Advanced AI Techniques (WS05)

Exercise sheet 5

Deadline: Tuesday, 13th, 2005

Suppose we want to determine the average annual temperature at a particular location on earth over a series of years. To make it interesting, suppose the years we are concerned with lie in the past, and no temperature measurements from that time are available. However, we have indirect evidence of the temperatures in the past by looking at the tree rings of trees cut down today.

To simplify the problem, we only consider two annual temperatures, **hot** (h) and **cold** (c), and three different tree ring sizes, **small**, **medium** and **large**, or s, m and l. Suppose that modern evidence indicates that the probability of a **hot** year to be followed by another **hot** year is **0.7** and the probability that a **cold** year is followed by another **cold** year is **0.6**. The information so far can be summarized as:

$$\begin{array}{c} \\ h \\ c \end{array} \begin{array}{c} \begin{array}{cc} h & c \end{array} \\ \left[\begin{array}{cc} 0.7 & 0.3 \\ 0.4 & 0.6 \end{array}\right] \end{array}$$

Also suppose that current research indicates the following correlation between the size of tree growth rings and temperature:

$$\begin{array}{c} \\ h \\ c \end{array} \begin{array}{c} \begin{array}{ccc} s & m & l \end{array} \\ \left[\begin{array}{ccc} 0.1 & 0.4 & 0.5 \\ 0.7 & 0.2 & 0.1 \end{array}\right] \end{array}$$

Hidden Markov models are good choice in this situation because the states h and c are hidden since we cannot directly observe the temperature in the past. The transition matrix A and the observation matrix B are

$$A = \begin{pmatrix} 0.7 & 0.3 \\ 0.4 & 0.6 \end{pmatrix}, B = \begin{pmatrix} 0.1 & 0.4 & 0.5 \\ 0.7 & 0.2 & 0.1 \end{pmatrix} .$$

Assume that there is additional evidence that the initial state distribution is

$$\pi = (0.6, 0.4) ,$$

i.e., a **hot** year is apriori more likely. Now consider a particular four-year period of interest where we observe the series of tree rings

$$s, m, s, l .$$

Exercise 1 (4 points) As discussed in the lecture, the *forward procedure* is a *dynamic programming* approach for efficiently evaluating observation sequences with *hidden Markov model* (HMMs). In order to compute $P(o_1, o_2, \ldots, o_n \mid M)$ for a given observation sequence $o_1, o_2, \ldots, o_n$ given a HMM M, a dynamic programming approach is employed. More precisely, the so called *forward probability*

$$P(o_1, o_2, \ldots, o_t, q_t = s \mid M)$$

is iteratively computed for $t = 1, 2, \ldots, n$. In the formula, $q_t = s$ denotes that the system is in states s at time t.

As shown in the lecture, this leads to the following iterative formulae:

1. Initialization: $\alpha_1(s) = \pi_s \cdot b_s(o_1)$
2. Induction: $\alpha_{t+1}(s) = \left[\sum_{s'} \alpha_t(s') \cdot a_{s's}\right] \cdot b_s(o_{t+1})$
3. Termination: $P(o_1 o_2 \ldots o_n \mid M) = \sum_s \alpha_n(s)$

Compute the probability of s, m, s, l using the forward procedure, and list all α-values.

Exercise 2 (4 points) The probability $P(o_1 o_2 \ldots o_n \mid M)$ can also be computed in a backward manner. The *backward procedure* computes the so called *backward probability*:

$$\beta_t(s) = P(o_{t+1}, o_{t+2}, \ldots, o_n \mid q_t = s, M) \ .$$

for $t = n, n-1, \ldots, 0$ as follows:

1. Initialization: $\beta_n(s) = 1$
2. Induction: $\beta_t(s) = \sum_{s'} a_{ss'} \cdot b_{s'}(o_{t+1}) \cdot \beta_{t+1}(s')$
3. Termination: $P(o_1 o_2 \ldots o_n \mid M) = \sum_s \pi_s \cdot b_{s'}(o_1) \cdot \beta_1(s)$

The correctness of the iterative formulae for calculating the α-values has been veryfied in the lecture. Do the same for the iterative formulae for the β-values given above.

Exercise 3 (4 points) Having a forward procedure, it is straightforward to decode an observation sequence $o_1, o_2, \ldots, o_n$ given a HMM M, i.e., computing the hidden state sequence $s_1, s_2, \ldots, s_n$ which most likely generated $o_1, o_2, \ldots, o_n$. Instead of summing over all $\alpha_t(s)$, one basically selects the maximum. This is what the so-called *Viterbi* algorithm does. Decode s, m, s, l.