(b) The Java code of the previous exercise provides a method to generate a sequence of states and symbols from a hidden Markov model. Follow the instructions in class `ExerciseList2` to compute the most probable path from a generated symbol sequence of length 100 using the *Viterbi* Algorithm and to compare it to the true state path.

(c) Implement a method to compute the posterior probabilities for a given emission sequence x using the forward and backward algorithm. Plot the posterior probability of being in the state corresponding to the fair die in the casino example for a generated emission sequence of length 100.

Exercise 4 Using the Java code of the previous exercise, generate an emission sequence of length 500 and inspect the most probable path returned by the implemented *Viterbi* algorithm as well as the posterior probabilities of the state path. Other than the path and emission sequence being longer, are the results any different from the previous simulation with sequence length 100? If so, can you explain why?

Exercise 5 (bonus) Construct an HMM and an emission sequence in which the posterior decoding $\hat{\pi}$ is not a legitimate state path.