Chapter 3

Hidden Markov models

This chapter first addresses the problem of incomplete information*, and provides the foundation upon which we construct decision-theoretic models that accommodate partial observability. We introduce* hidden Markov models *(HMMs) as an extension of Markov chains to situations where the state can only be perceived through indirect and noisy observations, and go over the main approaches for estimation and inference in HMMs.*

3.1 Basic notions

We start with a widely used example, corresponding to a simplified version of a scenario proposed by Rabiner (1989).

Example 3.1

An oracle has available two urns, each filled with a different number of *white* (w) and *black* (b) balls. At each time step, the oracle draws a ball from one of the urns and discloses the color of the ball without revealing which urn the ball was drawn from. The balls are put back after their color is disclosed. After drawing a ball from a urn, the oracle will draw the next ball from the same urn with 80% probability. Conversely, the oracle will draw the next ball from a different urn with only 20% probability.

Regarding the percentage of balls in each urn, 25% of the balls in the first urn (Urn 1) are white, and 75% are black. Conversely, 75% of the balls in the second urn (Urn 2) are white, against only 25% black balls.

We can model the ball drawing process of the oracle by using a Markov chain as follows. The state space corresponds to all possible pairs (urn, ball)—since this is the information required to predict the next (urn, ball) pair. And, from the percentage of balls in each urn and the urn selection process of the oracle, we get the transition diagram of Fig. 3.1.

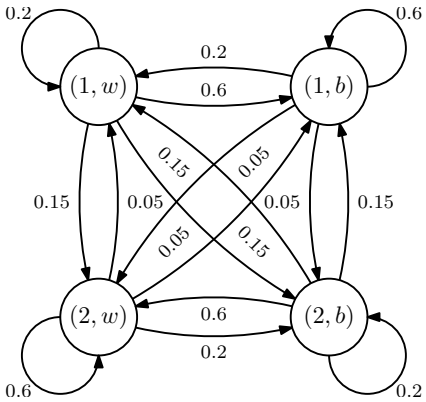

Figure 3.1 Transition diagram representing the Markov chain model for the urn example.

Alternatively, we can represent the Markov chain as a pair $(\mathcal{X}, \boldsymbol{P})$, where

- $\mathcal{X} = \{(1, w), (1, b), (2, w), (2, b)\}$;
- Numbering the states as 1 for $(1, w)$, 2 for $(1, b)$, 3 for $(2, w)$, and 4 for $(2, b)$, we get

$$\boldsymbol{P} = \begin{bmatrix} 0.2 & 0.6 & 0.15 & 0.05 \\ 0.2 & 0.6 & 0.15 & 0.05 \\ 0.05 & 0.15 & 0.6 & 0.2 \\ 0.05 & 0.15 & 0.6 & 0.2 \end{bmatrix}.$$

Suppose that the initial ball was drawn from urn 1 and was white, i.e., $\mathrm{x}_0 = 1$. At time step $t = 1$ we observe a black ball and we wish to predict the state x_2. Unlike the chains in Chapter 2, we cannot observe the full state of the chain at each time step t. For example, at time step $t = 1$ we are unable to discern between states 2 ($(1, b)$) and 4 ($(2, b)$).

Therefore, in the present domain, we face the novel challenge of how to leverage the knowledge that $\mathrm{x}_1 \in \{2, 4\}$ in the prediction of x_2. Let us denote by z_t the r.v. corresponding to the *observation* at time step t. The r.v. z_t can only take the values *white* (w) and *black* (b). Using this notation, our prediction translates into computing $\mathbb{P}[\mathrm{x}_2 = x \mid \mathrm{x}_0 = 1, \mathrm{z}_1 = b]$. Using the total probability law, we have that

$$\begin{aligned} &\mathbb{P}[\mathrm{x}_2 = x \mid \mathrm{x}_0 = 1, \mathrm{z}_1 = b] \\ &\quad = \mathbb{P}[\mathrm{x}_2 = x \mid \mathrm{x}_0 = 1, \mathrm{x}_1 = 2, \mathrm{z}_1 = b]\, \mathbb{P}[\mathrm{x}_1 = 2 \mid \mathrm{x}_0 = 1, \mathrm{z}_1 = b] \\ &\qquad + \mathbb{P}[\mathrm{x}_2 = x \mid \mathrm{x}_0 = 1, \mathrm{x}_1 = 4, \mathrm{z}_1 = b]\, \mathbb{P}[\mathrm{x}_1 = 4 \mid \mathrm{x}_0 = 1, \mathrm{z}_1 = b]. \end{aligned}$$

Using Bayes rule leads to

$$\begin{aligned}\mathbb{P}\left[\mathrm{x}_2 = x \mid \mathrm{x}_0 = 1, \mathrm{z}_1 = b\right] &= \frac{1}{\rho}\mathbb{P}\left[\mathrm{x}_2 = x \mid \mathrm{x}_1 = 2\right]\mathbb{P}\left[\mathrm{x}_1 = 2 \mid \mathrm{x}_0 = 1\right] \\ &\quad + \frac{1}{\rho}\mathbb{P}\left[\mathrm{x}_2 = x \mid \mathrm{x}_1 = 4\right]\mathbb{P}\left[\mathrm{x}_1 = 4 \mid \mathrm{x}_0 = 1\right] \\ &= \frac{1}{\rho}\Big(\boldsymbol{P}(x \mid 2)\boldsymbol{P}(2 \mid 1) + \boldsymbol{P}(x \mid 4)\boldsymbol{P}(4 \mid 1)\Big) =\end{aligned}$$

where $\rho = \mathbb{P}\left[\mathrm{z}_1 = b \mid \mathrm{x}_0 = 1\right]$. Finally, if $\mu_{2|b}(x) \stackrel{\text{def}}{=} \mathbb{P}\left[\mathrm{x}_2 = x \mid \mathrm{x}_0 = 1, \mathrm{z}_1 = b\right]$, we get our prediction

$$\boldsymbol{\mu}_{2|b} = \begin{bmatrix} 0.19 & 0.57 & 0.18 & 0.06 \end{bmatrix}.$$

To formalize the model in Example 3.1, we note that it actually comprises *two processes*: a process $\{\mathrm{x}_t, t \in \mathbb{N}\}$, corresponding to the succession of urn-color pairs and where each x_t takes values in some set $\mathcal{X}$; and a process $\{\mathrm{z}_t, t \in \mathbb{N}\}$, corresponding to what can be seen, where each z_t takes values in some set $\mathcal{Z}$. In the example, $\mathcal{Z}$ was a component of $\mathcal{X}$ but, in general, we allow $\mathcal{Z}$ and $\mathcal{X}$ to be completely different sets.

We introduce the following concept: a *transition distribution* between two sets U and V is a mapping $F : U \times V \to [0, 1]$ such that, for all $u \in U$,

$$\sum_{v \in V} F(u, v) = 1. \tag{3.1}$$

We can now in position to define a hidden Markov model, or HMM.

Hidden Markov model (HMM)

Let $\mathcal{X}$ and $\mathcal{Z}$ denote two arbitrary sets and μ_0 a probability distribution over $\mathcal{X}$. Let P be a transition distribution between $\mathcal{X}$ and itself and O a transition distribution between $\mathcal{X}$ and $\mathcal{Z}$. A *hidden Markov model* (HMM) is a Markov chain with state space $\mathcal{X}_{\text{HMM}} = \mathcal{X} \times \mathcal{Z}$, transition probabilities

$$\boldsymbol{P}_{\text{HMM}}\big((x', z') \mid (x, z)\big) = P(x, x')O(x', z'),$$

with $x, x' \in \mathcal{X}$ and $z, z' \in \mathcal{Z}$ and initial distribution

$$\mu_{\text{HMM}}(x, z) = \mu_0(x)O(x, z).$$

We focus on the case where both $\mathcal{X}$ and $\mathcal{Z}$ are discrete, but the above definition of HMM extends trivially to general spaces, by replacing the summation by an integral in (3.1).

3.1.1 Alternative interpretation

We now establish two important properties of the model just introduced that, together, provide an alternative interpretation for HMM that corresponds to the one more commonly found in the literature.

We first show that the process $\{x_t, t \in \mathbb{N}\}$ obtained by taking only the first component of the Markov chain $\mathcal{M}_{\text{HMM}} = (\mathcal{X}_{\text{HMM}}, \boldsymbol{P}_{\text{HMM}})$, is a (time homogeneous) Markov chain in its own right. In fact,

$$\begin{aligned}
&\mathbb{P}\left[\mathrm{x}_{t+1} = x \mid \mathbf{x}_{0:t} = \boldsymbol{x}_{0:t}, \mathbf{z}_{0:t} = \boldsymbol{z}_{0:t}\right] \\
&\quad = \sum_{z \in \mathcal{Z}} \mathbb{P}\left[\mathrm{x}_{t+1} = x, \mathrm{z}_{t+1} = z \mid \mathbf{x}_{0:t} = \boldsymbol{x}_{0:t}, \mathbf{z}_{0:t} = \boldsymbol{z}_{0:t}\right] \\
&\quad = \sum_{z \in \mathcal{Z}} \mathbb{P}\left[\mathrm{x}_{t+1} = x, \mathrm{z}_{t+1} = z \mid \mathrm{x}_t = x_t, \mathrm{z}_t = z_t\right] \\
&\quad = \sum_{z \in \mathcal{Z}} P(x_t, x) O(x, z) = P(x_t, x),
\end{aligned}$$

where the second equality follows from the Markov property of the chain $\mathcal{M}_{\text{HMM}}$.

The second property is that the observations z_t are conditionally independent of the state and observation process up to time step t given x_t. Such property relies on the following fact:

Proposition 3.1. *Given an arbitrary set of indices $t_1, \ldots, t_N$ such that $t_1 < \ldots < t_N$ and a family of functions $F_{t_1}, \ldots, F_{t_N}$, where each F_{t_n} is a real-valued function defined on $\mathcal{Z}$,*

$$\mathbb{E}\left[\prod_{n=1}^{N} F_{t_n}(\mathrm{z}_{t_n}) \mid \mathrm{x}_{t_1} = x_{t_1}, \ldots, \mathrm{x}_{t_n} = x_{t_n}\right] = \prod_{n=1}^{N} \mathbb{E}\left[F_{t_n}(\mathrm{z}_{t_n}) \mid \mathrm{x}_{t_n} = x_{t_n}\right].$$

Proof. See Section 3.6. □

We can use Proposition 3.1 to show that

$$\mathbb{P}\left[\mathrm{z}_t = z \mid \mathbf{x}_{0:t} = \boldsymbol{x}_{0:t}, \mathbf{z}_{0:t-1} = \boldsymbol{z}_{0:t-1}\right] = \mathbb{P}\left[\mathrm{z}_t = z \mid \mathrm{x}_t = x_t\right], \tag{3.2}$$

i.e., each r.v. z_t is independent of $\mathbf{x}_{0:t-1}$ and $\mathbf{z}_{0:t}$ given x_t (see Exercise 3.1).

The two properties above imply that we can interpret an HMM as comprising a Markov chain $\{\mathrm{x}_t, t \in \mathbb{N}\}$ associated with an *observation process* $\{\mathrm{z}_t, t \in \mathbb{N}\}$ such that, for each t, z_t is completely determined by x_t. For this reason, we refer to x_t as the *state at time step t* and z_t as the *observation at time step t*. We refer to $\mathcal{X}$ as the *state space* and to $\mathcal{Z}$ as the *observation space*.

Additionally, note that the transition distribution P defines the *transition probabilities* for the Markov chain $\{\mathrm{x}_t, t \in \mathbb{N}\}$, and we can let

$$\boldsymbol{P}(y \mid x) = P(x, y) \stackrel{\text{def}}{=} \mathbb{P}\left[\mathrm{x}_{t+1} = y \mid \mathrm{x}_t = x\right].$$

Similarly, we can assign to O the probabilistic interpretation

$$O(x, z) \stackrel{\text{def}}{=} \mathbb{P}\left[\mathrm{z}_t = z \mid \mathrm{x}_t = x\right].$$

Therefore, we refer to the values $O(x, z)$ as *observation probabilities*, and collect them in a *observation probability matrix* $\boldsymbol{O}$ where

$$[\boldsymbol{O}]_{x,z} = O(x, z).$$

As with the transition probabilities, we write $\boldsymbol{O}(z \mid x)$ to highlight the fact that the elements of $\boldsymbol{O}$ correspond to conditional probabilities. An HMM can thus be completely specified by

- Its state space, $\mathcal{X}$;
- Its observation space, $\mathcal{Z}$;
- Its transition probabilities, $\boldsymbol{P}$;
- Its observation probabilities, $\boldsymbol{O}$;
- Its initial distribution, μ_0.

For this reason, we henceforth denote an HMM as a tuple $(\mathcal{X}, \mathcal{Z}, \boldsymbol{P}, \boldsymbol{O}, \mu_0)$. As with Markov chains, when the initial distribution is immaterial for the discussion we use the most compact representation $(\mathcal{X}, \mathcal{Z}, \boldsymbol{P}, \boldsymbol{O})$.

To conclude, the designation of *hidden* Markov model arises from the fact that, as seen in Example 3.1, the state x_t is not directly accessible, and only z_t can be observed. Such situation is referred as *partial observability* and can be found in many real-world problems.

3.1.2 Examples

We now illustrate the use of HMMs in different application domains. The goal of these examples is to highlight the modeling process, while conveying a broad view of HMM applications. We start with our standard example of the household robot.

Household robot

We reprise the household robot example from Chapters 1 and 2. As before, we consider the situation where the robot is set to "monitoring mode", now repeatedly visiting "living areas" of the house (Kitchen, Pantry, Dining room, Hallway 1, Living room and Bedroom). As before, we assume that the transitions between areas always succeed but, upon reaching an area, the robot randomly selects which area to visit next.

We now consider that the robot possesses a laser sensor that is able to detect walls in each of the four directions (up, down, left and right). Each division can be uniquely identified by its "wall pattern":

- The Kitchen has walls at the top and at the bottom (*tb*);

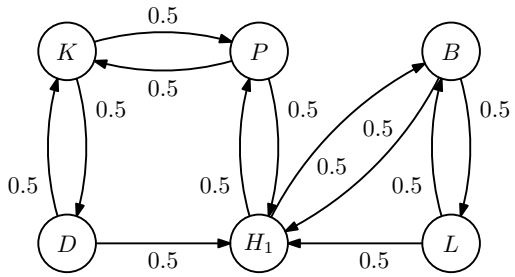

Figure 3.2 Transition diagram corresponding to the motion of the robot in the household domain.

- The Pantry has walls at the top, on the left and at the bottom (ltb);
- The Dining room has walls at the bottom and on the right (rb);
- Hallway 1 has no walls ($\emptyset$);
- The Bedroom has walls at the top and on the left (lt);
- The Living room has a wall at the top (t).

However, the laser sensor is not perfect. In the presence of a wall, it fails to detect it with a 5% probability. Conversely, with a 10% probability, it will detect a wall where there is none, except in the left passage in the Kitchen, the top passage in Hallway 1 and the bottom passages in Hallway 1 and the Living Room, where the laser detects a wall with a 20% probability. We assume that detections are independent, i.e., the fact that a wall was correctly/incorrectly independent has no implications regarding the detection of nearby walls (or absence thereof).

We can represent this problem as an HMM. The dynamics of the robot correspond to the Markov chain represented in Fig. 3.2. The state space is given by

$$\mathcal{X} = \{K, P, D, H_1, B, L\}$$

and the transition probability matrix is simply

$$\boldsymbol{P} = \begin{bmatrix} 0.0 & 0.5 & 0.5 & 0.0 & 0.0 & 0.0 \\ 0.5 & 0.0 & 0.0 & 0.5 & 0.0 & 0.0 \\ 0.5 & 0.0 & 0.0 & 0.5 & 0.0 & 0.0 \\ 0.0 & 0.5 & 0.0 & 0.0 & 0.5 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.5 & 0.0 & 0.5 \\ 0.0 & 0.0 & 0.0 & 0.5 & 0.5 & 0.0 \end{bmatrix}$$

The observation process for the HMM models the laser sensor of the robot. The possible perceptions thus correspond to all possible wall configurations, yielding

$$\mathcal{Z} = \{\emptyset, l, t, r, b, lt, lr, lb, tr, tb, rb, ltr, ltb, lrb, trb, ltrb\}\,.$$

Alternatively, we can treat the detection of an individual wall as a binary r.v., where 1 corresponds to a positive detection, and 0 to a negative detection. Then, each observation is a 4-element binary vector and

$$\mathcal{Z} = \{0,1\}^4 .$$

We adopt the latter representation, as it is computationally more convenient. Then, at each room, we can determine the probability of a given wall configuration by simply multiplying the individual wall detections. For example, the probability of observing *ltr* in the Pantry is given by

$$\begin{aligned}
&\mathbb{P}\left[\mathbf{z} = [1,1,1,0] \mid \mathrm{x} = P\right] \\
&= \mathbb{P}\left[\mathrm{z}_1 = 1 \mid \mathrm{x} = P\right] \mathbb{P}\left[\mathrm{z}_2 = 1 \mid \mathrm{x} = P\right] \mathbb{P}\left[\mathrm{z}_3 = 1 \mid \mathrm{x} = P\right] \mathbb{P}\left[\mathrm{z}_4 = 0 \mid \mathrm{x} = P\right] \\
&= 0.9 \times 0.2 \times 0.9 \times 0.1 = 0.0162.
\end{aligned}$$

Repeating this process for all state-observation pairs finally yields the observation matrix

$$\boldsymbol{O} = \begin{bmatrix}
0.0018 & 0.0004 & 0.0342 & \dots & 0.0009 & 0.0722 & 0.0180 \\
0.0001 & 0.0021 & 0.0021 & \dots & 0.0045 & 0.0045 & 0.0857 \\
0.0020 & 0.0002 & 0.0384 & \dots & 0.0005 & 0.0812 & 0.0090 \\
0.5184 & 0.0576 & 0.1296 & \dots & 0.0016 & 0.0036 & 0.0004 \\
0.0020 & 0.0385 & 0.0002 & \dots & 0.0812 & 0.0004 & 0.0090 \\
0.0324 & 0.0036 & 0.0081 & \dots & 0.0076 & 0.0171 & 0.0019
\end{bmatrix}.$$

We can now use the HMM just derived for *localization*: given the initial position of the robot and the sequence of observations $\{z_0, \dots, z_t\}$, we want to determine the position of the robot at time step t. We can use Bayes rule to get

$$\mathbb{P}\left[\mathrm{x}_t = x \mid \mathbf{z}_{0:t} = \boldsymbol{z}_{0:t}, \mathrm{x}_0 = x_0\right] = \frac{\mathbb{P}\left[\mathbf{z}_{0:t} = \boldsymbol{z}_{0:t} \mid \mathrm{x}_t = x, \mathrm{x}_0 = x_0\right] \mathbb{P}\left[\mathrm{x}_t = x \mid \mathrm{x}_0 = x_0\right]}{\mathbb{P}\left[\mathbf{z}_{0:t} = \boldsymbol{z}_{0:t} \mid \mathrm{x}_0 = x_0\right]}.$$

Determining the distribution over states from the sequence of observations is a problem known as *filtering*, and is discussed in Section 3.2.1.

Digital communication

We now model a noisy digital communication channel using an HMM. The emitter inputs a sequence of bits $\{x_0, x_1, \dots, x_T\}$ with each $x_t \in \{0,1\}$. The receiver, in turn, will observe at the output of the channel a sequence of bits $\{y_1, \dots, y_T\}$, with each $y_t \in \{0,1\}$.

We consider the case of a *memoryless channel*, in which the current output of the channel depends only on the current input thereof, i.e.,

$$\mathbb{P}\left[\mathrm{y}_t = y \mid \mathbf{x}_{0:t} = \boldsymbol{x}_{0:t}, \mathbf{y}_{0:t-1} = \boldsymbol{y}_{0:t}\right] = \mathbb{P}\left[\mathrm{y}_t = y \mid \mathrm{x}_t = x_t\right].$$

Due to errors in the channel, there is a probability p_{01} that an input bit x_t is flipped from 0 to 1 in the output, and a probability p_{10} of the input bit appearing flipped in the other direction. For simplicity, in this example we assume that $p_{01} = p_{10} = \varepsilon$,

for some $\varepsilon > 0$.[1] Suppose that the input sequence of bits results from one of a number of words $\{w_1, \ldots, w_N\}$, and the goal of the receiver is to identify (decode) the word w_n sent by the emitter from the bit stream $\{z_1, \ldots, z_T\}$ observed in the output. We can thus construct an HMM model $(\mathcal{X}, \mathcal{Z}, \boldsymbol{P}, \boldsymbol{O})$ for the channel, where $\mathcal{X} = \mathcal{Z} = \{0, 1\}$; additionally, we can build a naïve estimate for the probabilities

$$\boldsymbol{P}(y \mid x) \stackrel{\text{def}}{=} \mathbb{P}\left[\mathrm{x}_{t+1} = y \mid \mathrm{x}_t = x\right]$$

simply by counting how often, in the set of words $\{w_1, \ldots, w_N\}$, bit y follows bit x. Finally, from the properties of the communication channel, we have that

$$\boldsymbol{O}(z \mid x) = (1 - \varepsilon)\mathbb{I}\left[x = z\right] + \varepsilon(1 - \mathbb{I}\left[x = z\right]).$$

To identify the word sent by the emitter, we can adopt a Bayesian approach to get

$$\mathbb{P}\left[\mathbf{x}_{0:t} = \boldsymbol{x}_{0:t} \mid \mathbf{z}_{0:t} = \boldsymbol{z}_{0:t}\right] = \frac{\mathbb{P}\left[\mathbf{z}_{0:t} = \boldsymbol{z}_{0:t} \mid \mathbf{x}_{0:t} = \boldsymbol{x}_{0:t}\right]\mathbb{P}\left[\mathbf{x}_{0:t} = \boldsymbol{x}_{0:t}\right]}{\mathbb{P}\left[\mathbf{z}_{0:t} = \boldsymbol{z}_{0:t}\right]}.$$

Determining the probability of the complete state sequence from the sequence of observations is a problem known as *smoothing*, and is discussed in Section 3.2.2.

Finding genes in DNA sequences

A significant volume of research in bioinformatics focuses on the analysis of DNA, in part due to the fact that DNA information is cheap to obtain and abundant. DNA (which stands for Deoxyribonucleic Acid) is a molecule comprising two intertwined long strands of *nucleotides*. There are four basic types of nucleotides, each represented as one of the letters A, T, G, and C.

Certain portions of the DNA sequence encode the necessary information to synthesize proteins. In those encoding portions (the genes), each group of three consecutive nucleotides is called a *triplet* (see Fig. 3.3 for an illustration). In the protein synthesis process, these triplets are transcribed into the so-called *codons* that, in turn, encode the order by which amino-acids are arranged during protein synthesis.

However, not all parts of the DNA sequence correspond to genes. In fact, large portions of the DNA of many living beings serve no known function and appear somewhat "random", making the identification of genes a complex but very important problem. On the other hand, the sequences of triplets in the coding segments obey a distinctive distribution that sets coding segments apart from non-coding segments.

Therefore, an HMM model can be used to identify coding segments in the DNA (Krogh, Mian, and Haussler, 1994). The resulting transition diagram is depicted in Fig. 3.4, where the observations correspond to the nucleotides observed (A, T, G or C). By identifying the most likely underlying sequence of states (again, the problem of *smoothing*), we can use the HMM to identify the segments corresponding to genes.

[1] If $p_{01} = p_{10}$, the channel is called *symmetric*.

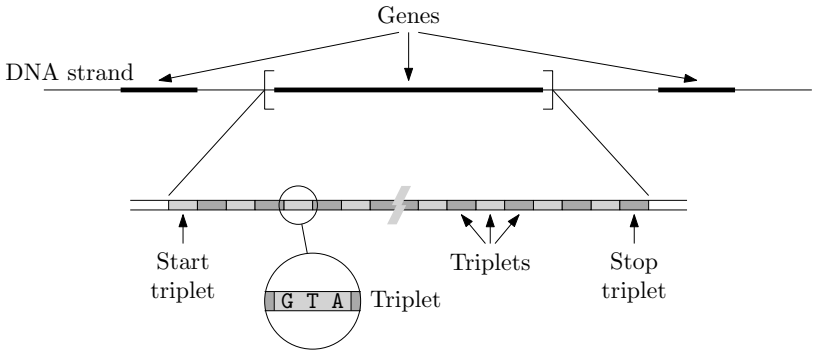

Figure 3.3 Diagram illustrating the organization of genes in DNA strands (adapted from Durbin et al., 1998).

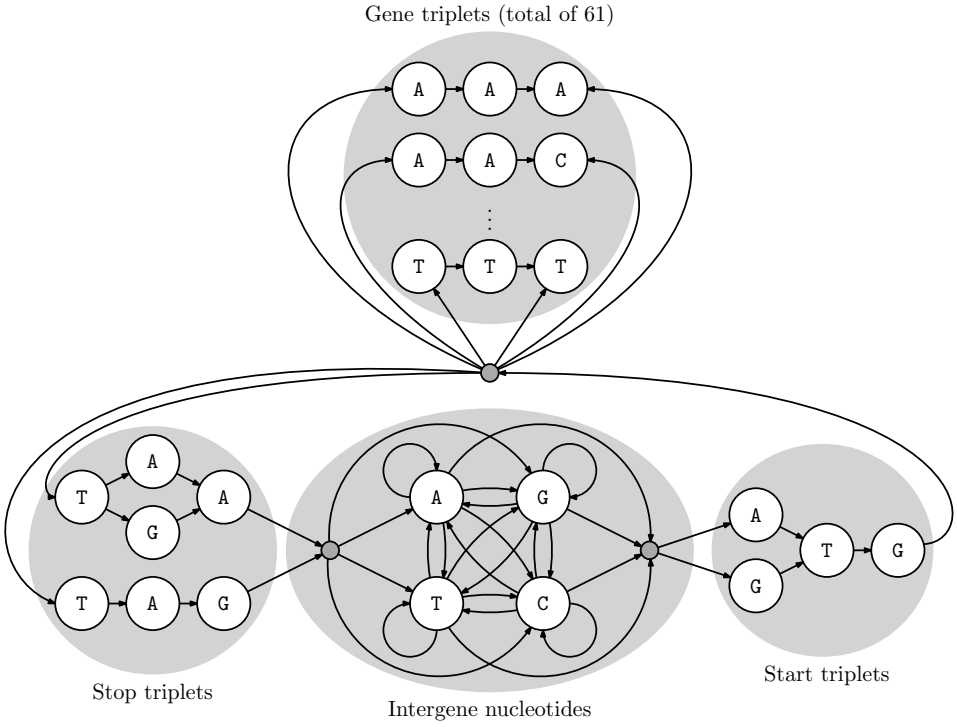

Figure 3.4 Transition diagram for the HMM used for identifying coding genes in DNA strands (adapted from Krogh, Mian, and Haussler, 1994). Transition probabilities have been omitted, to avoid cluttering the diagram. The letter in each state corresponds to the observation made in that state. The shaded states have no associated observation. The sequence of symbols generated by the corresponding HMM (roughly) corresponds to possible DNA sequences, where coding triplets are generated by the states in the uppermost part of the diagram.

3.2 Estimation

The examples discussed in the previous section showcased some of the problems that HMMs are usually cast to solve. We now describe these problems in further detail and introduce the standard algorithmic approaches to solving them. In particular, given an HMM $(\mathcal{X}, \mathcal{Z}, \boldsymbol{P}, \boldsymbol{O}, \mu_0)$ we discuss

Filtering Given the sequence of observations up to time step t, $\boldsymbol{z}_{0:t}$, we want to estimate the distribution $\mu_{t|0:t}$, defined for each $x \in \mathcal{X}$ as

$$\mu_{T|0:T}(x) \stackrel{\text{def}}{=} \mathbb{P}_{\mu_0}\left[\mathrm{x}_T = x \mid \mathbf{z}_{0:T} = \boldsymbol{z}_{0:T}\right],$$

where we use the subscript μ_0 in $\mathbb{P}_{\mu_0}$ to emphasizes the initial distribution of the chain.

Smoothing Given the sequence of observations $\boldsymbol{z}_{0:T}$, we want to estimate the sequence $\boldsymbol{x}^*_{0:T}$ such that

$$\boldsymbol{x}^*_{0:T} = \operatorname*{argmax}_{\boldsymbol{x}_{0:T}} \mathbb{P}_{\mu_0}\left[\mathbf{x}_{0:T} = \boldsymbol{x}_{0:T} \mid \mathbf{z}_{0:T} = \boldsymbol{z}_{0:T}\right].$$

We also discuss *marginal smoothing*, in which we want to determine x^*_t such that

$$x^*_t = \operatorname*{argmax}_{x \in \mathcal{X}} \mathbb{P}_{\mu_0}\left[\mathrm{x}_t = x \mid \mathbf{z}_{0:T} = \boldsymbol{z}_{0:T}\right],$$

with $0 < t \leq T$.

Prediction Given the sequence of observations $\boldsymbol{z}_{0:T}$, we want to estimate the distribution $\mu_{T+1|0:T}$ (or, more generally, $\mu_{T+t|0:T}$), defined for each $x \in \mathcal{X}$ as

$$\mu_{T+1|0:T}(x) \stackrel{\text{def}}{=} \mathbb{P}_{\mu_0}\left[\mathrm{x}_{T+1} = x \mid \mathbf{z}_{0:T} = \boldsymbol{z}_{0:T}\right].$$

To address filtering, smoothing and prediction we introduce two dynamic programming tools that lie at the core of the algorithms discussed in this section: the *forward recursion* and the *backward recursion*. However,

3.2.1 Filtering

The filtering problem consists in taking the history of observations up to time step T, $\boldsymbol{z}_{0:T}$ and estimating the state x_T, i.e., computing

$$\mu_{T|0:T}(x) \stackrel{\text{def}}{=} \mathbb{P}_{\mu_0}\left[\mathrm{x}_T = x \mid \mathbf{z}_{0:T} = \boldsymbol{z}_{0:T}\right]. \tag{3.3}$$

To that purpose, we introduce the *forward mapping* $\alpha_t : \mathcal{X} \to \mathbb{R}$, defined for each $x \in \mathcal{X}$ and each $t \in \mathbb{N}$ as

$$\alpha_t(x) \stackrel{\text{def}}{=} \mathbb{P}_{\mu_0}\left[\mathrm{x}_t = x, \mathbf{z}_{0:t} = \boldsymbol{z}_{0:t}\right]. \tag{3.4}$$

The usefulness of the forward mapping α_t in computing $\mu_{T|0:T}$ stems from the fact that

$$\mathbb{P}_{\mu_0}\left[\mathbf{z}_{0:T} = \boldsymbol{z}_{0:T}\right] = \sum_{x \in \mathcal{X}} \mathbb{P}_{\mu_0}\left[\mathrm{x}_T = x, \mathbf{z}_{0:T} = \boldsymbol{z}_{0:T}\right] = \sum_{x \in \mathcal{X}} \alpha_T(x)$$

and hence

$$\mu_{T|0:T}(x) = \frac{\alpha_T(x)}{\sum_{y\in\mathcal{X}} \alpha_T(y)}.$$

Additionally, the forward mapping is computationally interesting, due to its intrinsically recursive nature. In fact, using standard manipulations, we get

$$\begin{aligned}
&\alpha_T(x) \\
&= \mathbb{P}_{\mu_0}\left[\mathrm{x}_T = x, \mathbf{z}_{0:T} = \boldsymbol{z}_{0:T}\right] \\
&= \mathbb{P}_{\mu_0}\left[\mathrm{z}_T = z_T \mid \mathrm{x}_T = x, \mathbf{z}_{0:T-1} = \boldsymbol{z}_{0:T-1}\right] \mathbb{P}_{\mu_0}\left[\mathrm{x}_T = x, \mathbf{z}_{0:T-1} = \boldsymbol{z}_{0:T-1}\right] \\
&= \boldsymbol{O}(z_T \mid x)\ \mathbb{P}_{\mu_0}\left[\mathrm{x}_T = x, \mathbf{z}_{0:T-1} = \boldsymbol{z}_{0:T-1}\right],
\end{aligned}$$

where the last equality follows from (3.2). Expanding the second term on the right-hand side using the total probability law, we get

$$\begin{aligned}
\alpha_T(x) = \boldsymbol{O}(z_T \mid x) \sum_{y\in\mathcal{X}} &\mathbb{P}_{\mu_0}\left[\mathrm{x}_T = x \mid \mathrm{x}_{T-1} = y, \mathbf{z}_{0:T-1} = \boldsymbol{z}_{0:T-1}\right] \\
&\cdot \mathbb{P}_{\mu_0}\left[\mathrm{x}_{T-1} = y, \mathbf{z}_{0:T-1} = \boldsymbol{z}_{0:T-1}\right].
\end{aligned}$$

Finally, using the Markov property of the process $\{\mathrm{x}_t, t \in \mathbb{N}\}$, we get

$$\alpha_T(x) = \boldsymbol{O}(z_T \mid x) \sum_{y\in\mathcal{X}} \boldsymbol{P}(x \mid y)\alpha_{T-1}(y) \tag{3.5}$$

or, in vector notation,

$$\boldsymbol{\alpha}_T = \mathrm{diag}(\boldsymbol{O}_{:,z_T})\boldsymbol{P}^\top \boldsymbol{\alpha}_{T-1}$$

where $\mathrm{diag}(\boldsymbol{O}_{:,z_T})$ denotes the diagonal matrix obtained from the z_T column of $\boldsymbol{O}$. The recursion in (3.5) is known as *forward recursion*, since we compute the mappings $\alpha_0, \alpha_1, \ldots, \alpha_T$ successively by "going forward" in time. The forward computation of $\mu_{T|0:T}$ is summarized in Algorithm 3.1.

Algorithm 3.1 Forward computation of $\mu_{T|0:T}$. We write $\mathbf{1}$ to denote the all-ones column vector.

Require: Observation sequence $\boldsymbol{z}_{0:T}$
1: Initialize $\boldsymbol{\alpha}_0 \leftarrow \mathrm{diag}(\boldsymbol{O}_{:,z_0})\boldsymbol{\mu}_0^\top$
2: **for** $t = 1, \ldots, T$ **do**
3: $\quad \boldsymbol{\alpha}_t \leftarrow \mathrm{diag}(\boldsymbol{O}_{:,z_t})\boldsymbol{P}^\top \boldsymbol{\alpha}_{t-1}$
4: **end for**
5: **return** $\boldsymbol{\mu}_{T|0:T} = \boldsymbol{\alpha}_T^\top / (\boldsymbol{\alpha}_T^\top \mathbf{1})$

3.2.2 Smoothing

We start by considering the simpler problem of marginal smoothing.

Marginal smoothing

We want to estimate

$$\mu_{t|0:T}(x) \stackrel{\text{def}}{=} \mathbb{P}_{\mu_0}\left[\mathrm{x}_t = x \mid \mathbf{z}_{0:T} = \boldsymbol{z}_{0:T}\right], \tag{3.6}$$

where $0 < t \leq T$. The particular case where $t = T$ reduces to the filtering problem that, as discussed, can be addressed using Algorithm 3.1. For the case were $t < T$, we introduce the *backward mapping* $\beta_t : \mathcal{X} \to \mathbb{R}$, defined for all $x \in \mathcal{X}$ as

$$\beta_t(x) \stackrel{\text{def}}{=} \mathbb{P}_{\mu_0}\left[\mathbf{z}_{t+1:T} = \boldsymbol{z}_{t+1:T} \mid \mathrm{x}_t = x\right], \tag{3.7}$$

with the convention that $\beta_T(x) \equiv 1$ for all $x \in \mathcal{X}$. We have that, for any $0 < t < T$,

$$\begin{aligned}
&\mathbb{P}_{\mu_0}\left[\mathbf{z}_{0:T} = \boldsymbol{z}_{0:T}\right] \\
&= \sum_{x \in \mathcal{X}} \mathbb{P}_{\mu_0}\left[\mathbf{z}_{0:T} = \boldsymbol{z}_{0:T}, \mathrm{x}_t = x\right] \\
&= \sum_{x \in \mathcal{X}} \mathbb{P}_{\mu_0}\left[\mathbf{z}_{t+1:T} = \boldsymbol{z}_{t+1:T} \mid \mathrm{x}_t = x, \mathbf{z}_{0:t} = \boldsymbol{z}_{0:t}\right] \mathbb{P}_{\mu_0}\left[\mathrm{x}_t = x, \mathbf{z}_{0:t} = \boldsymbol{z}_{0:t}\right] \\
&= \sum_{x \in \mathcal{X}} \mathbb{P}_{\mu_0}\left[\mathbf{z}_{t+1:T} = \boldsymbol{z}_{t+1:T} \mid \mathrm{x}_t = x\right] \mathbb{P}_{\mu_0}\left[\mathrm{x}_t = x, \mathbf{z}_{0:t} = \boldsymbol{z}_{0:t}\right] \\
&= \sum_{x \in \mathcal{X}} \beta_t(x)\alpha_t(x),
\end{aligned}$$

which implies that

$$\begin{aligned}
\mu_{t|0:T}(x) &= \frac{\mathbb{P}_{\mu_0}\left[\mathbf{z}_{0:T} = \boldsymbol{z}_{0:T}, \mathrm{x}_t = x\right]}{\mathbb{P}_{\mu_0}\left[\mathbf{z}_{0:T} = \boldsymbol{z}_{0:T}\right]} \\
&= \frac{\mathbb{P}_{\mu_0}\left[\mathbf{z}_{t+1:T} = \boldsymbol{z}_{t+1:T} \mid \mathrm{x}_t = x\right] \mathbb{P}_{\mu_0}\left[\mathrm{x}_t = x, \mathbf{z}_{0:t} = \boldsymbol{z}_{0:t}\right]}{\mathbb{P}_{\mu_0}\left[\mathbf{z}_{0:T} = \boldsymbol{z}_{0:T}\right]} \\
&= \frac{\beta_t(x)\alpha_t(x)}{\sum_{y \in \mathcal{X}} \beta_t(y)\alpha_t(y)}.
\end{aligned} \tag{3.8}$$

It follows that we can compute $\mu_{t|0:T}(x)$ directly from the forward and backward mappings α_t and β_t. Moreover, as the forward mapping, the backward mapping is also amenable to recursive computation.

Lemma 3.2. *Given an HMM $(\mathcal{X}, \mathcal{Z}, \boldsymbol{P}, \boldsymbol{O}, \mu_0)$ and a sequence of observations $\boldsymbol{z}_{0:T}$, the backward mapping β_t verifies, for all $0 < t < T$,*

$$\beta_t(x) = \sum_{y \in \mathcal{X}} \mathbb{P}\left[\mathrm{z}_{t+1} = z_{t+1} \mid \mathrm{x}_{t+1} = y\right] \beta_{t+1}(y) \boldsymbol{P}(y \mid x) \tag{3.9}$$

Proof. See Section 3.6. □

Writing the backward mapping β_t as a vector $\boldsymbol{\beta}_t$ with xth component given by $\beta_t(x)$, we can rewrite (3.9) as

$$\boldsymbol{\beta}_t = \boldsymbol{P}\ \mathrm{diag}(O_{:,z_{t+1}})\boldsymbol{\beta}_{t+1}.$$

The recursion in (3.9) is known as *backward recursion*, since we compute the mappings $\beta_T, \beta_{T-1}, \ldots, \beta_t$ successively by "going backward" in time. Finally, we summarize in Algorithm 3.2 the *forward-backward computation* necessary to determine $\mu_{t|0:T}$.

Algorithm 3.2 Forward-backward algorithm to compute $\mu_{t|0:T}$. We write $a \otimes b$ to denote the component-wise product of vectors a and b.

Require: Observation sequence $\boldsymbol{z}_{0:T}$
Initialize $\boldsymbol{\alpha}_0 \leftarrow \operatorname{diag}(\boldsymbol{O}_{:,z_0})\boldsymbol{\mu}_0^\top$
Initialize $\boldsymbol{\beta}_T \leftarrow \mathbf{1}$
for $\tau = 1, \ldots, t$ **do**
    $\boldsymbol{\alpha}_\tau \leftarrow \operatorname{diag}(\boldsymbol{O}_{:,z_\tau})\boldsymbol{P}^\top \boldsymbol{\alpha}_{\tau-1}$
end for
for $\tau = T-1, \ldots, t$ **do**
    $\boldsymbol{\beta}_\tau \leftarrow \boldsymbol{P} \operatorname{diag}(\boldsymbol{O}_{:,z_{\tau+1}})\boldsymbol{\beta}_{\tau+1}$
end for
return $\mu_{t|0:T}^\top = \boldsymbol{\alpha}_t \otimes \boldsymbol{\beta}_t / (\boldsymbol{\alpha}_t^\top \boldsymbol{\beta}_t)$

It is interesting to note at this point the relation between the forward distribution $\boldsymbol{\alpha}_t$ and the backward mapping $\boldsymbol{\beta}_t$. Given a sequence of observations $\boldsymbol{z}_{0:T}$, we get

$$\sum_{x \in \mathcal{X}} \alpha_T(x) = \mathbb{P}_{\mu_0}\left[\mathbf{z}_{0:T} = \boldsymbol{z}_{0:T}\right] = \sum_{x \in \mathcal{X}} \mu_0(x) \boldsymbol{O}(z_0 \mid x) \beta_0(x).$$

Noting that $\boldsymbol{\alpha}_0^\top = \boldsymbol{\mu}_0 \operatorname{diag}(\boldsymbol{O}_{:,z_0})$ and $\boldsymbol{\beta}_T = \mathbf{1}$, we can write the relation above as

$$\boldsymbol{\alpha}_T^\top \boldsymbol{\beta}_T = \boldsymbol{\alpha}_0^\top \boldsymbol{\beta}_0.$$

Joint smoothing

Let us now address the problem of *joint smoothing* where, given the sequence of observations $\boldsymbol{z}_{0:T}$, we wish to estimate the most likely sequence of states, i.e.,

$$\boldsymbol{x}_{0:T}^* = \operatorname*{argmax}_{\boldsymbol{x}_{0:T}} \mathbb{P}_{\mu_0}\left[\mathbf{x}_{0:T} = \boldsymbol{x}_{0:T} \mid \mathbf{z}_{0:T} = \boldsymbol{z}_{0:T}\right]. \tag{3.10}$$

A naïve approach to computing (3.10) could be to compute $\mu_{t|0:T}$ for $t = 0, \ldots, T$ and let

$$x_t^* = \operatorname*{argmax}_{x \in \mathcal{X}} \mu_{t|0:T}(x), \quad t = 1, \ldots, T.$$

Unfortunately, the sequence thus obtained ignores the dependence between successive states induced by the HMM dynamics. In the worst case, such simplistic approach may even result in a sequence $\boldsymbol{x}_{0:T}^*$ such that

$$\mathbb{P}_{\mu_0}\left[\mathbf{x}_{0:T} = \boldsymbol{x}_{0:T}^* \mid \mathbf{z}_{0:T} = \boldsymbol{z}_{0:T}\right] = 0.$$

The standard approach to determining the sequence of states in (3.10) is known as the *Viterbi algorithm*. It can be seen as a modified version of the forward-backward algorithm that, in the forward pass, successively identifies the most likely

state at each step $t = 0, \dots, T$, as a function of subsequent states. Then, in the backward pass, the algorithm traces back the most likely sequence of states.

To derive the Viterbi algorithm, we start by noting that

$$\mathbb{P}_{\mu_0}\left[\mathbf{x}_{0:T} = \boldsymbol{x}_{0:T} \mid \mathbf{z}_{0:T} = \boldsymbol{z}_{0:T}\right] = \frac{\mathbb{P}_{\mu_0}\left[\mathbf{x}_{0:T} = \boldsymbol{x}_{0:T}, \mathbf{z}_{0:T} = \boldsymbol{z}_{0:T}\right]}{\mathbb{P}_{\mu_0}\left[\mathbf{z}_{0:T} = \boldsymbol{z}_{0:T}\right]}$$

and, since the denominator does not depend on $\boldsymbol{x}_{0:t}$,

$$\operatorname*{argmax}_{\boldsymbol{x}_{0:T}} \mathbb{P}_{\mu_0}\left[\mathbf{x}_{0:T} = \boldsymbol{x}_{0:T} \mid \mathbf{z}_{0:T} = \boldsymbol{z}_{0:T}\right] = \operatorname*{argmax}_{\boldsymbol{x}_{0:T}} \mathbb{P}_{\mu_0}\left[\mathbf{x}_{0:T} = \boldsymbol{x}_{0:T}, \mathbf{z}_{0:T} = \boldsymbol{z}_{0:T}\right].$$

We introduce the *maximizing forward mapping* $m_t : \mathcal{X} \to \mathbb{R}$, defined as

$$m_t(x) \stackrel{\text{def}}{=} \max_{\boldsymbol{x}_{0:t-1}} \mathbb{P}_{\mu_0}\left[\mathrm{x}_t = x, \mathbf{x}_{0:t-1} = \boldsymbol{x}_{0:t-1}, \mathbf{z}_{0:t} = \boldsymbol{z}_{0:t}\right].$$

The value $m_t(x)$ corresponds to the probability of the most likely sequence of length t that ends up in state x and can also be computed recursively.

Lemma 3.3. *Given an HMM $(\mathcal{X}, \mathcal{Z}, \boldsymbol{P}, \boldsymbol{O}, \mu_0)$, the maximizing forward mapping m_t verifies, for all $t > 0$,*

$$m_t(x) = \boldsymbol{O}(z_t \mid x) \max_y \left\{\boldsymbol{P}(x \mid y) m_{t-1}(y)\right\}. \tag{3.11}$$

Proof. See Section 3.6. □

We can write (3.11) in vector form as

$$\boldsymbol{m}_t = \operatorname{diag}(\boldsymbol{O}_{:,z_t}) \max\left\{\boldsymbol{P}^\top \operatorname{diag}(\boldsymbol{m}_{t-1})\right\}, \tag{3.12}$$

where the max is taken row-wise. Note that the recursion in (3.12) is similar to (3.5), with the summation in the latter replaced by a maximization in the former.

The value $m_t(x)$ corresponds to the probability of the most likely t-step sequence that ends up in x (given the observations). The recursion in (3.12) states that such probability can be computed from the probability of the most likely $t-1$-step sequence ending up in the state x_{t-1} that maximizes $\boldsymbol{P}(x \mid x_{t-1}) m_{t-1}(x_{t-1})$. Such maximizing state must surely be part of the most likely sequence.

We can finally compute the desired state sequence by (i) determining $m_t, t = 0, \dots, T$; and (ii) for each t, tracking the state y that maximizes $\boldsymbol{P}(x \mid y) m_{t-1}(y)$. Defining the index

$$i_t(x) = \operatorname*{argmax}_{y \in \mathcal{X}} \left\{\boldsymbol{P}(x \mid y) m_{t-1}(y)\right\},$$

we can finally summarize the Viterbi algorithm in Algorithm 3.3.

Algorithm 3.3 The Viterbi algorithm. Both the max and argmax operators are taken row-wise.

Require: Observation sequence $\boldsymbol{z}_{0:T}$
Initialize $\boldsymbol{m}_0 \leftarrow \operatorname{diag}(\boldsymbol{O}_{:,z_0})\boldsymbol{\mu}_0^\top$
for $t = 1, \dots, T$ **do**
    $\boldsymbol{m}_t = \operatorname{diag}(\boldsymbol{O}_{:,z_t}) \max\left\{\boldsymbol{P}^\top \operatorname{diag}(\boldsymbol{m}_{t-1})\right\}$,
    $\boldsymbol{i}_t = \operatorname{argmax}\left\{\boldsymbol{P}^\top \operatorname{diag}(\boldsymbol{m}_{t-1})\right\}$
end for
$x_T^* = \operatorname{argmax}_{x \in \mathcal{X}} m_T(x)$
for $t = T-1, \dots, 0$ **do**
    $x_t^* = i_{t+1}(x_{t+1}^*)$
end for
return $\boldsymbol{x}_{0:T}^*$

3.2.3 Prediction

Finally, we consider the problem of prediction, i.e., given the sequence of observations $\boldsymbol{z}_{0:T}$, we want to estimate

$$\mu_{T+1|0:T}(x) \stackrel{\text{def}}{=} \mathbb{P}_{\mu_0}\left[\mathrm{x}_{T+1} = x \mid \mathbf{z}_{0:T} = \boldsymbol{z}_{0:T}\right].$$

Unsurprisingly, in light of the Markov property the distribution $\mu_{T+1|0:T}$ can be directly computed from $\mu_{T|0:T}$ (see Exercise 3.3) yielding, in vector form,

$$\boldsymbol{\mu}_{T+1|0:T} = \boldsymbol{\mu}_{T|0:T}\boldsymbol{P}.$$

3.2.4 Example

We conclude this section with an illustration of the application of the forward-backward and Viterbi algorithms in the urn scenario from Example 3.1. We can represent the ball-drawing process as an HMM $(\mathcal{X}, \mathcal{Z}, \boldsymbol{P}, \boldsymbol{O}, \mu_0)$, where

- $\mathcal{X} = \{(1,w), (1,b), (2,w), (2,b)\}$;
- $\mathcal{Z} = \{w, b\}$;
- The transition probabilities are given by

$$\boldsymbol{P} = \begin{bmatrix} 0.2 & 0.6 & 0.15 & 0.05 \\ 0.2 & 0.6 & 0.15 & 0.05 \\ 0.05 & 0.15 & 0.6 & 0.2 \\ 0.05 & 0.15 & 0.6 & 0.2 \end{bmatrix};$$

- The observation probabilities are given by

$$\boldsymbol{O} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 0 \\ 0 & 1 \end{bmatrix};$$

- The initial distribution is given by

$$\mu_0 = \begin{bmatrix} 0.125 & 0.375 & 0.375 & 0.125 \end{bmatrix}.$$

Given the observation sequence $\boldsymbol{z}_{0:2} = [w, w, b]$, we start by computing the most likely state at time $t = 2$ using the forward computation in Algorithm 3.1:

$$\begin{aligned}
\boldsymbol{\alpha}_0 &= \begin{bmatrix} 0.125 & 0.000 & 0.375 & 0.000 \end{bmatrix}^\top; \\
\boldsymbol{\alpha}_1 &= \begin{bmatrix} 0.044 & 0.000 & 0.244 & 0.000 \end{bmatrix}^\top; \\
\boldsymbol{\alpha}_2 &= \begin{bmatrix} 0.000 & 0.063 & 0.000 & 0.051 \end{bmatrix}^\top.
\end{aligned}$$

From $\boldsymbol{\alpha}_2$ we get

$$\boldsymbol{\mu}_{T|0:T} = \boldsymbol{\alpha}_T^\top / (\boldsymbol{\alpha}_T^\top \mathbf{1}) = \begin{bmatrix} 0.000 & 0.552 & 0 & 0.448 \end{bmatrix},$$

and the most likely state is $\mathrm{x}_2 = (1, b)$. Let us now compute the most likely state-sequence, using the Viterbi algorithm (Algorithm 3.3). In the forward pass, we get

$$\begin{aligned}
\boldsymbol{m}_0 &= \begin{bmatrix} 0.125 & 0.000 & 0.375 & 0.000 \end{bmatrix}^\top; \\
\boldsymbol{m}_1 &= \begin{bmatrix} 0.025 & 0.000 & 0.225 & 0.000 \end{bmatrix}^\top; \\
\boldsymbol{m}_2 &= \begin{bmatrix} 0.000 & 0.034 & 0.000 & 0.045 \end{bmatrix}^\top
\end{aligned}$$

and the corresponding indices

$$\begin{aligned}
\boldsymbol{i}_1 &= \begin{bmatrix} (1,w) & (1,w) & (2,w) & (2,w) \end{bmatrix}^\top; \\
\boldsymbol{i}_2 &= \begin{bmatrix} (2,w) & (2,w) & (2,w) & (2,w) \end{bmatrix}^\top.
\end{aligned}$$

In the backward pass, we finally get

$$\begin{aligned}
x_2^* &= (2, b); \\
x_1^* &= (2, w); \\
x_0^* &= (2, w).
\end{aligned}$$

3.3 Inference in HMMs $(\star)$

This section addresses a problem that, in a sense, is complementary to those discussed in Section 3.2. In particular, in Section 3.2 we saw how to compute estimates about the state process, $\{\mathrm{x}_t, t \in \mathbb{N}\}$, from the observation process, $\{\mathrm{z}_t, t \in \mathbb{N}\}$. In computing such estimates, we assumed that the HMM model was known/given.

We now devote to the problem of *inferring* the underlying HMM model from the observed sequences. In particular, we discuss how $\boldsymbol{P}$ and $\boldsymbol{O}$ can be estimated given one or more sequences of observations $\boldsymbol{z}_{0:T}$.[2] For economy of notation, we

[2] We assume that the initial distribution, μ_0, is known. However, most methods discussed in this section extend trivially to the case where μ_0 is unknown.

henceforth write $\boldsymbol{\theta}$ to denote the vector comprising the two unknown parameters $\boldsymbol{P}$ and $\boldsymbol{O}$, i.e.,

$$\boldsymbol{\theta} = [\boldsymbol{P}_{:}^{\top}, \boldsymbol{O}_{:}^{\top}]^{\top}, \tag{3.13}$$

where $\boldsymbol{P}_{:}$ and $\boldsymbol{O}_{:}$ denote the stacked forms of $\boldsymbol{P}$ and $\boldsymbol{O}$, respectively. We refer to $\boldsymbol{\theta}$ generically as the *HMM parameter vector*.

We discuss two variations of the problem:

Maximum likelihood approach Given a sequence of observations $\boldsymbol{z}_{0:T}$, we want to determine the parameter vector $\boldsymbol{\theta}$ that maximizes the *likelihood* of the observations, i.e.,

$$p(\boldsymbol{z}_{0:T}; \boldsymbol{\theta}) = \mathbb{P}\left[\mathbf{z}_{0:T} = \boldsymbol{z}_{0:T}; \boldsymbol{\theta}\right]. \tag{3.14}$$

Bayesian approach We treat the HMM parameter vector $\boldsymbol{\theta}$ as a r.v., θ. Then, given a sequence of observations $\boldsymbol{z}_{0:T}$ and a prior distribution μ_{prior} over the set of possible transition and observation matrices, we want to compute the *posterior distribution for* θ given the observations, i.e.,

$$\begin{aligned} \mu_{\text{post}}(\boldsymbol{\theta} \mid \boldsymbol{z}_{0:T}) &\stackrel{\text{def}}{=} \mathbb{P}\left[\theta = \boldsymbol{\theta} \mid \mathbf{z}_{0:t} = \boldsymbol{z}_{0:T}\right] \\ &= \frac{\mathbb{P}\left[\mathbf{z}_{0:t} = \boldsymbol{z}_{0:T} \mid \theta = \boldsymbol{\theta}\right] \mu_{\text{prior}}(\boldsymbol{\theta})}{\mathbb{P}\left[\mathbf{z}_{0:t} = \boldsymbol{z}_{0:T}\right]}. \end{aligned} \tag{3.15}$$

The latter problem is significantly more complex than the former and will be discussed only superficially. We refer to the literature on HMMs for a more detailed treatment of Bayesian inference for HMMs (see, for example, the book of Cappé, Moulines, and Rydén (2005)).

3.3.1 Maximum likelihood approach

We start with the problem of estimating the parameters of an HMM, namely $\boldsymbol{P}$ and $\boldsymbol{O}$, from data. We consider the simpler situation where state and observation data is available before moving to the significantly harder problem of estimating the HMM parameters from observation data only.

Inferring P and O from complete data

Let $\mathcal{M} = (\mathcal{X}, \mathcal{Z}, \boldsymbol{P}, \boldsymbol{O})$ denote an HMM where $\boldsymbol{P}$ and $\boldsymbol{O}$ are unknown. We want to infer the two model parameters from a state sequence, $\boldsymbol{x}_{0_T}$, and the corresponding observation sequence $\boldsymbol{z}_{0:T}$. In light of the Markov property and (3.2), the likelihood of the sequences $\boldsymbol{x}_{0_T}$ and $\boldsymbol{z}_{0:T}$ is given by

$$\begin{aligned} p(\boldsymbol{x}_{0:T}, \boldsymbol{z}_{0:T}; \boldsymbol{\theta}) &\stackrel{\text{def}}{=} \mathbb{P}\left[\mathbf{x}_{0:t} = \boldsymbol{x}_{0:T}, \mathbf{z}_{0:T} = \boldsymbol{z}_{0:T}; \boldsymbol{\theta}\right] \\ &= \mu_0(x_0)\boldsymbol{O}(z_0 \mid x_0) \prod_{t=1}^{T} \boldsymbol{P}(x_t \mid x_{t-1})\boldsymbol{O}(z_t \mid x_t) \end{aligned}$$

where, as before, $\boldsymbol{\theta}$ denotes the HMM parameter vector. We want to determine the vector $\boldsymbol{\theta}^*$ such that

$$\boldsymbol{\theta}^* = \operatorname*{argmax}_{\boldsymbol{\theta}} p(\boldsymbol{x}_{0:T}, \boldsymbol{z}_{0:T}; \boldsymbol{\theta})$$

or, equivalently,

$$\boldsymbol{\theta}^* = \operatorname*{argmax}_{\boldsymbol{\theta}} \log p(\boldsymbol{x}_{0:T}, \boldsymbol{z}_{0:T}; \boldsymbol{\theta}).$$

Looking at the term $\log p(\boldsymbol{x}_{0:T}, \boldsymbol{z}_{0:T}; \boldsymbol{\theta})$, we can replace the definition of p to get

$$\log p(\boldsymbol{x}_{0:T}, \boldsymbol{z}_{0:T}; \boldsymbol{\theta}) = \log \mu_0(x_0) + \sum_{t=0}^{T} \log \boldsymbol{O}(z_t \mid x_t) + \sum_{t=1}^{T} \log \boldsymbol{P}(x_t \mid x_{t-1}). \quad (3.16)$$

Only the first summation depends on $\boldsymbol{O}$ and only the second summation depends on $\boldsymbol{P}$. Therefore, $\boldsymbol{P}$ and $\boldsymbol{O}$ can easily be found using the method of Lagrange multipliers (see Exercise 3.4), yielding the solution

$$\boldsymbol{P}(y \mid x) = \frac{N_{x,y}}{N_x}, \qquad \text{and} \qquad \boldsymbol{O}(z \mid x) = \frac{N_{x,z}}{N_x},$$

where:

- $N_{x,y}$ denotes the number of times that a transition was observed from x to y in $\boldsymbol{x}_{0:T}$;
- N_x denotes the number of times that state x was visited in $\boldsymbol{x}_{0:T}$;
- $N_{x,z}$ denotes the number of times that z was observed in state x in $\boldsymbol{x}_{0:T}$ and $\boldsymbol{z}_{0:T}$.

Inferring $\boldsymbol{P}$ and $\boldsymbol{O}$ from incomplete data

We now address the more challenging problem of inferring $\boldsymbol{P}$ and $\boldsymbol{O}$ from the observation sequence only. To that purpose, we use the *expectation-maximization* (EM) algorithm, an iterative algorithm commonly used to compute maximum likelihood estimates from incomplete data. We provide a general overview of EM before specializing it to the problem of computing the parameters of an HMM.

$\diamond$

Let z denote a r.v. taking values in some finite set $\mathcal{Z}$ and x a second r.v. taking values in the finite set $\mathcal{X}$. We assume that z can be observed, while x is hidden. The two r.v.s are governed by the joint distribution

$$p(x, z; \theta) = \mathbb{P}\left[\mathrm{x} = x, \mathrm{z} = z; \theta\right],$$

where θ is an unknown parameter. We want to determine θ^* that maximizes the likelihood of the observed data, i.e.,

$$\theta^* = \operatorname*{argmax}_{\theta} p_{\mathrm{z}}(z; \theta),$$

or, equivalently,

$$\theta^* = \operatorname*{argmax}_{\theta} \log p_{\mathrm{z}}(z; \theta), \quad (3.17)$$

with

$$p_{\mathrm{z}}(z;\theta) = \sum_{x \in \mathcal{X}} p(x, z; \theta).$$

Unfortunately, the maximization in (3.17) is often intractable. In those situations, we select a "surrogate" function q in the hope that, by maximizing q, we are also maximizing $\log p_{\mathrm{z}}$. One natural candidate is the logarithm of the joint distribution, $\log p(x, z; \theta)$; we marginalize out the r.v. x to get the surrogate

$$q(\theta) = \mathbb{E}\left[\log p(x, z; \theta)\right] = \sum_{x \in \mathcal{X}} \log p(x, z; \theta) p(x), \tag{3.18}$$

for some distribution p. Ideally, p should be selected so that

$$q(\theta) - q(\theta') \leq \log p_{\mathrm{z}}(z;\theta) - \log p_{\mathrm{z}}(z;\theta').$$

The following result provides the necessary guarantee.

Proposition 3.4. *Selecting the distribution p in* (3.18) *to be $p(x) = p(x \mid z; \theta')$ yields*

$$\mathbb{E}\left[\log p(x, z; \theta) \mid z; \theta'\right] - \mathbb{E}\left[\log p(x, z; \theta') \mid z; \theta'\right] \leq \log p_{\mathrm{z}}(z;\theta) - \log p_{\mathrm{z}}(z;\theta').$$

Proof. See Section 3.6. □

The result above provides the necessary tool to build a sequence $\{\theta_0, \theta_1, \ldots\}$ that, as desired, successively improves $\log p_{\mathrm{z}}$. Let

$$q(\theta \mid \theta') = \mathbb{E}\left[\log p(x, z; \theta) \mid z; \theta'\right] = \sum_{x \in \mathcal{X}} \log p(x, z; \theta) p(x \mid z; \theta').$$

The sequence $\{\theta_t, t \in \mathbb{N}\}$ defined recursively as $\theta_{t+1} = \operatorname{argmax}_\theta q(\theta \mid \theta_t)$ verifies, by construction, $q(\theta_{t+1} \mid \theta_t) \geq q(\theta_t \mid \theta_t)$, thus implying that

$$\log p_{\mathrm{z}}(z;\theta_{t+1}) \geq \log p_{\mathrm{z}}(z;\theta_t).$$

The resulting algorithm is called *Expectation-Maximization* (EM) and is summarized in Algorithm 3.4.

◇

We now detail the EM algorithm for HMMs, also known as the *Baum-Welch algorithm*. We want to determine the parameter vector $\boldsymbol{\theta}$—defined in (3.13)—to maximize the likelihood

$$p(\boldsymbol{z}_{0:T}; \boldsymbol{\theta}) \stackrel{\text{def}}{=} \mathbb{P}\left[\mathbf{z}_{0:T} = \boldsymbol{z}_{0:T}; \boldsymbol{P}, \boldsymbol{O}\right],$$

where $\boldsymbol{z}_{0:T}$ is a sequence of observations and $\boldsymbol{P}$ and $\boldsymbol{O}$ are the unknown HMM parameters. The corresponding sequence of states, $\boldsymbol{x}_{0:T}$, cannot be observed. To

Algorithm 3.4 The Expectation-Maximization algorithm.

Require: $\log p(x, z; \theta)$ and $p(x \mid z; \theta)$ for all $x \in \mathcal{X}, z \in \mathcal{Z}, \theta$
Require: Stopping condition $\varepsilon > 0$
Initialize θ_0 arbitrarily
Initialize $t \leftarrow 0$
repeat
    E-step: Compute $q(\theta \mid \theta_t)$ for all θ
    M-step: $\theta_{t+1} \leftarrow \operatorname{argmax}_\theta q(\theta \mid \theta_t)$
    $t \leftarrow t + 1$
until $\|\theta_t - \theta_{t-1}\| < \varepsilon$
return θ_t

apply the EM algorithm to the problem of computing the maximum likelihood estimates for $\boldsymbol{P}$ and $\boldsymbol{O}$, we consider separately the E- and M-steps of Algorithm 3.4.

The E-step requires computing $q(\boldsymbol{\theta} \mid \boldsymbol{\theta}')$. From (3.16),

$$q(\boldsymbol{\theta} \mid \boldsymbol{\theta}') = \sum_{\boldsymbol{x}_{0:T}} \left[\log \mu_0(x_0) + \sum_{t=0}^{T} \log \boldsymbol{O}(z_t \mid x_t) + \sum_{t=1}^{T} \log \boldsymbol{P}(x_t \mid x_{t-1}) \right] p(\boldsymbol{x}_{0:T} \mid \boldsymbol{z}_{0:T}; \boldsymbol{\theta}').$$

where the summation is taken over all possible sequences $\boldsymbol{x}_{0:T}$. Considering separately the three terms in square brackets,

- The term $\log \mu_0(x_0)$ depends only on x_0, contributing to $q(\boldsymbol{\theta} \mid \boldsymbol{\theta}')$ with

$$\sum_{\boldsymbol{x}_{0:T}} \log \mu_0(x_0) p(\boldsymbol{x}_{0:T} \mid \boldsymbol{z}_{0:T}; \boldsymbol{\theta}') = \sum_{x_0 \in \mathcal{X}} \log \mu_0(x_0) p(x_0 \mid \boldsymbol{z}_{0:T}; \boldsymbol{\theta}').$$

- Each term $\log \boldsymbol{O}(z_t \mid x_t)$ depends only on x_t. As a whole, they contribute to $q(\boldsymbol{\theta} \mid \boldsymbol{\theta}')$ with

$$\begin{aligned} \sum_{\boldsymbol{x}_{0:T}} \sum_{t=0}^{T} \log \boldsymbol{O}(z_t \mid x_t) p(\boldsymbol{x}_{0:T} \mid \boldsymbol{z}_{0:T}; \boldsymbol{\theta}') \\ = \sum_{x_0 \in \mathcal{X}} \log \boldsymbol{O}(z_0 \mid x_0) p(x_0 \mid \boldsymbol{z}_{0:T}; \boldsymbol{\theta}') \\ + \sum_{x_1 \in \mathcal{X}} \log \boldsymbol{O}(z_1 \mid x_1) p(x_1 \mid \boldsymbol{z}_{0:T}; \boldsymbol{\theta}') \\ \vdots \\ + \sum_{x_T \in \mathcal{X}} \log \boldsymbol{O}(z_T \mid x_T) p(x_T \mid \boldsymbol{z}_{0:T}; \boldsymbol{\theta}'). \end{aligned} \tag{3.19}$$

- Similarly, each term $\log \boldsymbol{P}(x_t \mid x_{t-1})$ depends only on x_t and x_{t-1}, yielding

$$\begin{aligned}\sum_{\boldsymbol{x}_{0:T}} \sum_{t=1}^{T} \log \boldsymbol{P}(x_t \mid x_{t-1}) p(\boldsymbol{x}_{0:T} \mid \boldsymbol{z}_{0:T}; \boldsymbol{\theta}') \\ = \sum_{x_0, x_1 \in \mathcal{X}} \log \boldsymbol{P}(x_1 \mid x_0) p(x_0, x_1 \mid \boldsymbol{z}_{0:T}; \boldsymbol{\theta}') \\ + \sum_{x_1, x_2 \in \mathcal{X}} \log \boldsymbol{P}(x_2 \mid x_1) p(x_1, x_2 \mid \boldsymbol{z}_{0:T}; \boldsymbol{\theta}') \\ \vdots \\ + \sum_{x_{T-1}, x_T \in \mathcal{X}} \log \boldsymbol{P}(x_T \mid x_{T-1}) p(x_{T-1}, x_T \mid \boldsymbol{z}_{0:T}; \boldsymbol{\theta}').\end{aligned} \tag{3.20}$$

Since $p(x_t \mid \boldsymbol{z}_{0:T}; \boldsymbol{\theta}') = \mu_{t|0:T}(x_t), t = 0, \ldots, T$ (for the HMM with parameters $\boldsymbol{\theta}'$), we can use the forward-backward algorithm described in Section 3.2 to compute $p(x_t \mid \boldsymbol{z}_{0:T}; \boldsymbol{\theta}')$. On the other hand, let

$$\mu_{t-1,t|0:T}(x_{t-1}, x_t) \stackrel{\text{def}}{=} \mathbb{P}\left[\mathrm{x}_{t-1} = x_{t-1}, \mathrm{x}_t = x_t \mid \mathbf{z}_{0:T} = \boldsymbol{z}_{0:T}\right],$$

for $t > 0$. We can repeat the derivations leading to (3.8) to get (see Exercise 3.5)

$$\mu_{t-1,t|0:T}(x_{t-1}, x_t) = \frac{\alpha_{t-1}(x_{t-1}) \beta_t(x_t) \boldsymbol{P}(x_t \mid x_{t-1}) \boldsymbol{O}(z_t \mid x_t)}{\sum_{x,y} \alpha_{t-1}(x) \beta_t(y) \boldsymbol{P}(y \mid x) \boldsymbol{O}(z_t \mid y)}. \tag{3.21}$$

Therefore, we can compute $\mu_{t-1,t|0:T}(x_{t-1}, x_t)$ directly from the HMM parameters, $\boldsymbol{P}$ and $\boldsymbol{O}$, and $\boldsymbol{\alpha}_{t-1}$ and $\boldsymbol{\beta}_t$, which can, in turn, be computed using the forward-backward algorithm. Note also that $p(x_t, x_{t-1} \mid \boldsymbol{z}_{0:T}; \boldsymbol{\theta}') = \mu_{t-1,t|0:T}(x_{t-1}, x_t)$ where, once again, the latter refers to the HMM with parameter vector $\boldsymbol{\theta}'$.

In short, the E-step consists of running the forward-backward algorithm to compute $\boldsymbol{\alpha}_t$ and $\boldsymbol{\beta}_t$ for $t = 0, \ldots, T$, from which $p(x_t, x_{t-1} \mid \boldsymbol{z}_{0:T}; \boldsymbol{\theta}')$ and $p(x_t \mid \boldsymbol{z}_{0:T}; \boldsymbol{\theta}')$—and then q—can be computed.

Regarding the M-step, it involves optimizing $q(\boldsymbol{\theta} \mid \boldsymbol{\theta}')$ with respect to $\boldsymbol{\theta}$. This means optimizing (3.19) and (3.20) with respect to $\boldsymbol{O}$ and $\boldsymbol{P}$, respectively. The process is similar to that adopted to estimate $\boldsymbol{P}$ and $\boldsymbol{O}$ from complete data: we use the method of Lagrange multipliers to get

$$\boldsymbol{P}(y \mid x) = \frac{\sum_{t=1}^{T} \mu_{t-1,t|0:T}(x, y)}{\sum_{t=0}^{T-1} \mu_{t|0:T}(x)} \tag{3.22}$$

and

$$\boldsymbol{O}(z \mid x) = \frac{\sum_{t=0}^{T} \mu_{t|0:T}(x) \mathbb{I}\left[z_t = z\right]}{\sum_{t=0}^{T} \mu_{t|0:T}(x)}. \tag{3.23}$$

Algorithm 3.5 summarizes the complete EM algorithm for HMMs.

Algorithm 3.5 The Baum-Welch algorithm.

Require: Sequence of observations $\boldsymbol{z}_{0:T}$
Initialize $\boldsymbol{P}$ and $\boldsymbol{O}$ arbitrarily
repeat
  for $t = 0, \ldots, T$ **do** ▷ E-Step
    Compute $\boldsymbol{\alpha}_t$ and $\boldsymbol{\beta}_t$ using Algorithm 3.2
    Compute $\boldsymbol{\mu}_{t|0:T}$
    Compute $\boldsymbol{\mu}_{t-1,t|0:T}$
  end for
  for $x, y \in \mathcal{X}, z \in \mathcal{Z}$ **do** ▷ M-Step
    Update $\boldsymbol{P}$ using (3.22)
    Update $\boldsymbol{O}$ using (3.23)
  end for
until convergence
return $\boldsymbol{P}$ and $\boldsymbol{O}$

3.3.2 Bayesian approach

Section 3.3.1 addressed the problem of estimating $\boldsymbol{P}$ and $\boldsymbol{O}$ from a sequence of observations $\boldsymbol{z}_{0:T}$. In a sense, the maximum-likelihood approach implicitly assumes that there are "true values" for $\boldsymbol{P}$ and $\boldsymbol{O}$; the observations $\boldsymbol{z}_{0:T}$ provide a random and indirect view to those values.

The Bayesian approach, in contrast, treats the unknown parameters, $\boldsymbol{P}$ and $\boldsymbol{O}$, as r.v.s $\mathbf{P}$ and $\mathbf{O}$ described by a distribution μ_{prior}. The goal is to refine the distribution μ_{prior} using the information implicitly provided by the observations $\boldsymbol{z}_{0:T}$. The resulting distribution is given directly by Bayes theorem:

$$\begin{aligned}\mu_{\text{post}}(\boldsymbol{\theta} \mid \boldsymbol{z}_{0:T}) &= \frac{\mathbb{P}\left[\mathbf{z}_{0:t} = \boldsymbol{z}_{0:T} \mid \Theta = \boldsymbol{\theta}\right] \mu_{\text{prior}}(\boldsymbol{\theta})}{\mathbb{P}\left[\mathbf{z}_{0:t} = \boldsymbol{z}_{0:T}\right]} \\ &= \frac{\mathbb{P}\left[\mathbf{z}_{0:t} = \boldsymbol{z}_{0:T} \mid \Theta = \boldsymbol{\theta}\right] \mu_{\text{prior}}(\boldsymbol{\theta})}{\int \mathbb{P}\left[\mathbf{z}_{0:t} = \boldsymbol{z}_{0:T} \mid \Theta = \boldsymbol{\theta}'\right] \mu_{\text{prior}}(\boldsymbol{\theta}') d\boldsymbol{\theta}'} \end{aligned} \tag{3.24}$$

where, once again, $\boldsymbol{\theta}$ is the HMM parameter vector. Given the *posterior distribution* in (3.24), it is possible to select $\boldsymbol{\theta}$ to optimize whichever criterion is more adequate. For example, if the goal is to minimize the expected squared error,

$$\boldsymbol{\theta}^* = \operatorname*{argmin}_{\boldsymbol{\theta}} \mathbb{E}_{\mu_{\text{prior}}}\left[\|\boldsymbol{\theta} - \Theta\|^2 \mid \mathbf{z}_{0:T} = \boldsymbol{z}_{0:T}\right] = \int \boldsymbol{\theta}\, \mu_{\text{post}}(\boldsymbol{\theta} \mid \boldsymbol{z}_{0:T}) d\boldsymbol{\theta}. \tag{3.25}$$

The estimator $\boldsymbol{\theta}^*$ in (3.25) is known as the *mean estimator*, and is often used in the absence of additional information. Another popular criterion (mostly in discrete settings) is minimizing the $0-1$-loss,

$$\boldsymbol{\theta}^* = \operatorname*{argmin}_{\boldsymbol{\theta}} \mathbb{E}_{\mu_{\text{prior}}}\left[\delta_{\Theta}(\boldsymbol{\theta}) \mid \mathbf{z}_{0:T} = \boldsymbol{z}_{0:T}\right] = \max_{\boldsymbol{\theta}} \mu_{\text{post}}(\boldsymbol{\theta} \mid \boldsymbol{z}_{0:T}), \tag{3.26}$$

where $\delta_x(y)$ is the Dirac delta centered in x. The estimator θ^* in (3.26) is known as the *maximum a posteriori* (MAP) estimator.

The difficulty in the Bayesian approach lies, however, in determining the posterior distribution μ_{post}—mostly due to the normalizing factor in the denominator of (3.24). One common approach is to use a sampling algorithm such as those discussed in Chapter 2. For example, the Metropolis-Hastings algorithm (Algorithm 2.2) uses the target distribution only to compute the acceptance probabilities in (2.19), which do not required computing the normalizing factor.

3.4 Derivation of the Kalman filter using HMMs ($\star$)

This section illustrates an application of the HMM theory to the derivation of a widely known filtering technique used in systems with continuous states—the *Kalman filter*. The derivation of the Kalman filter presented herein is by no means standard; it is included to illustrate the universal nature of the principles that underlie many apparently distinct areas.

We consider a more general version of the dynamic system from Section 2.3.3,

$$\mathrm{x}_{t+1} = a\mathrm{x}_t + \mathrm{w}_t,$$

where each noise term w_t is independent of $\mathbf{x}_{0:t-1}$ and $\mathrm{w}_{0:t-1}$ and follows a distribution $\mathsf{Normal}(0, \sigma_w^2)$, with $\sigma_w > 0$. Assume, moreover, that x_t cannot be observed directly. Instead, we can access only an observation z_t given by

$$\mathrm{z}_t = c\mathrm{x}_t + \mathrm{v}_t,$$

where, once again, each noise term v_t is independent of $\mathbf{x}_{0:t}$ and $\mathrm{w}_{0:t}$ and follows a distribution $\mathsf{Normal}(0, \sigma_v^2)$, with $\sigma_v > 0$. The resulting model can be cast as an HMM with (continuous) state space $\mathcal{X} = \mathbb{R}$ and (continuous) observation space $\mathcal{Z} = \mathbb{R}$, and described by the transition probabilities

$$\boldsymbol{P}(y \mid x) = \frac{1}{\sqrt{2\pi\sigma_w^2}} \exp\left[-\frac{(y-ax)^2}{2\sigma_w^2}\right]$$

and the observation probabilities

$$\boldsymbol{O}(z \mid x) = \frac{1}{\sqrt{2\pi\sigma_v^2}} \exp\left[-\frac{(z-cx)^2}{2\sigma_v^2}\right].$$

Finally, we assume that the initial state, x_0, follows a distribution $\mathsf{Normal}(\bar{x}_0, \sigma_{x_0})$.

Now given $U \subset \mathcal{X}$ and $V \subset \mathcal{Z}^{t+1}$, we define the distributions $\mu_{t|0:t}$ and α_t as verifying

$$\begin{aligned}\mathbb{P}\left[\mathrm{x}_t \in U \mid \mathbf{z}_{0:t} = \boldsymbol{z}_{0:t}\right] &= \int_U \mu_{t|0:t}(x)dx \\ \mathbb{P}\left[\mathrm{x}_t \in U, \mathbf{z}_{0:t} \in V\right] &= \int_U \int_V \alpha_t(x, \boldsymbol{z})d\boldsymbol{z}dv.\end{aligned}$$

It follows that

$$\mu_{t|0:t}(x) = \frac{\alpha_t(x, \boldsymbol{z}_{0:t})}{\int_{\mathcal{X}} \alpha_t(y, \boldsymbol{z}_{0:t})dy}$$

or, equivalently,

$$\alpha_t(x, \boldsymbol{z}_{0:t}) = K\mu_{t|0:t}(x), \tag{3.27}$$

for some constant $K > 0$. Suppose that we are given the distribution $\mu_{t|0:t}$ and the most recent observation, z_{t+1}, from which we want to determine $\mu_{t+1|0:t+1}$. We use the forward recursion discussed in Section 3.2 to get

$$\alpha_{t+1}(x, \boldsymbol{z}_{0:t+1}) = \boldsymbol{O}(z_{t+1} \mid x) \int_{\mathcal{X}} \alpha_t(y, \boldsymbol{z}_{0:t}) \boldsymbol{P}(x \mid y) dy. \tag{3.28}$$

Replacing (3.27) in (3.28) yields

$$\begin{aligned} \alpha_{t+1}(x, \boldsymbol{z}_{0:t+1}) &= K\boldsymbol{O}(z_{t+1} \mid x) \int_{\mathcal{X}} \mu_{t|0:t}(y) \boldsymbol{P}(x \mid y) dy \\ &= K\boldsymbol{O}(z_{t+1} \mid x) \mathbb{P}\left[\mathrm{x}_{t+1} = x \mid \mathbf{z}_{0:t} = \boldsymbol{z}_{0:T}\right]. \end{aligned} \tag{3.29}$$

Let us now consider the term $\mathbb{P}\left[\mathrm{x}_{t+1} = x \mid \mathbf{z}_{0:t} = \boldsymbol{z}_{0:T}\right]$. Assume that

$$\mu_{t|0:t} = \mathsf{Normal}(\bar{x}_{t|0:t}, \sigma^2_{t|0:t}), \tag{3.30}$$

for some $\bar{x}_{t|0:t}$ and $\sigma^2_{t|0:t}$. From (3.29), it becomes apparent that, conditioned on $\mathbf{z}_{0:t} = \boldsymbol{z}_{0:t}$, x_{t+1} also follows a normal distribution with mean

$$\bar{x}_{t+1|0:t} = a\bar{x}_{t|0:t} \tag{3.31a}$$

and variance

$$\sigma^2_{t+1|0:t} = a^2\sigma^2_{t|0:t} + \sigma^2_w. \tag{3.31b}$$

and we have that

$$\mathbb{P}\left[\mathrm{x}_{t+1} = x \mid \mathbf{z}_{0:t} = \boldsymbol{z}_{0:T}\right] = \frac{1}{\sqrt{2\pi\sigma^2_{t+1|0:t}}} \exp\left[-\frac{(x - \bar{x}_{t+1|0:t})^2}{2\sigma^2_{t+1|0:t}}\right].$$

It follows that $\alpha_{t+1}(x, \boldsymbol{z}_{0:t+1})$ is the product of two normal distributions and, as such, proportional to a normal distribution (see Appendix A), with mean

$$\bar{x}_{t+1|0:t+1} = \frac{\sigma^2_v\, \bar{x}_{t+1|0:t} + z\, c\, \sigma^2_x}{c^2\sigma^2_x + \sigma^2_v}$$

and variance

$$\sigma^2_{t+1|0:t+1} = \frac{\sigma^2_v\sigma^2_x}{c^2\sigma^2_x + \sigma^2_v}.$$

To write the expressions above in a more familiar form, let us define

$$K = \frac{c\sigma^2_x}{c^2\sigma^2_x + \sigma^2_v}$$

from which we get

$$\begin{aligned} \bar{x}_{t+1|0:t+1} &= \bar{x}_{t+1|0:t} + K(z - c\bar{x}_{t+1|0:t}), \\ \sigma^2_{t+1|0:t+1} &= (1 - Kc)\sigma^2_{t+1|0:t+1}. \end{aligned} \tag{3.32}$$

Since $\mu_{t+1|0:t+1}$ is just a normalized version of α_t, we can conclude that

$$\mu_{t+1|0:t+1} = \mathsf{Normal}(\bar{x}_{t+1|0:t+1}, \sigma^2_{t+1|0:t+1}).$$

Moreover, x_0 follows a normal distribution, by assumption. Therefore, an induction argument establishes that the assumption in (3.30) is fully justified.

We conclude by noting that (3.31) corresponds to the *prediction equations* of the Kalman filter, while (3.32) corresponds to the *update equations*. Therefore, an iteration of the Kalman filter can be seen as corresponding to one iteration of the forward recursion discussed in Section 3.2.

3.5 Bibliographical notes

Overall, our presentation follows the book of Cappé, Moulines, and Rydén (2005), although our presentation is kept significantly simpler. A lighter overview can be found in the classical survey of Rabiner (1989). The two HMM-specific algorithms surveyed—namely, the forward-backward and the Viterbi algorithms—were introduced, respectively, by Baum and Petrie (1966) and Gilbert (1959), and Viterbi (1967). The original EM algorithm was developed by Dempster, Laird, and Rubin (1977).

HMMs have been widely used to model partially observable processes. The examples in Section 3.1.2 provide just a brief glimpse of the vast literature that applies these models. For example, Nikovski and Nourbakhsh (2002) describe the motion of a mobile robot as an HMM that is then used to track the robot's underlying position. The HMM is learned using the Baum-Welch algorithm described in Section 3.3.1. The digital communication example is inspired in the work of Yonezaki, Yoshida, and Yagi (1998), while the keyword spotting example is inspired in **XXX**. Examples of the application of HMMs in biology are particularly numerous. We refer to the book of Durbin et al. (1998) for a fair survey. The example presented herein is based on the work of Krogh, Mian, and Haussler (1994).

3.6 Proofs

This section collects the proofs of the results used throughout the text.

Proof of Proposition 3.1

By definition,

$$\begin{aligned}
&\mathbb{E}\left[\prod_{n=1}^{N} F_{t_n}(\mathrm{z}_{t_n}) \mid \mathbf{x}_{t_1,\dots,t_N} = \boldsymbol{x}_{t_1,\dots,t_n}\right] \\
&= \sum_{\boldsymbol{z}_{t_1,\dots,t_N}} \prod_{n=1}^{N} F_{t_n}(z_{t_n}) \mathbb{P}\left[\mathbf{z}_{t_1,\dots,t_N} = \boldsymbol{z}_{t_1,\dots,t_N} \mid \mathrm{x}_{t_1,\dots,t_N} = \boldsymbol{x}_{t_1,\dots,t_N}\right] \\
&= \frac{1}{\rho(\boldsymbol{x}_{t_1,\dots,t_N})} \sum_{\boldsymbol{z}_{t_1,\dots,t_N}} \prod_{n=1}^{N} F_{t_n}(z_{t_n}) \mathbb{P}\left[\mathbf{z}_{t_1,\dots,t_N} = \boldsymbol{z}_{t_1,\dots,t_N}, \mathrm{x}_{t_1,\dots,t_N} = \boldsymbol{x}_{t_1,\dots,t_N}\right],
\end{aligned}$$

where $\rho(\boldsymbol{x}_{t_1,\dots,t_N}) = \mathbb{P}\left[\mathbf{x}_{t_1,\dots,t_N} = \boldsymbol{x}_{t_1,\dots,t_N}\right]$. Let $\mathcal{T} = \{t_1, \dots, t_N\}$. Using the properties of the Markov chain $\mathcal{M}_{\text{HMM}}$, we get

$$\begin{aligned}
&\mathbb{E}\left[\prod_{n=1}^{N} F_{t_n}(\mathrm{z}_{t_n}) \mid \mathbf{x}_{t_1,\dots,t_N} = \boldsymbol{x}_{t_1,\dots,t_n}\right] \\
&= \frac{1}{\rho(\boldsymbol{x}_{t_1,\dots,t_N})} \sum_{\boldsymbol{z}_t, t\in\mathcal{T}} \prod_{n=1}^{N} F_{t_n}(z_{t_n}) \sum_{\boldsymbol{x}_t, \boldsymbol{z}_t, t\notin\mathcal{T}} \mu_0(x_0) O(x_0, z_0) \prod_{t=1}^{t_N} P(x_{t-1}, x_t) O(x_t, z_t)
\end{aligned}$$

which, rearranging the terms, yields

$$\begin{aligned}
&\mathbb{E}\left[\prod_{n=1}^{N} F_{t_n}(\mathrm{z}_{t_n}) \mid \mathbf{x}_{t_1,\dots,t_N} = \boldsymbol{x}_{t_1,\dots,t_n}\right] \\
&= \frac{1}{\rho(\boldsymbol{x}_{t_1,\dots,t_N})} \sum_{\boldsymbol{x}_t, t\notin\mathcal{T}} \mu_0(x_0) \prod_{t=1}^{t_N} P(x_{t-1}, x_t) \sum_{\boldsymbol{z}_t, t=0,\dots,t_N} \prod_{t=0}^{t_N} O(x_t, z_t) \prod_{n=1}^{N} F_{t_n}(z_{t_n}) \\
&= \sum_{\boldsymbol{z}_t, t=0,\dots,t_N} \prod_{t=0}^{t_N} O(x_t, z_t) \prod_{n=1}^{N} F_{t_n}(z_{t_n}).
\end{aligned}$$

Finally, separating the summation in $\mathbf{z}_t$, we get

$$\begin{aligned}
&\mathbb{E}\left[\prod_{n=1}^{N} F_{t_n}(\mathrm{z}_{t_n}) \mid \mathbf{x}_{t_1,\dots,t_N} = \boldsymbol{x}_{t_1,\dots,t_n}\right] \\
&\quad= \sum_{\boldsymbol{z}_t, t\notin\mathcal{T}} \prod_{t\notin\mathcal{T}} O(x_t, z_t) \sum_{\boldsymbol{z}_t, t\in\mathcal{T}} \prod_{t\in\mathcal{T}} O(x_t, z_t) F_{t_n}(z_{t_n}) \\
&\quad= \sum_{\boldsymbol{z}_t, t\in\mathcal{T}} \prod_{t\in\mathcal{T}} O(x_t, z_t) F_{t_n}(z_{t_n}) \\
&\quad= \prod_{t\in\mathcal{T}} \sum_{\boldsymbol{z}_t} O(x_t, z_t) F_{t_n}(z_{t_n}).
\end{aligned}$$

Proof of Lemma 3.2

Using the total probability law,

$$\begin{aligned}
\beta_t(x) &\stackrel{\text{def}}{=} \mathbb{P}_{\mu_0}\left[\mathbf{z}_{t+1:T} = \boldsymbol{z}_{t+1:T} \mid \mathrm{x}_t = x\right] \\
&= \sum_{y\in\mathcal{X}} \mathbb{P}_{\mu_0}\left[\mathbf{z}_{t+1:T} = \boldsymbol{z}_{t+1:T} \mid \mathrm{x}_{t+1} = y, \mathrm{x}_t = x\right] \mathbb{P}_{\mu_0}\left[\mathrm{x}_{t+1} = x' \mid \mathrm{x}_t = x\right] \\
&= \sum_{y\in\mathcal{X}} \mathbb{P}_{\mu_0}\left[\mathbf{z}_{t+1:T} = \boldsymbol{z}_{t+1:T} \mid \mathrm{x}_{t+1} = y\right] \boldsymbol{P}(y \mid x),
\end{aligned}$$

where we used the fact that $\mathbf{z}_{t+1:T}$ is conditionally independent of x_t given x_{t+1} (see Exercise 3.2). Breaking down the first term in the summation yields

$$\begin{aligned}
\beta_t(x) = \sum_{y\in\mathcal{X}} &\mathbb{P}\left[\mathrm{z}_{t+1} = z_{t+1} \mid \mathbf{z}_{t+2:T} = \boldsymbol{z}_{t+2:T}, \mathrm{x}_{t+1} = y\right] \\
&\times \mathbb{P}\left[\mathbf{z}_{t+2:T} = \boldsymbol{z}_{t+2:T} \mid \mathrm{x}_{t+1} = y\right] \boldsymbol{P}(y \mid x).
\end{aligned}$$

Finally, using the fact that z_{t+1} is fully determined from x_{t+1},

$$\beta_t(x) = \sum_{y \in \mathcal{X}} \mathbb{P}\left[\mathrm{z}_{t+1} = z_{t+1} \mid \mathrm{x}_{t+1} = y\right] \beta_{t+1}(y) \boldsymbol{P}(y \mid x).$$

Proof of Lemma 3.3

Using standard manipulations, we have that

$$\begin{aligned} m_t(x) &\stackrel{\text{def}}{=} \max_{\boldsymbol{x}_{0:t-1}} \mathbb{P}_{\mu_0}\left[\mathrm{x}_t = x, \mathbf{x}_{0:t-1} = \boldsymbol{x}_{0:t-1}, \mathbf{z}_{0:t} = \boldsymbol{z}_{0:t}\right] \\ &= \max_{\boldsymbol{x}_{0:t-1}} \Big\{ \mathbb{P}_{\mu_0}\left[\mathrm{z}_t = z_t \mid \mathrm{x}_t = x, \mathbf{x}_{0:t-1} = \boldsymbol{x}_{0:t-1}, \mathbf{z}_{0:t-1} = \boldsymbol{z}_{0:t-1}\right] \\ &\qquad \times \mathbb{P}_{\mu_0}\left[\mathrm{x}_t = x, \mathbf{x}_{0:t-1} = \boldsymbol{x}_{0:t-1}, \mathbf{z}_{0:t-1} = \boldsymbol{z}_{0:t-1}\right] \Big\}. \end{aligned}$$

From (3.2), the first term inside the max is just $\boldsymbol{O}(z_t \mid x)$, which does not depend on $\boldsymbol{x}_{0:t-1}$. Therefore,

$$\begin{aligned} &m_t(x) \\ &= \boldsymbol{O}(z_t \mid x) \max_{\boldsymbol{x}_{0:t-1}} \mathbb{P}_{\mu_0}\left[\mathrm{x}_t = x, \mathbf{x}_{0:t-1} = \boldsymbol{x}_{0:t-1}, \mathbf{z}_{0:t-1} = \boldsymbol{z}_{0:t-1}\right] \\ &= \boldsymbol{O}(z_t \mid x) \max_{\boldsymbol{x}_{0:t-1}} \Big\{ \mathbb{P}_{\mu_0}\left[\mathrm{x}_t = x \mid \mathrm{x}_{t-1} = x_{t-1}, \mathbf{x}_{0:t-2} = \boldsymbol{x}_{0:t-2}, \mathbf{z}_{0:t-1} = \boldsymbol{z}_{0:t-1}\right] \\ &\qquad \times \mathbb{P}_{\mu_0}\left[\mathrm{x}_{t-1} = x_{t-1}, \mathbf{x}_{0:t-2} = \boldsymbol{x}_{0:t-2}, \mathbf{z}_{0:t-1} = \boldsymbol{z}_{0:t-1}\right] \Big\}. \end{aligned}$$

Using the Markov property, we finally get

$$\begin{aligned} &m_t(x) \\ &= \boldsymbol{O}(z_t \mid x) \max_{x_{t-1}} \Big\{ \boldsymbol{P}(x \mid x_{t-1}) \max_{\boldsymbol{x}_{0:t-2}} \mathbb{P}_{\mu_0}\left[\mathrm{x}_{t-1} = x_{t-1}, \mathbf{x}_{0:t-2} = \boldsymbol{x}_{0:t-2}, \mathbf{z}_{0:t-1} = \boldsymbol{z}_{0:t-1}\right] \Big\} \\ &= \boldsymbol{O}(z_t \mid x) \max_{x_{t-1}} \Big\{ \boldsymbol{P}(x \mid x_{t-1}) m_{t-1}(x_{t-1}) \Big\}. \end{aligned}$$

Proof of Proposition 3.4

Let

$$q(\theta \mid \theta') = \mathbb{E}\left[\log p(x, z; \theta) \mid z, \theta'\right] = \sum_{x \in \mathcal{X}} \log p(x, z \mid \theta) p(x \mid z, \theta').$$

We have that

$$q(\theta \mid \theta') = \log p_{\mathrm{z}}(z; \theta) - H(\theta \mid \theta')$$

where

$$H(\theta \mid \theta') \stackrel{\text{def}}{=} -\sum_{x \in \mathcal{X}} \log p(x \mid z, \theta) p(x \mid z, \theta').$$

Clearly, if $\theta = \theta'$, H reduces to the entropy of the distribution $p(\cdot \mid z, \theta)$. On the other hand,

$$\begin{aligned} H(\theta \mid \theta') - H(\theta' \mid \theta') &= \sum_{x \in \mathcal{X}} \log \frac{p(x \mid z, \theta')}{p(x \mid z, \theta)} p(x \mid z, \theta') \\ &= \mathrm{KL}\big(p(\cdot \mid z, \theta') \parallel p(\cdot \mid z, \theta)\big) \geq 0. \end{aligned}$$

Therefore,

$$\begin{aligned} q(\theta \mid \theta') - q(\theta' \mid \theta') &= \log p_{\mathrm{z}}(z;\theta) - \log p_{\mathrm{z}}(z;\theta') - H(\theta \mid \theta') + H(\theta' \mid \theta') \\ &\leq \log p_{\mathrm{z}}(z;\theta) - \log p_{\mathrm{z}}(z;\theta'). \end{aligned}$$

3.7 Exercises

Exercise 3.1.
Using Proposition 3.1 show that, in an HMM,

$$\mathbb{P}\left[\mathrm{z}_t = z \mid \mathbf{x}_{0:t} = \boldsymbol{x}_{0:t}, \mathbf{z}_{0:t-1} = \boldsymbol{z}_{0:t-1}\right] = \mathbb{P}\left[\mathrm{z}_t = z \mid \mathrm{x}_t = x_t\right].$$

Exercise 3.2.
Given an HMM $\mathcal{M} = (\mathcal{X}, \mathcal{Z}, \boldsymbol{P}, \boldsymbol{O})$ and any $t, T > 0$ and $0 < t_0 < T$, show that

$$\mathbb{P}\left[\mathrm{z}_{T+t} = z \mid \mathrm{x}_T = x, \mathrm{x}_{t_0} = y\right] = \mathbb{P}\left[\mathrm{z}_{T+t} = z \mid \mathrm{x}_T = x\right]$$

for any $x, y \in \mathcal{X}$ and $z \in \mathcal{Z}$, i.e., z_{T+t} is conditionally independent of x_{t_0} given x_T.

Exercise 3.3.
Show that, given an HMM $(\mathcal{X}, \mathcal{Z}, \boldsymbol{P}, \boldsymbol{O}, \mu_0)$ and a sequence of observations $\boldsymbol{z}_{0:T}$,

$$\mu_{T+1|0:T}(x) = \sum_{y \in \mathcal{X}} \boldsymbol{P}(x \mid y) \mu_{T|0:T}(y),$$

for all $x \in \mathcal{X}$.

Exercise 3.4.
In this exercise, we will determine the maximum likelihood estimate for the HMM parameter $\boldsymbol{P}$ using the method of Lagrange multipliers. The parameter $\boldsymbol{O}$ can be computed in exactly the same manner.

As seen in Section 3.3.1,

$$\log \ell(\boldsymbol{x}_{0:T}, \boldsymbol{z}_{0:T} \mid \boldsymbol{\theta}) = \log \mu_0(x_0) + \sum_{t=0}^{T} \log \boldsymbol{O}(z_t \mid x_t) + \sum_{t=1}^{T} \log \boldsymbol{P}(x_t \mid x_{t-1}).$$

Only the second summation depends on $\boldsymbol{P}$. Therefore, determining the parameter $\boldsymbol{P}$ that maximizes $\log \ell(\boldsymbol{x}_{0:T}, \boldsymbol{z}_{0:T} \mid \boldsymbol{\theta})$ reduces to solving the following constrained optimization problem:

$$\begin{aligned} &\text{maximize} && \sum_{t=1}^{T} \log \boldsymbol{P}(x_t \mid x_{t-1}) && \\ &\text{subject to} && \sum_{y \in \mathcal{X}} \boldsymbol{P}(y \mid x) = 1 && x \in \mathcal{X} \\ & && \boldsymbol{P}(y \mid x) \geq 0 && x, y \in \mathcal{X}. \end{aligned}$$

The corresponding Lagrangian is given by

$$L(\boldsymbol{P}; \boldsymbol{\lambda}) = \sum_{t=1}^{T} \log \boldsymbol{P}(x_t \mid x_{t-1}) + \sum_{x,y \in \mathcal{X}} \lambda_x (1 - \boldsymbol{P}(y \mid x)).$$

(a) Letting $P_{x,y} = \boldsymbol{P}(y \mid, x)$, show that

$$\frac{\partial L}{\partial P_{x,y}} = \frac{N_{x,y}}{P_{x,y}} - \lambda_x,$$

where $N_{x,y}$ denotes the number of times that a transition was observed from x to y in $\boldsymbol{x}_{0:T}$.

Suggestion: Use the fact that

$$\sum_{t=1}^{T} \log \boldsymbol{P}(x_t \mid x_{t-1}) = \sum_{t=1}^{T} \sum_{x,y \in \mathcal{X}} \log \boldsymbol{P}(y \mid x) \mathbb{I}\left[x_t = j, x_{t-1} = i\right].$$

(b) Using the result from (a), show that the maximum likelihood estimate for $\boldsymbol{P}$ is given by

$$\boldsymbol{P}(y \mid x) = \frac{N_{x,y}}{N_x},$$

where N_x denotes the number of times that state x was visited in $\boldsymbol{x}_{0:T}$.

Exercise 3.5.
Prove the equality in (3.21).