Assignment 3
EQ2341 Pattern Recognition and Machine Learning

Names
Oriol Closa and Clara Escorihuela Altaba

Date
January 23, 2022

1 Introduction

This assignment aims at implementing and verifying the forward algorithm, which allows to calculate the probability density $P[\underline{X} = \underline{x}|\lambda]$, when an HMM model ($\lambda$) and a finite or infinite observed feature sequence $\underline{x}$ = (x_1,x_2,...,x_t) are defined.

2 Code implementation

2.1 Verifying the forward algorithm

The following section shows the implementation of the forward algorithm based on the methodology exposed by Mark Stamp in «A Revelion Introduction to Hidden Markov Models»[1]. The forward or α-pass algorithm is an iterative algorithm which defines the forward variable as the probability of observing the sequence of observations until time t, being in state q_i and time t. Equation 1, eq. 2 and eq. 3 expose the mathematical methodology of this algorithm.

Given $\lambda = (q, a_{ij}, bij)$ and O_t

1. For $i = 0, 1, ..., N - 1$

$$\alpha_t(i) = q_i b_i(O_0) \tag{1}$$

2. For $t = 1, 2, ..., T - 1$ and $i = 0, 1, ..., N - 1$

$$\alpha_t(i) = \left(\sum_{j=0}^{N-1} \alpha_{t-1}(j) a_{ij} \right) b_i(O_0). \tag{2}$$

3. Finally,

$$P(O|\lambda) = \left(\sum_{i=0}^{N-1} \alpha_{t-1}(i) \right) \tag{3}$$

Our finale algorithm is almost equal to the detailed by Mark Stamp [1] with the exception of the observation matrix (B): The latter presents an HMM model where $\lambda = (A, B, q)$ are defined, while in our case, we have $\lambda = (A, q)$ and the probability distribution for $State_1$ and $State_2$, therefore we must *"simulate"* a B matrix based on these distributions.

Function `@GaussD/prob` allows us to accomplish that: It looks for the y_{value} of the x_t observation in each Gaussian Distribution and scales them, with regard to the maximum value, in order to find a relation between both parameters. This have been achieved by the `stats.norm.pdf` function, available in the `stats` package in Python.

The following finite HMM model has been applied in order to compare the obtained and the expected results of a given sequence. Morevoer, figure 1 presents the code for the `main` function.

$$q = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \quad A = \begin{pmatrix} 0,9 & 0,1 & 0 \\ 0 & 0,9 & 0,1 \end{pmatrix} \quad B = \begin{pmatrix} \mathcal{N}(0,\ 1) \\ \mathcal{N}(3,\ 2) \end{pmatrix} \tag{4}$$

2.2 Logarithm Probability Verification

The following section aims at calculating the $logprobability$ of a given sequence, which corresponds to the multiplication of the forward scale factors obtained from the forward algorithm. The log probability is obtained by applying the logarithm this multiplication, which results in the sumation of the $ln(c_t)$.

```
q = np.array([1.0, 0.0])
a = np.array([np.array([0.9, 0.1, 0.0]), np.array([0.0, 0.9, 0.1])])
obs = np.array([-0.2, 2.6, 1.3])


g1 = GaussD(means=[0.0], stdevs=[1.0])
g2 = GaussD(means=[3.0], stdevs=[2.0])
prob_obs = [g.prob(obs) for g in [g1,g2]]
mc = MarkovChain(initial_prob=q, transition_prob=a, dist_obs=prob_obs, obs=obs, scaling=True)
h = HMM(mc, [g1, g2])
a_ts, c_ts = mc.forward()
log_prob = h.logprob(c_ts)
print("A_hat = ",a_ts, "C_ts = ", c_ts)
print("LogProbability = ", log_prob)

mc = MarkovChain(initial_prob=q, transition_prob=a, dist_obs=prob_obs, obs=obs, scaling=False)
h = HMM(mc, [g1, g2])
a_ts, c_ts = mc.forward()
log_prob = h.logprob(c_ts)
print("A_hat = ",a_ts, "C_ts = ", c_ts)
print("LogProbability = ", log_prob)
```

Figure 1: Forward and log probability calls for scaled and not scaled factors.

Finally, equation 5 and eq. 6 expose the results of the forward variables ($alpha_{hat}$), the forward scale factors (c_t), and the results while applying the `logpro` function with and without scaling the probability observations. We can observed that the forward scale factors vector (c_t) have one element more, because the final state is the $exit$ state, which does not give an observation.

It is also interesting to highlight the difference in the forward scale factors when the probabilities of the x_t observations are scaled or not.

1. With scaled observation probabilities.

$$A_{hat} = \begin{pmatrix} 1 & 0,38 & 0,419 \\ 0 & 0,615 & 0,581 \end{pmatrix} \qquad c_{ts} = \begin{pmatrix} 1 & 0,1625 & 0,8256 & 0,058 \end{pmatrix} \qquad logProb = -4,853 \tag{5}$$

2. Without scaled observation probabilities.

$$A_{hat} = \begin{pmatrix} 1 & 0,38 & 0,419 \\ 0 & 0,615 & 0,581 \end{pmatrix} \qquad c_{ts} = \begin{pmatrix} 0,391 & 0,032 & 0,142 & 0,059 \end{pmatrix} \qquad logProb = -9,188 \tag{6}$$

References

1. STAMP, Mark. A revealing introduction to hidden Markov models. In: Department of Computer Science San Jose State University, 2004, pp. 26–56.