

A1 - HMM

Artificial Intelligence

Franco Ruggeri

December 18, 2019

1 Grade E-D

1.1 Question 1

$$\mathbf{A} = \begin{bmatrix} 0.5 & 0.5 \\ 0.5 & 0.5 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 0.1 & 0.9 \\ 0.5 & 0.5 \end{bmatrix}, \quad \pi = [0.5 \quad 0.5] \quad (1)$$

1.2 Question 2

$$P(X_6) \mathbf{A} = P(X_7) \quad (2)$$

1.3 Question 3

$$P(X_7) \mathbf{B} = P(O_7) \quad (3)$$

1.4 Question 4

$$\begin{aligned} P(O_{1:t} = o_{1:t}, X_t = x_i) &= P(O_t = o_t, O_{1:t-1} = o_{1:t-1}, X_t = x_i) \\ &= \{\text{product rule}\} \\ &= P(O_t = o_t | X_t = x_i, O_{1:t-1} = o_{1:t-1}) P(X_t = x_i, O_{1:t-1} = o_{1:t-1}) \\ &= \{\text{conditional independence}\} \\ &= P(O_t = o_t | X_t = x_i) P(X_t = x_i, O_{1:t-1} = o_{1:t-1}) \end{aligned} \quad (4)$$

1.5 Question 5

- δ has TxN elements
- $\delta^i dx$ has $(T - 1)xN$ elements (no predecessor for $t = 0$)

1.6 Question 6

$$\begin{aligned} P(X_t = x_i, X_{t+1} = x_j | O_{1:T} = o_{1:T}) &= \{\text{definition of conditional probability}\} \\ &= \frac{P(X_t = x_i, X_{t+1} = x_j, O_{1:T} = o_{1:T})}{P(O_{1:T} = o_{1:T})} \end{aligned} \quad (5)$$

The denominator of (5) can be computed using the forward algorithm as $\sum_{k=1}^N \alpha_T(k)$. This term represents a normalization factor.

2 Grade C

2.1 Question 7

According to [1], a possible distance measure between two HMMs is:

$$D(\lambda_1, \lambda_2) = \frac{1}{T} [\log P(O_{1:T} | \lambda_1) - \log P(O_{1:T} | \lambda_2)] \quad (6)$$

It can be used to define the convergence of the algorithm; that is, if the distance between the result of Baum-Welch algorithm and the generating HMM is little, the algorithm has converged.

With 1000 observations:

$$\mathbf{A} = \begin{bmatrix} 0.7 & 0.1 & 0.29 \\ 0.1 & 0.81 & 0.09 \\ 0.19 & 0.3 & 0.51 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 0.69 & 0.23 & 0.08 & 0.01 \\ 0.07 & 0.41 & 0.28 & 0.24 \\ 0 & 0 & 0.35 & 0.65 \end{bmatrix}, \quad D(\lambda_1, \lambda_2) \approx 0.00592 \quad (7)$$

With 10000 observations:

$$\mathbf{A} = \begin{bmatrix} 0.69 & 0.04 & 0.26 \\ 0.12 & 0.75 & 0.14 \\ 0.15 & 0.26 & 0.59 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 0.71 & 0.19 & 0.1 & 0 \\ 0.1 & 0.42 & 0.31 & 0.17 \\ 0.03 & 0.17 & 0.19 & 0.61 \end{bmatrix}, \quad D(\lambda_1, \lambda_2) \approx 0.00079 \quad (8)$$

As we can see, more observations give a better convergence.

2.2 Question 8

Assuming no prior knowledge, a good initialization is:

$$a_{ij} \approx 1/N, \quad b_{ij} \approx 1/K, \quad \pi_i \approx 1/N \quad (9)$$

Because, being in the middle, it is more probable to reach the global maximum instead of getting stuck in a local one.

So, using this initialization the result after learning with 1000 observations is:

$$\mathbf{A} = \begin{bmatrix} 0.7 & 0.29 & 0.01 \\ 0.19 & 0.51 & 0.3 \\ 0.1 & 0.09 & 0.81 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 0.69 & 0.23 & 0.08 & 0.01 \\ 0 & 0 & 0.35 & 0.65 \\ 0.07 & 0.41 & 0.28 & 0.24 \end{bmatrix}, \quad D(\lambda_1, \lambda_2) \approx 0.00592 \quad (10)$$

that is exactly the same obtained in question 7 with the initialization close to the generating model.

We can notice that the states 2 and 3 are swapped (i.e. X_2 of the learned model is X_3 of the generating one and vice versa). This is a problem if we want to compute an element-by-element distance (e.g. Euclidean distance), but if we use the distance measure defined in question 7 it does not matter.

2.3 Question 9

More hidden states require more observations, as the Baum-Welch algorithm is based on statistics on how frequent transitions/emissions are and more parameters require more data to have significant statistics.

5 states:

$$\mathbf{A} = \begin{bmatrix} 0.47 & 0.22 & 0.31 & 0 & 0 \\ 0 & 0.46 & 0 & 0.37 & 0.17 \\ 0 & 0.17 & 0.45 & 0.16 & 0.21 \\ 0.32 & 0 & 0.01 & 0.66 & 0 \\ 0 & 0 & 0.29 & 0 & 0.7 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 0.43 & 0.57 & 0 & 0 \\ 0 & 0.4 & 0.6 & 0 \\ 0 & 0 & 0.32 & 0.68 \\ 0 & 0.24 & 0.28 & 0.48 \\ 0.75 & 0.19 & 0.06 & 0 \end{bmatrix}, \quad D(\lambda_1, \lambda_2) \approx 0.016819 \quad (11)$$

2.4 Question 10

Initializing with (exact) uniform distribution, the Baum-Welch algorithm produces:

$$\mathbf{A} = \begin{bmatrix} 0.33 & 0.33 & 0.33 \\ 0.33 & 0.33 & 0.33 \\ 0.33 & 0.33 & 0.33 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 0.24 & 0.25 & 0.24 & 0.27 \\ 0.24 & 0.25 & 0.24 & 0.27 \\ 0.24 & 0.25 & 0.24 & 0.27 \end{bmatrix} \quad (12)$$

As you can see, the algorithm gets stuck in the initial point, that is a local maximum. Indeed, if the initial \mathbf{B} is uniform, the observations give no information.

Initializing with a diagonal \mathbf{A} matrix and $\pi = [0, 0, 1]$, we get:

$$\mathbf{A} = \begin{bmatrix} NaN & NaN & NaN \\ NaN & NaN & NaN \\ NaN & NaN & NaN \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} NaN & NaN & NaN & NaN \\ NaN & NaN & NaN & NaN \\ NaN & NaN & NaN & NaN \end{bmatrix} \quad (13)$$

Finally, initializing the matrices close to the solution guarantees a good convergence, since the global maximum is reached.

3 Grade B-A

3.1 Shooting

The most likely next move for one single bird is computed as:

$$\begin{aligned}
 \text{next move} &= \arg \max_m P(O_{T+1} = o_m | O_{1:T}) \\
 &= \sum_{i=1}^N P(O_{T+1} | X_{T+1} = x_i) P(X_{T+1} = x_i | O_{1:T}) \\
 &= \sum_{i=1}^N b_i(O_{T+1}) P(X_{T+1} = x_i | O_{1:T})
 \end{aligned} \tag{14}$$

Let's compute $P(X_{T+1} = x_i | O_{1:T})$:

$$\begin{aligned}
 P(X_{T+1} = x_i | O_{1:T}) &= \sum_{j=1}^N P(X_{T+1} = x_i | X_T = x_j) P(X_T = x_j | O_{1:T}) \\
 &= \sum_{j=1}^N a_{ji} \frac{P(X_T = x_j, O_{1:T})}{P(O_1 : T)} \\
 &= \sum_{j=1}^N a_{ji} \frac{\alpha_T(j)}{\sum_{k=1}^N \alpha_T(k)}
 \end{aligned} \tag{15}$$

$$= \sum_{j=1}^N a_{ji} \hat{\alpha}_T(j) \tag{16}$$

So the complete formula is:

$$\text{next move} = \arg \max_m \sum_{i=1}^N b_i(O_{T+1}) \sum_{j=1}^N a_{ji} \hat{\alpha}_T(j) \tag{17}$$

The idea is to use not only the model of the current bird, but also the models of all the birds belonging to the guessed species, which would have to behave like the current bird. So, I pick the most likely next move overall.

Improvements:

- Do not shoot if the guess is unknown or black stork.
- Shoot only if *confidence* > *threshold* (set to 0.75).

3.2 Guessing

As for shooting, the idea is to use all the available models of the birds whose species has been revealed. So, I pick the species of the bird whose model maximizes the likelihood of the observation sequence (i.e. evaluation problem solved with α -pass, species recognition).

Improvements:

- First round: guess randomly to get information (actually it is better to guess always one species).
- Next rounds: guess randomly when the recognition fails (unknown result) to get information.

References

- [1] Lawrence R Rabiner. “A tutorial on hidden Markov models and selected applications in speech recognition”. In: *Proceedings of the IEEE* 77.2 (1989), pp. 257–286.