Chapter 8

Hidden Markov models

By the end of this chapter, we will have presented
- *Statistical ways of modeling dynamics*
- *Fundamentals of hidden Markov models*
- *Specialized computational algorithms*

In this chapter we are exclusively concerned with modeling *time dependent* measurements. We revisit some of the systems introduced in chapter 2 and present, in a unified framework, several methods to combine dynamic and observation likelihoods. It will become apparent soon that computational tractability is by no means guaranteed in time dependent problems and often we need to consider specialized algorithms that built upon or extend those of chapter 5. For this reason, we also present appropriate computational methods that can be used to train the resulting models in an efficient manner. In this chapter, we focus on modeling *discrete systems* evolving in *discrete time*; while, we present more general systems in the subsequent chapters.

8.1 Introduction

Throughout this chapter, we consider a system that may access a number of discrete states similar to the systems seen in section 2.4. For convenience, throughout this chapter, we denote the constitutive states of the system with σ_m, and use numerical labels $m = 1 : M$ to distinguish between them. The number of different states, M, that the system may occupy depends upon the problem at hand.

When such a system evolves in time, these fundamental questions arise: *"what is the sequence of successive states the system occupies across time?"* and *"what are the properties of the states occupied across time?"*. To help formulate our questions more precisely, we consider ordered time levels t_n, indexed $n = 1 : N$, and use s_n to denote the state occupied by the system at t_n. That is, for a given n, the passing state s_n takes its value from the constitutive states $\sigma_{1:M}$. Thus, our questions about the system at hand can be answered by estimating the trajectory $s_{1:N}$ and the properties of each σ_m.

Note 8.1: Label and index conventions

Just as with the systems we encountered earlier in note 2.5, only the labeled states σ_m carry meaning while the m labels themselves are otherwise only an arbitrary index. Such distinction does not carry over to the time level indices, n. By convention, our time levels are ordered $t_{n-1} < t_n$ indicating that, contrary to the m labels, our n indices carry information.

A critical aspect of modeling time evolving systems is to recognize that states are not directly observed. Rather only a version of them corrupted by measurement noise is typically assessed experimentally. Thus, whenever a system occupies a state σ_m, it generates observations according to a probability distribution $\mathbb{F}_{\sigma_m}$, or its associated density $F_{\sigma_m}(w)$, unique to σ_m.

To derive a concise formulation incorporating measurement noise, we will assume the case where only *one* observation, denoted by w_n, is gathered per time level t_n. In other words, our *assessment rule* reads

$$w_n|s_n \sim \mathbb{F}_{s_n}, \tag{8.1}$$

with the understanding that each observation w_n may consist of more than one scalar quantity, *i.e.*, our individual observations may be array-valued.

As we have seen in section 7.1.1, a more convenient way of representing eq. (8.1) is through a mother distribution $\mathbb{G}_\phi$ with state specific parameters ϕ_{σ_m}. In this case, our assessment rules take the form

$$w_n|s_n, \boldsymbol{\phi} \sim \mathbb{G}_{\phi_{\sigma_m}}. \tag{8.2}$$

Here, for convenience, we use $\boldsymbol{\phi}$ to gather all emission parameters $\phi_{\sigma_{1:M}}$. Equation (8.1) or eq. (8.2) provide a means to incorporate measurement and, unlike in chapter 7, the *order in which observations are made* provides important information on *dynamics* including, say, the probability of transitioning to particular states σ_m at subsequent time levels.

We have seen in chapter 2 that dynamics for systems with discrete state-spaces evolving in discrete time are best described by assigning appropriate probability distributions on the passing states s_1 and $s_n|s_{n-1}$ dictating the *initialization* and *transition rules*. Next, we discuss some modeling options to consider when selecting such distributions.

8.2 The Hidden Markov Model

8.2.1 Modeling dynamics

From the modeling point of view, the simplest and often most convenient way to incorporate dynamics into an observation model is to adopt transition probabilities between any pair σ_m and $\sigma_{m'}$ of states in the system's state-space. That is, such formulations generally lead to intuitive and computationally tractable problems.

In general, our system's transitions need not be reversible. As such, we may adopt different probabilities for transitions $\sigma_m \to \sigma_{m'}$ and $\sigma_{m'} \to \sigma_m$. In the most general case, we denote with $\pi_{\sigma_m \to \sigma_{m'}}$ the probability of the system starting at σ_m and, within *one* time step, transitioning to $\sigma_{m'}$. In this setup, some $\pi_{\sigma_m \to \sigma_{m'}}$ can be zero indicating that the system cannot undergo transitions $\sigma_m \to \sigma_{m'}$ in a single step.

To facilitate the presentation that follows, we gather all transition probabilities out of the same state σ_m into an array $\boldsymbol{\pi}_{\sigma_m} = [\pi_{\sigma_m \to \sigma_1}, \pi_{\sigma_m \to \sigma_2}, \cdots, \pi_{\sigma_m \to \sigma_M}]$. Since once the system departs from any σ_m, it necessarily lands somewhere *within* the state-space $\sigma_{1:M}$, the individual transition probabilities assigned must satisfy $\sum_{m=1}^{M} \pi_{\sigma_m \to \sigma_{m'}} = 1$. Consequently, each $\boldsymbol{\pi}_{\sigma_m}$ is, in fact, a *probability* array.

Note 8.2: Transition probability matrix

To simplify the notation, we tabulate the transition probabilities into

$$\begin{array}{c} \\ \sigma_1 \\ \sigma_2 \\ \vdots \\ \sigma_M \end{array} \begin{array}{c} \begin{array}{cccc} \sigma_1 & \sigma_2 & \cdots & \sigma_M \end{array} \\ \begin{bmatrix} \pi_{\sigma_1 \to \sigma_1} & \pi_{\sigma_1 \to \sigma_2} & \cdots & \pi_{\sigma_1 \to \sigma_M} \\ \pi_{\sigma_2 \to \sigma_1} & \pi_{\sigma_2 \to \sigma_2} & \cdots & \pi_{\sigma_2 \to \sigma_M} \\ \vdots & \vdots & \ddots & \vdots \\ \pi_{\sigma_M \to \sigma_1} & \pi_{\sigma_M \to \sigma_2} & \cdots & \pi_{\sigma_M \to \sigma_M} \end{bmatrix} \end{array} = \begin{bmatrix} \boldsymbol{\pi}_{\sigma_1} \\ \boldsymbol{\pi}_{\sigma_2} \\ \vdots \\ \boldsymbol{\pi}_{\sigma_M} \end{bmatrix} = \boldsymbol{\Pi}.$$

This matrix is similar to the transition probability matrices we encountered in chapter 2.

Under this formulation, dynamics are represented generically by the transition rules

$$s_n|s_{n-1} \sim \text{Categorical}_{\sigma_{1:M}}\left(\boldsymbol{\pi}_{s_{n-1}}\right). \tag{8.3}$$

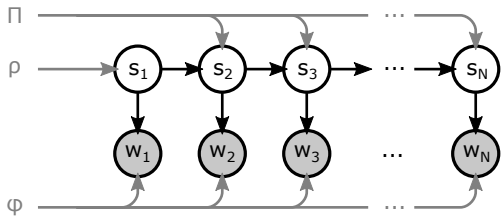

Figure 8.1: Graphical representation of a HMM. Here, the parameters $\boldsymbol{\rho}, \boldsymbol{\Pi}$ and $\boldsymbol{\phi}$ are assumed known.

The system's initial state s_1 is not included in eq. (8.3) as there is no predecessor passing state. To complete our formulation, we need to adopt separate probabilities for s_1, which we denote with $\boldsymbol{\rho} = [\rho_{\sigma_1}, \rho_{\sigma_2}, \cdots, \rho_{\sigma_M}]$. As such, eq. (8.3) is combined with the initialization rule

$$s_1 \sim \text{Categorical}_{\sigma_{1:M}}(\boldsymbol{\rho}),$$

thereby completing the description of the system's dynamics.

Note 8.3: Deterministic initialization

When the initial state of our dynamical system is specified deterministically, we may still maintain the same formulation by simply setting $\rho_{\sigma_m} = 1$ for the constitutive state σ_m from which the system is initialized and, thus, $\rho_{\sigma_{m'}} = 0$ for every other state. For example, for a system initialized at σ_2, the initial probabilities are $\boldsymbol{\rho} = [0, 1, 0, \cdots, 0]$.

8.2.2 Modeling overview

The statistical model we described so far is depicted graphically in fig. 8.1 and is summarized in

$$s_1|\boldsymbol{\rho} \sim \text{Categorical}_{\sigma_{1:M}}(\boldsymbol{\rho}), \tag{8.4}$$

$$s_n|s_{n-1}, \boldsymbol{\Pi} \sim \text{Categorical}_{\sigma_{1:M}}(\boldsymbol{\pi}_{s_{n-1}}), \qquad n = 2:N \tag{8.5}$$

$$w_n|s_n, \boldsymbol{\phi} \sim \mathbb{G}_{\phi_{s_n}}, \qquad n = 1:N \tag{8.6}$$

where, for clarity, we emphasize the dependencies upon the parameters $\boldsymbol{\rho}, \boldsymbol{\Pi}, \boldsymbol{\phi}$ by conditioning explicitly upon them. The three equations model: initialization, transitions, and observations of the system under study, respectively, and combined with a clear specification of the state-space $\sigma_{1:M}$, provide a complete description of our problem.

The model just defined is termed the *hidden Markov model* (HMM). It contains two sets of parameters: dynamical $\boldsymbol{\rho}, \boldsymbol{\Pi}$ and observational $\boldsymbol{\phi}$. From the inference point of view, the trajectory $s_{1:N}$ gathers latent (*i.e.*, hidden) variables; $w_{1:N}$ gathers measurements; and, $\boldsymbol{\rho}, \boldsymbol{\Pi}, \boldsymbol{\phi}$ gather parameters that, depending on context, may either be known or unknown. The dependencies among variables are illustrated in fig. 8.1.

The HMM's formulation in eqs. (8.4) to (8.6) is very general and, for this reason, is one of the mostly widely used models in time series analysis. Since it is already formulated in generative form, when tackling direct problems, it is straightforward to use this model for the simulation of synthetic measurements $w_{1:N}$ via ancestral sampling, algorithm 1.3. However, as we will see shortly, the HMM is mostly useful in tackling inverse problems. In particular, an inverse formulation of the HMM can be used to shed light on the following questions:

1. Given observations $w_{1:N}$ and parameter values $\boldsymbol{\rho}, \boldsymbol{\Pi}, \boldsymbol{\phi}$ what is the likelihood of $w_{1:N}$?
2. Given observations $w_{1:N}$ and parameter values $\boldsymbol{\rho}, \boldsymbol{\Pi}, \boldsymbol{\phi}$ what are the passing states $s_{1:N}$?
3. Given observations $w_{1:N}$ what are the values of the parameters $\boldsymbol{\rho}, \boldsymbol{\Pi}, \boldsymbol{\phi}$?

These questions are commonly referred to as: *evaluation*, *decoding*, and *estimation*, respectively. To answer them, we can follow two complementary routes: frequentist and Bayesian. We describe these separately in the subsequent sections.

Note 8.4: Time indexing and missing observations

With the indexing convention adopted, we designate with $n = 1$ the *earliest* time level associated with an observation and $n = N$ with the *latest* one. Further, we assume that *every* intermediate time level $n = 2, \ldots, N-1$ is also associated with an observation. This is a measurement-centric convention in the sense that the timing schedule of the measurement acquisition protocol determines the precise structure of the hidden state sequence.

Occasionally we might encounter situations where we need to incorporate into our formulation time levels *without* observations, for example when modeling measurements collected at *irregular times*. In such circumstances, we may generalize our formulation in at least two possible ways:

1. Use the same indexing convention with precisely one observation per time level and adopt *time dependent kinetics*, for example by explicitly requiring transition probabilities $\pi_{n,\sigma_m \to \sigma_{m'}}$ that may change across time levels.
2. Use an indexing scheme with *redundant time levels*. In particular, we may chose to maintain a hidden state sequence at a finer, but regular, time spacing and associate only some of the passing states with the observations while leaving the others unassociated with.

The theory we present next can readily accommodate both cases above with minor modification.

8.3 The Hidden Markov Model in the frequentist paradigm

The concepts of this section are direct extensions of chapter 2. As we have already introduced them, here we treat mostly computational aspects specifically tailored to HMMs.

8.3.1 Evaluation of the likelihood

Evaluation of a HMM requires the computation of the (marginal) likelihood

$$p\left(w_{1:N}|\boldsymbol{\rho}, \boldsymbol{\Pi}, \boldsymbol{\phi}\right) = \sum_{s_{1:N}} p\left(w_{1:N}, s_{1:N}|\boldsymbol{\rho}, \boldsymbol{\Pi}, \boldsymbol{\phi}\right)$$

with the sum taken over every possible state sequence $s_{1:N}$. Naive evaluation of this enormous sum, where a term $p\left(w_{1:N}, s_{1:N}|\boldsymbol{\rho}, \boldsymbol{\Pi}, \boldsymbol{\phi}\right) = p\left(w_{1:N}|s_{1:N}, \boldsymbol{\phi}\right) p\left(s_{1:N}|\boldsymbol{\rho}, \boldsymbol{\Pi}\right)$ is computed for each possible trajectory $s_{1:N}$ and summed requires the evaluation and addition of M^N terms. This is prohibitively large even for small problems. Instead, below, we describe a particular computational scheme, termed *filtering*, scaling as $M^2 N$.

Instead of completing over the entire state sequence $s_{1:N}$, the computation of the likelihood is achieved most efficiently by completing first only with respect to the terminal passing state s_N as follows

$$p\left(w_{1:N}|\boldsymbol{\rho}, \boldsymbol{\Pi}, \boldsymbol{\phi}\right) = \sum_{s_N} p\left(w_{1:N}, s_N|\boldsymbol{\rho}, \boldsymbol{\Pi}, \boldsymbol{\phi}\right) = \sum_{s_N} \mathcal{A}_N\left(s_N\right). \tag{8.7}$$

This sum is readily computed so long as $\mathcal{A}_N(s_N)$, called *forward variables*, are available for all possible values of s_N. Written explicitly these are $\mathcal{A}_N(\sigma_1), \mathcal{A}_N(\sigma_2), \ldots, \mathcal{A}_N(\sigma_M)$. We can define forward variables, more generally, for any time level by the joint distributions

$$\mathcal{A}_n(s_n) = p(w_{1:n}, s_n|\boldsymbol{\rho}, \boldsymbol{\Pi}, \boldsymbol{\phi}) \tag{8.8}$$

and compute them recursively. Our recursion relies on

$$\mathcal{A}_n(s_n) = G_{\phi_{s_n}}\left(w_n\right) \sum_{s_{n-1}} \pi_{s_{n-1} \to s_n} \mathcal{A}_{n-1}(s_{n-1}), \qquad n = 2 : N \tag{!8.9}$$

and requires the initial condition $\mathcal{A}_1(s_1)$ to iterate forward. As a direct consequence of the definition of $\mathcal{A}_1(s_1)$, the initial condition is

$$\mathcal{A}_1(s_1) = G_{\phi_{s_1}}\left(w_1\right) \rho_{s_1}. \tag{!8.10}$$

The steps involved are summarized in algorithm 8.1.

Algorithm 8.1: Forward recursion for the HMM (unstable version)

Given observations $w_{1:N}$ and parameters $\boldsymbol{\rho}, \boldsymbol{\Pi}, \boldsymbol{\phi}$, the forward terms $\mathcal{A}_n(\sigma_m)$, for each time level n and each state σ_m, are computed as follows:

- At $n = 1$ initialize with

$$\mathcal{A}_1(\sigma_m) = G_{\phi_{\sigma_m}}(w_1)\rho_{\sigma_m}.$$

- For $n = 2 : N$ compute recursively

$$\mathcal{A}_n(\sigma_m) = G_{\phi_{\sigma_m}}(w_n) \sum_{m'=1}^{M} \pi_{\sigma_{m'} \to \sigma_m} \mathcal{A}_{n-1}(\sigma_{m'}).$$

Upon completion, the algorithm yields every $\mathcal{A}_n(\sigma_m)$ which may be tabulated as follows

$$\begin{array}{c} \\ t_1 \\ t_2 \\ \vdots \\ t_n \\ \vdots \\ t_N \end{array} \begin{array}{c} \begin{array}{cccc} \sigma_1 & \sigma_2 & \cdots & \sigma_M \end{array} \\ \begin{bmatrix} \mathcal{A}_1(\sigma_1) & \mathcal{A}_1(\sigma_2) & \cdots & \mathcal{A}_1(\sigma_M) \\ \mathcal{A}_2(\sigma_1) & \mathcal{A}_2(\sigma_2) & \cdots & \mathcal{A}_2(\sigma_M) \\ \vdots & \vdots & \ddots & \vdots \\ \mathcal{A}_n(\sigma_1) & \mathcal{A}_n(\sigma_2) & \cdots & \mathcal{A}_n(\sigma_M) \\ \vdots & \vdots & \ddots & \vdots \\ \mathcal{A}_N(\sigma_1) & \mathcal{A}_N(\sigma_2) & \cdots & \mathcal{A}_N(\sigma_M) \end{bmatrix} \end{array} \begin{array}{c} \\ \mathcal{A}_1(s_1) \\ \mathcal{A}_2(s_2) \\ \vdots \\ \mathcal{A}_n(s_n) \\ \vdots \\ \mathcal{A}_N(s_N) \end{array}$$

Note 8.5: Vectorization

Gathering the forward terms of the same time level in *row* arrays

$$\mathbb{A}_n = \begin{bmatrix} \mathcal{A}_n(\sigma_1) & \mathcal{A}_n(\sigma_2) & \cdots & \mathcal{A}_n(\sigma_M) \end{bmatrix},$$

and similarly for the likelihood terms

$$\mathbb{\Gamma}_n = \begin{bmatrix} G_{\phi_{\sigma_1}}(w_n) & G_{\phi_{\sigma_2}}(w_n) & \cdots & G_{\phi_{\sigma_M}}(w_n) \end{bmatrix},$$

the filtering recursions can be executed in vectorized form

$$\begin{aligned} &\mathbb{A}_1 = \mathbb{\Gamma}_n \odot \boldsymbol{\rho}, \\ &\mathbb{A}_n = \mathbb{\Gamma}_n \odot (\mathbb{A}_{n-1}\boldsymbol{\Pi}), \qquad\qquad n = 2 : N \end{aligned}$$

where $\odot$ denotes the Hadamard (element-wise) product. If, instead of a row array, we represent $\mathbb{\Gamma}_n$ as a diagonal matrix $\mathbb{D}_{\mathbb{\Gamma}_n}$, then the filtering recursions take a more conventional form

$$\begin{aligned} &\mathbb{A}_1 = \boldsymbol{\rho}\, \mathbb{D}_{\mathbb{\Gamma}_n}, \\ &\mathbb{A}_n = (\mathbb{A}_{n-1}\boldsymbol{\Pi})\, \mathbb{D}_{\mathbb{\Gamma}_n}, \qquad\qquad n = 2 : N \end{aligned}$$

that only use ordinary matrix-vector operations. From these two sets of filtering equations, the first is preferred for computational implementations while the second is more convenient in theoretical derivations.

8.3.2 Decoding the state sequence

Decoding a HMM may be achieved in at least two meaningful ways. Depending on the problem specifics, we might be interested in finding a single passing state s_n^* maximizing the marginal $p(s_n|w_{1:N}, \boldsymbol{\rho}, \boldsymbol{\Pi}, \boldsymbol{\phi})$; or in finding the sequence $s_{1:N}^\sharp$ maximizing the joint $p(s_{1:N}|w_{1:N}, \boldsymbol{\rho}, \boldsymbol{\Pi}, \boldsymbol{\phi})$. Generally, individual states s_n^* are useful in problems where optimal passing states for a *particular time level* are sought. By contrast, $s_{1:N}^\sharp$ is useful in problems where the optimal trajectory over the *entire time course* is sought.

Note 8.6: Marginal and joint state sequences

Collecting passing states s_n^* across all time levels, we may form a state sequence $s_{1:N}^*$. This sequence, however, must be used with caution as it might violate the kinetics in $\boldsymbol{\Pi}$. In particular, since $s_{1:N}^*$ considers each s_n^* *irrespective* of s_{n-1}^*; it may very well contain transitions $s_{n-1}^* \to s_n^*$ coinciding with forbidden probabilities, *i.e.*, $\pi_{s_{n-1}^* \to s_n^*} = 0$. By contrast, $s_{1:N}^\sharp$ is *guaranteed* to obey the kinetics in $\boldsymbol{\Pi}$ as any sequence containing prohibited transitions $\pi_{s_{n-1}^\sharp \to s_n^\sharp}$ is, as we will see, excluded by construction.

Marginal decoding

To obtain each s_n^*, termed *marginal decoding*, we need to compute $p(s_n|w_{1:N}, \boldsymbol{\rho}, \boldsymbol{\Pi}, \boldsymbol{\phi})$. This can be efficiently achieved using the $\mathcal{A}_n(s_n)$ variables through the relations

$$p(s_N|w_{1:N}, \boldsymbol{\rho}, \boldsymbol{\Pi}, \boldsymbol{\phi}) \propto \mathcal{A}_N(s_N), \tag{!8.11}$$

$$p(s_n|w_{1:N}, \boldsymbol{\rho}, \boldsymbol{\Pi}, \boldsymbol{\phi}) \propto \mathcal{A}_n(s_n)\mathcal{B}_n(s_n), \qquad n = 1 : N-1 \tag{!8.12}$$

where we define *backward variables* by

$$\mathcal{B}_n(s_n) = p\left(w_{n+1:N}|s_n, \boldsymbol{\Pi}, \boldsymbol{\phi}\right), \qquad n = 1 : N-1. \tag{8.13}$$

In both eq. (!8.11) and eq. (!8.12), the missing proportionality constants do not affect the maximization of $p(s_n|w_{1:N}, \boldsymbol{\rho}, \boldsymbol{\Pi}, \boldsymbol{\phi})$ and, as such, need not be computed. Given $\mathcal{A}_n(s_n)$ and $\mathcal{B}_n(s_n)$, the sequence $s_{1:N}^*$ is readily computed by

$$s_n^* = \underset{\sigma_m}{\operatorname{argmax}}\, \mathcal{A}_n(\sigma_m)\mathcal{B}_n(\sigma_m).$$

The terms $\mathcal{B}_n(s_n)$, similarly to $\mathcal{A}_n(s_n)$, may be computed recursively. In this case, the recursion relies on

$$\mathcal{B}_n(s_n) = \sum_{s_{n+1}} \mathcal{B}_{n+1}(s_{n+1}) G_{\phi_{s_{n+1}}}\left(w_{n+1}\right) \pi_{s_n \to s_{n+1}}, \qquad n = 1 : N-1 \tag{!8.14}$$

and requires the final condition $\mathcal{B}_N(s_N)$ to iterate backward. By comparing eq. (!8.11) and eq. (!8.12), we fulfill the terminal condition, conventionally, by setting

$$\mathcal{B}_N(s_N) = 1.$$

The steps involved are summarized in algorithm 8.2.

Algorithm 8.2: Backward recursion for HMM (unstable version)

Given observations $w_{1:N}$ and parameters $\mathbf{\Pi}, \boldsymbol{\phi}$, the backward terms $\mathcal{B}_n(\sigma_m)$, for each time level n and each state σ_m, are computed as follows:

- At $n = N$ initialize by

$$\mathcal{B}_N(\sigma_m) = 1.$$

- For $n = N - 1 : 1$ compute recursively

$$\mathcal{B}_n(\sigma_m) = \sum_{m'=1}^{M} \mathcal{B}_{n+1}(\sigma_{m'}) G_{\phi_{\sigma_{m'}}}(w_{n+1})\, \pi_{\sigma_m \to \sigma_{m'}}.$$

Upon completion, the algorithm yields every $\mathcal{B}_n(\sigma_m)$ which may be tabulated as follows

$$\begin{array}{c} \\ t_1 \\ t_2 \\ \vdots \\ t_n \\ \vdots \\ t_N \end{array} \begin{array}{c} \begin{array}{cccc} \sigma_1 & \sigma_2 & \cdots & \sigma_M \end{array} \\ \begin{bmatrix} \mathcal{B}_1(\sigma_1) & \mathcal{B}_1(\sigma_2) & \cdots & \mathcal{B}_1(\sigma_M) \\ \mathcal{B}_2(\sigma_1) & \mathcal{B}_2(\sigma_2) & \cdots & \mathcal{B}_2(\sigma_M) \\ \vdots & \vdots & \ddots & \vdots \\ \mathcal{B}_n(\sigma_1) & \mathcal{B}_n(\sigma_2) & \cdots & \mathcal{B}_n(\sigma_M) \\ \vdots & \vdots & \ddots & \vdots \\ \mathcal{B}_N(\sigma_1) & \mathcal{B}_N(\sigma_2) & \cdots & \mathcal{B}_N(\sigma_M) \end{bmatrix} \end{array} \begin{array}{c} \\ \mathcal{B}_1(s_1) \\ \mathcal{B}_2(s_2) \\ \vdots \\ \mathcal{B}_n(s_n) \\ \vdots \\ \mathcal{B}_N(s_N) \end{array}$$

Joint decoding

The computation of $s^{\sharp}_{1:N}$, termed *joint decoding*, relies on the factorization

$$p(s_{1:N}|w_{1:N}, \boldsymbol{\rho}, \mathbf{\Pi}, \boldsymbol{\phi}) = p(s_N|w_{1:N}, \boldsymbol{\rho}, \mathbf{\Pi}, \boldsymbol{\phi}) \prod_{n=1}^{N-1} p(s_n|s_{n+1:N}, w_{1:N}, \boldsymbol{\rho}, \mathbf{\Pi}, \boldsymbol{\phi})$$

which implies that the maximizer $s^{\sharp}_{1:N}$ can be computed by the maximizers $s^{\sharp}_n$ of the individual factors $p(s_N|w_{1:N}, \boldsymbol{\rho}, \mathbf{\Pi}, \boldsymbol{\phi})$ and $p(s_n|s^{\sharp}_{n+1:N}, w_{1:N}, \boldsymbol{\rho}, \mathbf{\Pi}, \boldsymbol{\phi})$. Given $\mathcal{A}_n(s_n)$, maximization of each factor can be simplified through the relations

$$p(s_N|w_{1:N}, \boldsymbol{\rho}, \mathbf{\Pi}, \boldsymbol{\phi}) \propto \mathcal{A}_N(s_N), \tag{8.15}$$

$$p(s_n|s_{n+1:N}, w_{1:N}, \boldsymbol{\rho}, \mathbf{\Pi}, \boldsymbol{\phi}) \propto \mathcal{A}_n(s_n)\pi_{s_n \to s_{n+1}}, \qquad n = 1 : N - 1. \tag{!8.16}$$

The steps involved, known as *Viterbi recursion*, are summarized in algorithm 8.3.

Algorithm 8.3: Viterbi recursion for HMM

Given observations $w_{1:N}$, kinetic parameters $\mathbf{\Pi}$, and every $\mathcal{A}_n(\sigma_m)$, the Viterbi sequence $s^\sharp_{1:N}$ is computed as follows:

- At $n = N$ initialize by

$$s^\sharp_N = \underset{\sigma_m}{\operatorname{argmax}}\, \mathcal{A}_N(\sigma_m).$$

- For $n = N-1:1$ compute recursively

$$s^\sharp_n = \underset{\sigma_m}{\operatorname{argmax}}\, \mathcal{A}_n(\sigma_m)\pi_{\sigma_m \to s^\sharp_{n+1}}.$$

8.3.3 Estimation of the parameters

Estimation of a HMM seeks the maximizer of the likelihood

$$\{\boldsymbol{\rho}^*, \mathbf{\Pi}^*, \boldsymbol{\phi}^*\} = \underset{\boldsymbol{\rho},\mathbf{\Pi},\boldsymbol{\phi}}{\operatorname{argmax}}\, p\left(w_{1:N}|\boldsymbol{\rho},\mathbf{\Pi},\boldsymbol{\phi}\right).$$

Completing the likelihood $p\left(w_{1:N}|\boldsymbol{\rho},\mathbf{\Pi},\boldsymbol{\phi}\right)$ with the state sequence $s_{1:N}$, for instance, as

$$p\left(w_{1:N}|\boldsymbol{\rho},\mathbf{\Pi},\boldsymbol{\phi}\right) = \sum_{s_{1:N}} p\left(s_{1:N}, w_{1:N}|\boldsymbol{\rho},\mathbf{\Pi},\boldsymbol{\phi}\right)$$

we may perform this maximization with an EM procedure, where we iterate between an expectation (E) step and a maximization (M) step, similar to section 3.2.2. The entire procedure, adapted to the HMM, is known as the *Baum-Welch algorithm* and the steps involved are summarized in algorithm 8.4. In the next sections, we describe the steps of algorithm 8.4 in detail.

Algorithm 8.4: Baum-Welch algorithm

Given observations $w_{1:N}$ and an initial guess for the model parameters $\boldsymbol{\rho}, \boldsymbol{\Pi}, \boldsymbol{\phi}$, the Baum-Welch algorithm computes successively improved approximations of the maximizer of $p\left(w_{1:N}|\boldsymbol{\rho}, \boldsymbol{\Pi}, \boldsymbol{\phi}\right)$ by repeating the following steps

- E-step:
  - Use $\boldsymbol{\rho}, \boldsymbol{\Pi}, \boldsymbol{\phi}$ to compute $\mathcal{A}_n(\sigma_m)$ and $\mathcal{B}_n(\sigma_m)$.
  - Use $\mathcal{A}_n(\sigma_m)$ and $\mathcal{B}_n(\sigma_m)$ to compute

$$\zeta_n(\sigma_m) = \mathcal{A}_n(\sigma_m)\mathcal{B}_n(\sigma_m),$$
$$\eta_n(\sigma_m, \sigma_{m'}) = \mathcal{A}_{n-1}(\sigma_m)\mathcal{B}_n(\sigma_{m'})G\left(w_n; \phi_{\sigma_{m'}}\right)\pi_{\sigma_m \to \sigma_{m'}}.$$

  - Use $\eta_n(\sigma_m, \sigma_{m'})$ to compute

$$\xi_{\sigma_m}(\sigma_{m'}) = \sum_{n=2}^{N} \eta_n(\sigma_m, \sigma_{m'}).$$

- M-step:
  - Update $\boldsymbol{\rho}$ by replacing with

$$\left(\frac{\zeta_1(\sigma_1)}{\sum_{\sigma_m}\zeta_1(\sigma_m)}, \cdots, \frac{\zeta_1(\sigma_M)}{\sum_{\sigma_m}\zeta_1(\sigma_m)}\right).$$

  - Update $\boldsymbol{\Pi}$ by replacing each $\boldsymbol{\pi}_{\sigma_m}$ with

$$\left(\frac{\xi_{\sigma_m}(\sigma_1)}{\sum_{\sigma_{m'}}\xi_{\sigma_m}(\sigma_{m'})}, \cdots, \frac{\xi_{\sigma_m}(\sigma_M)}{\sum_{\sigma_{m'}}\xi_{\sigma_m}(\sigma_{m'})}\right).$$

  - Update $\boldsymbol{\phi}$ by replacing each ϕ_{σ_m} with the maximizer of

$$\sum_{n=1}^{N} \zeta_n(\sigma_m) \log G_{\phi_{\sigma_m}}\left(w_n\right).$$

The iterations are terminated either after a fixed number of repetitions or when the improvement between successive approximations of $\boldsymbol{\rho}, \boldsymbol{\Pi}, \boldsymbol{\phi}$ falls below a predetermined threshold.

Like any EM method, convergence of algorithm 8.4 to the *global* optimizer $\boldsymbol{\rho}^*, \boldsymbol{\Pi}^*, \boldsymbol{\phi}^*$ is *not* guaranteed. In practice, we need to try multiple initial guesses spanning a wide region of parameter space and, at the end, select the best optimizer found according to, say, the numerical value for the likelihood, $p\left(w_{1:N}|\boldsymbol{\rho}^*, \boldsymbol{\Pi}^*, \boldsymbol{\phi}^*\right)$. As algorithm 8.4 does not provide the value of $p\left(w_{1:N}|\boldsymbol{\rho}^*, \boldsymbol{\Pi}^*, \boldsymbol{\phi}^*\right)$ to compare the resulting maximizers, we need to compute it separately through, for example, eq. (8.7).

Expectation step*

In the E-step, we start from an initial approximation $\boldsymbol{\rho}^{\text{old}}, \boldsymbol{\Pi}^{\text{old}}, \boldsymbol{\phi}^{\text{old}}$ of the maximizer $\boldsymbol{\rho}^*, \boldsymbol{\Pi}^*, \boldsymbol{\phi}^*$ we wish to obtain and compute the expectation function that we will later maximize in the M-step. Specifically, we compute the expectation of

$$\log p\left(s_{1:N}, w_{1:N}|\boldsymbol{\rho}, \boldsymbol{\Pi}, \boldsymbol{\phi}\right) = \log \rho_{s_1} + \sum_{n=2}^{N} \log \pi_{s_{n-1} \to s_n} + \sum_{n=1}^{N} \log G_{\phi_{s_n}}\left(w_n\right) \qquad (!8.17)$$

*This is an advanced topic and could be skipped on a first reading.

with respect to the probability distribution of $s_{1:N}|w_{1:N}, \boldsymbol{\rho}^{\text{old}}, \boldsymbol{\Pi}^{\text{old}}, \boldsymbol{\phi}^{\text{old}}$. Since this expectation is a function of $\boldsymbol{\rho}, \boldsymbol{\Pi}, \boldsymbol{\phi}$ and also depends on $\boldsymbol{\rho}^{\text{old}}, \boldsymbol{\Pi}^{\text{old}}, \boldsymbol{\phi}^{\text{old}}$, we denote it with $Q_{\boldsymbol{\rho}^{\text{old}}, \boldsymbol{\Pi}^{\text{old}}, \boldsymbol{\phi}^{\text{old}}}(\boldsymbol{\rho}, \boldsymbol{\Pi}, \boldsymbol{\phi})$. In particular, this expectation is given by

$$\begin{aligned} Q_{\boldsymbol{\rho}^{\text{old}}, \boldsymbol{\Pi}^{\text{old}}, \boldsymbol{\phi}^{\text{old}}}(\boldsymbol{\rho}, \boldsymbol{\Pi}, \boldsymbol{\phi}) = & \sum_{s_1} p\left(s_1 | w_{1:N}, \boldsymbol{\rho}^{\text{old}}, \boldsymbol{\Pi}^{\text{old}}, \boldsymbol{\phi}^{\text{old}}\right) \log \rho_{s_1} \\ & + \sum_{s_{n-1}} \sum_{n=2}^{N} \sum_{s_n} p\left(s_{n-1}, s_n | w_{1:N}, \boldsymbol{\rho}^{\text{old}}, \boldsymbol{\Pi}^{\text{old}}, \boldsymbol{\phi}^{\text{old}}\right) \log \pi_{s_{n-1} \to s_n} \\ & + \sum_{s_n} \sum_{n=1}^{N} p\left(s_n | w_{1:N}, \boldsymbol{\rho}^{\text{old}}, \boldsymbol{\Pi}^{\text{old}}, \boldsymbol{\phi}^{\text{old}}\right) \log G_{\phi_{s_n}}(w_n). \end{aligned} \tag{!8.18}$$

The distributions over $s_n|w_{1:N}, \boldsymbol{\rho}^{\text{old}}, \boldsymbol{\Pi}^{\text{old}}, \boldsymbol{\phi}^{\text{old}}$ and $s_{n-1}, s_n|w_{1:N}, \boldsymbol{\rho}^{\text{old}}, \boldsymbol{\Pi}^{\text{old}}, \boldsymbol{\phi}^{\text{old}}$ can be computed in terms of the forward and backward terms of eqs. (8.8) and (8.13). As these are computed using on $\boldsymbol{\rho}^{\text{old}}, \boldsymbol{\Pi}^{\text{old}}, \boldsymbol{\phi}^{\text{old}}$, we use the associated $\mathcal{A}_n^{\text{old}}(s_n)$ and $\mathcal{B}_n^{\text{old}}(s_n)$. These distributions are

$$p\left(s_n | w_{1:N}, \boldsymbol{\rho}^{\text{old}}, \boldsymbol{\Pi}^{\text{old}}, \boldsymbol{\phi}^{\text{old}}\right) = \frac{\mathcal{A}_n^{\text{old}}(s_n) \mathcal{B}_n^{\text{old}}(s_n)}{p\left(w_{1:N} | \boldsymbol{\rho}^{\text{old}}, \boldsymbol{\Pi}^{\text{old}}, \boldsymbol{\phi}^{\text{old}}\right)}, \tag{!8.19}$$

$$p\left(s_{n-1}, s_n | w_{1:N}, \boldsymbol{\rho}^{\text{old}}, \boldsymbol{\Pi}^{\text{old}}, \boldsymbol{\phi}^{\text{old}}\right) = \frac{\mathcal{A}_{n-1}^{\text{old}}(s_{n-1}) \mathcal{B}_n^{\text{old}}(s_n) G_{\phi_{s_n}^{\text{old}}}(w_n) \, \pi_{s_{n-1} \to s_n}^{\text{old}}}{p\left(w_{1:N} | \boldsymbol{\rho}^{\text{old}}, \boldsymbol{\Pi}^{\text{old}}, \boldsymbol{\phi}^{\text{old}}\right)}. \tag{!8.20}$$

Finally, because $p\left(w_{1:N}|\boldsymbol{\rho}^{\text{old}}, \boldsymbol{\Pi}^{\text{old}}, \boldsymbol{\phi}^{\text{old}}\right)$ does not depend upon $\boldsymbol{\rho}, \boldsymbol{\Pi}, \boldsymbol{\phi}$, this term does not affect the maximization of $Q_{\boldsymbol{\rho}^{\text{old}}, \boldsymbol{\Pi}^{\text{old}}, \boldsymbol{\phi}^{\text{old}}}(\boldsymbol{\rho}, \boldsymbol{\Pi}, \boldsymbol{\phi})$ with respect to $\boldsymbol{\rho}, \boldsymbol{\Pi}, \boldsymbol{\phi}$. As such, we can safely drop it to obtain

$$\begin{aligned} Q_{\boldsymbol{\rho}^{\text{old}}, \boldsymbol{\Pi}^{\text{old}}, \boldsymbol{\phi}^{\text{old}}}(\boldsymbol{\rho}, \boldsymbol{\Pi}, \boldsymbol{\phi}) \propto & \sum_{m=1}^{M} \zeta_1^{\text{old}}(\sigma_m) \log \rho_{\sigma_m} \\ & + \sum_{m=1}^{M} \sum_{n=2}^{N} \sum_{m'=1}^{M} \eta_n^{\text{old}}(\sigma_m, \sigma_{m'}) \log \pi_{\sigma_m \to \sigma_{m'}} \\ & + \sum_{m=1}^{M} \sum_{n=1}^{N} \zeta_n^{\text{old}}(\sigma_m) \log G_{\phi_{\sigma_m}}(w_n) \end{aligned} \tag{8.21}$$

where

$$\zeta_n^{\text{old}}(\sigma_m) = \mathcal{A}_n^{\text{old}}(\sigma_m) \mathcal{B}_n^{\text{old}}(\sigma_m), \qquad n = 1:N \tag{8.22}$$

$$\eta_n^{\text{old}}(\sigma_m, \sigma_{m'}) = \mathcal{A}_{n-1}^{\text{old}}(\sigma_m) \mathcal{B}_n^{\text{old}}(\sigma_{m'}) G_{\phi_{\sigma_{m'}}^{\text{old}}}(w_n) \, \pi_{\sigma_m \to \sigma_{m'}}^{\text{old}}, \qquad n = 2:N. \tag{8.23}$$

Maximization step*

In the M-step, we obtain an improved approximation $\boldsymbol{\rho}^{\text{new}}, \boldsymbol{\Pi}^{\text{new}}, \boldsymbol{\phi}^{\text{new}}$ of the maximizer sought by maximizing the expectation function obtained in the E-step. Specifically, we maximize $Q_{\boldsymbol{\rho}^{\text{old}}, \boldsymbol{\Pi}^{\text{old}}, \boldsymbol{\phi}^{\text{old}}}(\boldsymbol{\rho}, \boldsymbol{\Pi}, \boldsymbol{\phi})$ under the constraints

$$\sum_{m=1}^{M} \rho_{\sigma_m} = 1, \qquad \sum_{m'=1}^{M} \pi_{\sigma_m \to \sigma'_m} = 1.$$

*This is an advanced topic and could be skipped on a first reading.

needed to ensure that $\boldsymbol{\rho}^{\text{new}}, \boldsymbol{\Pi}^{\text{new}}$ consists of valid probability vectors. As our objective function in eq. (8.21) is separable, the computation of the new maximizer can be broken down into separate maximizations

$$\begin{aligned}
\boldsymbol{\rho}^{\text{new}} &= \underset{\boldsymbol{\rho}}{\operatorname{argmax}} \sum_{m=1}^{M} \zeta_1^{\text{old}}(\sigma_m) \log \rho_{\sigma_m}, \\
\boldsymbol{\pi}_{\sigma_m}^{\text{new}} &= \underset{\boldsymbol{\pi}_{\sigma_m}}{\operatorname{argmax}} \sum_{n=2}^{N} \sum_{m'=1}^{M} \eta_n^{\text{old}}(\sigma_m, \sigma_{m'}) \log \pi_{\sigma_m \to \sigma_{m'}}, \\
\phi_{\sigma_m}^{\text{new}} &= \underset{\phi_{\sigma_m}}{\operatorname{argmax}} \sum_{n=1}^{N} \zeta_n^{\text{old}}(\sigma_m) \log G_{\phi_{\sigma_m}}(w_n).
\end{aligned}$$

Maximization for initial probabilities The first optimization entails one constraint, which we can solve by using a single Lagrange multiplier λ under the Lagrangian

$$\mathbb{L}(\lambda, \rho_{\sigma_1}, \ldots, \rho_{\sigma_M}) = \left(1 - \sum_{m=1}^{M} \rho_{\sigma_m}\right) \lambda + \sum_{m=1}^{M} \zeta_1^{\text{old}}(\sigma_m) \log \rho_{\sigma_m}.$$

Accordingly, the optimizer solves

$$\frac{\partial \mathbb{L}(\lambda, \rho_{\sigma_1}, \ldots, \rho_{\sigma_M})}{\partial \lambda} = 0, \qquad \frac{\partial \mathbb{L}(\lambda, \rho_{\sigma_1}, \ldots, \rho_{\sigma_M})}{\partial \rho_{\sigma_m}} = 0.$$

This system can be solved analytically. The solution, which provides the improved value $\boldsymbol{\rho}^{\text{new}}$ of the optimizer $\boldsymbol{\rho}^*$, is

$$\boldsymbol{\rho}^{\text{new}} = \left(\frac{\zeta_1^{\text{old}}(\sigma_1)}{\sum_{m=1}^{M} \zeta_1^{\text{old}}(\sigma_m)}, \cdots, \frac{\zeta_1^{\text{old}}(\sigma_M)}{\sum_{m=1}^{M} \zeta_1^{\text{old}}(\sigma_m)} \right). \tag{!8.24}$$

Maximization for transition probabilities For each m, the second optimization also entails one constraint, that we can solve using a single Lagrange multiplier κ_m under the Langrangian

$$\mathbb{K}_m(\kappa_m, \pi_{\sigma_m \to \sigma_1}, \ldots, \pi_{\sigma_m \to \sigma_M}) = \left(1 - \sum_{m'=1}^{M} \pi_{\sigma_m \to \sigma_{m'}}\right) \kappa_m + \sum_{n=2}^{N} \sum_{m'=1}^{M} \eta_n^{\text{old}}(\sigma_m, \sigma_{m'}) \log \pi_{\sigma_m \to \sigma_{m'}}.$$

Accordingly, the optimizer solves

$$\frac{\partial \mathbb{K}_m(\kappa_m, \pi_{\sigma_m \to \sigma_1}, \ldots, \pi_{\sigma_m \to \sigma_M})}{\partial \kappa_m} = 0, \qquad \frac{\partial \mathbb{K}_m(\kappa_m, \pi_{\sigma_m \to \sigma_1}, \ldots, \pi_{\sigma_m \to \sigma_M})}{\partial \pi_{\sigma_m \to \sigma_{m'}}} = 0.$$

Again, this system can be solved analytically. The solution, which provides the improved value $\boldsymbol{\pi}_{\sigma_m}^{\text{new}}$ of the optimizer $\boldsymbol{\pi}_{\sigma_m}^*$, is

$$\boldsymbol{\pi}_{\sigma_m}^{\text{new}} = \left(\frac{\xi_{\sigma_m}^{\text{old}}(\sigma_1)}{\sum_{m'=1}^{M} \xi_{\sigma_m}^{\text{old}}(\sigma_{m'})}, \cdots, \frac{\xi_{\sigma_m}^{\text{old}}(\sigma_M)}{\sum_{m'=1}^{M} \xi_{\sigma_m}^{\text{old}}(\sigma_{m'})} \right) \tag{!8.25}$$

where

$$\xi_{\sigma_m}^{\text{old}}(\sigma_{m'}) = \sum_{n=2}^{N} \eta_n^{\text{old}}(\sigma_m, \sigma_{m'}).$$

Maximization for emission parameters Unlike the first two optimizations, the third one generally cannot be solved analytically. Instead, depending on the functional form of the density $G_\phi(w)$, numerical techniques are needed to compute improved values $\phi_{\sigma_m}^{\text{new}}$ of the optimizers $\phi_{\sigma_m}^*$. In example 8.1 below, we illustrate a simpler case where numerical optimization is unnecessary.

Example 8.1: Estimation in a HMM with Normal observations

We consider a HMM with state-space $\sigma_{1:M}$ and Normal emissions

$$G_{\mu_{\sigma_m}, v_{\sigma_m}}(w) = \text{Normal}(w; \mu_{\sigma_m}, v_{\sigma_m})$$

where the state parameters are $\phi_{\sigma_m} = (\mu_{\sigma_m}, v_{\sigma_m})$. Further, we suppose that an approximation $\boldsymbol{\rho}^{\text{old}}, \boldsymbol{\Pi}^{\text{old}}, \boldsymbol{\phi}^{\text{old}}$ of the maximizer $\boldsymbol{\rho}^*, \boldsymbol{\Pi}^*, \boldsymbol{\phi}^*$ has already been computed and we seek an improved one $\boldsymbol{\rho}^{\text{new}}, \boldsymbol{\Pi}^{\text{new}}, \boldsymbol{\phi}^{\text{new}}$ using the Baum-Welch algorithm.

Due to the exponential form of $G_{\mu_{\sigma_m}, v_{\sigma_m}}(w)$, we can derive a maximization procedure for the emission parameters analytically. That is, for each σ_m, the improved emission parameters $\mu_{\sigma_m}^{\text{new}}, v_{\sigma_m}^{\text{new}}$ maximize

$$\sum_{n=1}^{N} \zeta_n^{\text{old}}(\sigma_m) \log G_{\mu_{\sigma_m}, v_{\sigma_m}}(w_n) \propto \sum_{n=1}^{N} \zeta_n^{\text{old}}(\sigma_m) \left(-\log v_{\sigma_m} - \frac{(w_n - \mu_{\sigma_m})^2}{v_{\sigma_m}} \right).$$

Since the maximizers are obtained by maximizing the above, they are found by solving

$$\frac{\partial}{\partial \mu_{\sigma_m}} \sum_{n=1}^{N} \zeta_n^{\text{old}}(\sigma_m) \left(-\log v_{\sigma_m} - \frac{(w_n - \mu_{\sigma_m})^2}{v_{\sigma_m}} \right) = 0,$$
$$\frac{\partial}{\partial v_{\sigma_m}} \sum_{n=1}^{N} \zeta_n^{\text{old}}(\sigma_m) \left(-\log v_{\sigma_m} - \frac{(w_n - \mu_{\sigma_m})^2}{v_{\sigma_m}} \right) = 0.$$

The solution is

$$\mu_{\sigma_m}^{\text{new}} = \frac{\sum_{n=1}^{N} \zeta_n^{\text{old}}(\sigma_m) w_n}{\sum_{n=1}^{N} \zeta_n^{\text{old}}(\sigma_m)}, \qquad v_{\sigma_m}^{\text{new}} = \frac{\sum_{n=1}^{N} \zeta_n^{\text{old}}(\sigma_m) (w_n - \mu_{\sigma_m}^{\text{new}})^2}{\sum_{n=1}^{N} \zeta_n^{\text{old}}(\sigma_m)}.$$

8.3.4 Some computational considerations

As we have seen already, the forward $\mathcal{A}_n(s_n)$ and backward $\mathcal{B}_n(s_n)$ variables are central to nearly every algorithm we have encountered so far and their accurate evaluation is essential in a HMM. Unfortunately, the computations in algorithms 8.1 and 8.2, which rely on the recursions of eqs. (!8.9) and (!8.14), involve a large number of multiplications between small numbers. Consequently, these algorithms are of limited practical value as most often they lead to rapid *underflow* and erroneous results.

Underflow is prevented if we consider *normalized* forward and backward terms

$$\hat{\mathcal{A}}_n(s_n) = \mathcal{A}_n(s_n) \frac{1}{p(w_{1:n}|\boldsymbol{\rho}, \boldsymbol{\Pi}, \boldsymbol{\phi})}, \tag{8.26}$$
$$\check{\mathcal{B}}_n(s_n) = \mathcal{B}_n(s_n) \frac{1}{p(w_{n+1:N}|w_{1:n}, \boldsymbol{\rho}, \boldsymbol{\Pi}, \boldsymbol{\phi})} \tag{8.27}$$

and perform the recursions for $\hat{\mathcal{A}}_n(s_n)$ and $\check{\mathcal{B}}_n(s_n)$ instead of $\mathcal{A}_n(s_n)$ and $\mathcal{B}_n(s_n)$. In these cases, the recursions needed rely on

$$\hat{\mathcal{A}}_1(s_1) = \frac{1}{\hat{\mathcal{C}}_1} G_{\phi_{s_1}}(w_1) \rho_{s_1},$$
$$\hat{\mathcal{A}}_n(s_n) = \frac{1}{\hat{\mathcal{C}}_n} G_{\phi_{s_n}}(w_n) \sum_{s_{n-1}} \pi_{s_{n-1} \to s_n} \hat{\mathcal{A}}_{n-1}(s_{n-1}), \qquad n = 2:N \tag{!8.28}$$
$$\check{\mathcal{B}}_n(s_n) = \frac{1}{\hat{\mathcal{C}}_{n+1}} \sum_{s_{n+1}} \check{\mathcal{B}}_{n+1}(s_{n+1}) G_{\phi_{s_{n+1}}}(w_{n+1}) \pi_{s_n \to s_{n+1}}, \qquad n = 1:N-1 \tag{!8.29}$$
$$\check{\mathcal{B}}_N(s_N) = 1$$

with the constants $\hat{\mathcal{C}}_n$ given by

$$\begin{aligned}\hat{\mathcal{C}}_1 &= p(w_1|\boldsymbol{\rho}, \boldsymbol{\phi}),\\ \hat{\mathcal{C}}_n &= p(w_n|w_{1:n-1}, \boldsymbol{\rho}, \boldsymbol{\Pi}, \boldsymbol{\phi}), \qquad\qquad n = 2:N.\end{aligned}$$

As the normalized terms $\hat{\mathcal{A}}_n(\sigma_1), \cdots, \hat{\mathcal{A}}_n(\sigma_M)$ are valid probabilities themselves, they are already scaled self-consistently and underflow is avoided. Further, because $\sum_{m=1}^{M} \hat{\mathcal{A}}_n(\sigma_m) = 1$, the constants $\hat{\mathcal{C}}_n$, can be easily computed during the forward recursion. The steps involved in both recursions are summarized in algorithms 8.5 and 8.6.

Algorithm 8.5: Forward recursion for HMM (stable version)

Given observations $w_{1:N}$ and parameters $\boldsymbol{\rho}, \boldsymbol{\Pi}, \boldsymbol{\phi}$ the forward terms $\hat{\mathcal{A}}_n(\sigma_m)$, for each time level n and each state σ_m, are computed as follows:

- At $n = 1$ initialize by

$$\begin{aligned}\hat{\mathcal{A}}'_1(\sigma_m) &= G_{\phi_{\sigma_m}}(w_1)\rho_{\sigma_m},\\ \hat{\mathcal{C}}_1 &= \sum_{m=1}^{M} \hat{\mathcal{A}}'_n(\sigma_m),\\ \hat{\mathcal{A}}_n(\sigma_m) &= \frac{1}{\hat{\mathcal{C}}_1}\hat{\mathcal{A}}'_n(\sigma_m).\end{aligned}$$

- For $n = 2, \ldots, N$ compute recursively

$$\begin{aligned}\hat{\mathcal{A}}'_n(\sigma_m) &= G_{\phi_{\sigma m}}(w_n) \sum_{m'=1}^{M} \pi_{\sigma_{m'} \to \sigma_m} \hat{\mathcal{A}}_{n-1}(\sigma_{m'}),\\ \hat{\mathcal{C}}_n &= \sum_{m=1}^{M} \hat{\mathcal{A}}'_n(\sigma_m),\\ \hat{\mathcal{A}}_n(\sigma_m) &= \frac{1}{\hat{\mathcal{C}}_n}\hat{\mathcal{A}}'_n(\sigma_m).\end{aligned}$$

Upon completion, the algorithm provides every $\hat{\mathcal{A}}_n(\sigma_m)$ and $\hat{\mathcal{C}}_n$ which may be tabulated as follows

$$\begin{array}{c} \\ t_1 \\ t_2 \\ \vdots \\ t_n \\ \vdots \\ t_N \end{array}\!\!\begin{array}{c} \begin{array}{cccc} \sigma_1 & \sigma_2 & \cdots & \sigma_M \end{array} \\ \begin{bmatrix} \hat{\mathcal{A}}_1(\sigma_1) & \hat{\mathcal{A}}_1(\sigma_2) & \cdots & \hat{\mathcal{A}}_1(\sigma_M) \\ \hat{\mathcal{A}}_2(\sigma_1) & \hat{\mathcal{A}}_2(\sigma_2) & \cdots & \hat{\mathcal{A}}_2(\sigma_M) \\ \vdots & \vdots & \ddots & \vdots \\ \hat{\mathcal{A}}_n(\sigma_1) & \hat{\mathcal{A}}_n(\sigma_2) & \cdots & \hat{\mathcal{A}}_n(\sigma_M) \\ \vdots & \vdots & \ddots & \vdots \\ \hat{\mathcal{A}}_N(\sigma_1) & \hat{\mathcal{A}}_N(\sigma_2) & \cdots & \hat{\mathcal{A}}_N(\sigma_M) \end{bmatrix} \end{array} \begin{bmatrix} \hat{\mathcal{C}}_1 \\ \hat{\mathcal{C}}_2 \\ \vdots \\ \hat{\mathcal{C}}_n \\ \vdots \\ \hat{\mathcal{C}}_N \end{bmatrix}.$$

Algorithm 8.6: Backward recursion for HMM (stable version)

Given observations $w_{1:N}$, parameters $\boldsymbol{\Pi}, \boldsymbol{\phi}$ and $\hat{\mathcal{C}}_{2:N}$, the backward terms $\mathcal{B}_n(\sigma_m)$, for each time level n and each state σ_m, are computed as follows:

- At $n = N$ initialize by

$$\check{\mathcal{B}}_N(\sigma_m) = 1.$$

- For $n = N-1, \ldots, 1$ compute recursively

$$\check{\mathcal{B}}_n(\sigma_m) = \frac{1}{\hat{\mathcal{C}}_{n+1}} \sum_{m'=1}^{M} \check{\mathcal{B}}_{n+1}(\sigma_{m'}) G_{\phi_{\sigma_{m'}}}(w_{n+1})\, \pi_{\sigma_m \to \sigma_{m'}}.$$

Upon completion, the algorithm provides every $\check{\mathcal{B}}_n(\sigma_m)$ which may be tabulated as follows

$$\begin{array}{c} \\ t_1 \\ t_2 \\ \vdots \\ t_n \\ \vdots \\ t_N \end{array} \begin{array}{c} \begin{array}{cccc} \sigma_1 & \sigma_2 & \cdots & \sigma_M \end{array} \\ \begin{bmatrix} \check{\mathcal{B}}_1(\sigma_1) & \check{\mathcal{B}}_1(\sigma_2) & \cdots & \check{\mathcal{B}}_1(\sigma_M)) \\ \check{\mathcal{B}}_2(\sigma_1) & \check{\mathcal{B}}_2(\sigma_2) & \cdots & \check{\mathcal{B}}_2(\sigma_M) \\ \vdots & \vdots & \ddots & \vdots \\ \check{\mathcal{B}}_n(\sigma_1) & \check{\mathcal{B}}_n(\sigma_2) & \cdots & \check{\mathcal{B}}_n(\sigma_M) \\ \vdots & \vdots & \ddots & \vdots \\ \check{\mathcal{B}}_N(\sigma_1) & \check{\mathcal{B}}_N(\sigma_2) & \cdots & \check{\mathcal{B}}_N(\sigma_M) \end{bmatrix} \end{array}.$$

As we can see, algorithms 8.5 and 8.6 involve more computations than algorithms 8.1 and 8.2. Nevertheless, this difference is almost negligible as the most expensive operation in both versions is a matrix-vector multiplication, appearing in eq. (!8.28) and eq. (!8.29), which scales with $M^2 N$. In any case, although less efficient than $\mathcal{A}_n(s_n)$ and $\mathcal{B}_n(s_n)$, computing $\hat{\mathcal{A}}_n(s_n)$ and $\check{\mathcal{B}}_n(s_n)$ avoids underflow which is an indispensable.

Furthermore, on account of eqs. (8.26) and (8.27), the normalized terms $\hat{\mathcal{A}}_n(s_n)$ and $\check{\mathcal{B}}_n(s_n)$ can be used almost anywhere both $\mathcal{A}_n(s_n)$ and $\mathcal{B}_n(s_n)$ are required. For example, both ways of decoding a HMM, *e.g.* eqs. (!8.11) and (!8.12) or eqs. (8.15) and (!8.16), are unaffected by the normalization. Similarly, the maximization of eq. (8.21) for estimating a HMM, *e.g.*, eqs. (8.22) and (8.23), remains similarly unaffected.

However, an important exception occurs when evaluating a HMM. In particular, because eq. (8.7) depends explicitly upon $\mathcal{A}_N(s_N)$, the normalization *does* have an effect and the marginal likelihood $p(w_{1:N}|\boldsymbol{\rho}, \boldsymbol{\Pi}, \boldsymbol{\phi})$ needs to be evaluated differently. The most convenient way is through the factorization

$$p(w_{1:N}|\boldsymbol{\rho}, \boldsymbol{\Pi}, \boldsymbol{\phi}) = \prod_{n=1}^{N} \hat{\mathcal{C}}_n \tag{!8.30}$$

with the constants $\hat{\mathcal{C}}_n$ computed, most efficiently, with the forward recursion of algorithm 8.5 and stored in logarithm form.

8.3.5 State-space labeling and likelihood*

The algorithms presented so far are routinely used to answer questions pertaining to a HMM. These algorithms exhibit maximum efficiency for their tasks. However, they are limited to yielding point estimates *only*. That is, at best, these algorithms provide a single choice for the values of the variables of interest, for example $s_n^*, s_{1:N}^\sharp$ or $\boldsymbol{\rho}^*, \boldsymbol{\Pi}^*, \boldsymbol{\phi}^*$. Unfortunately, they fail to quantify the uncertainty associated with each estimator, which is a serious limitation by itself.

*This is an advanced topic and could be skipped on a first reading.

Error bars around the estimators may be obtained with generic likelihood-based strategies, for example through Fisher information or bootstrapping techniques that we do not dwell upon here. Indeed, such approaches are possible, at least in theory, under Monte Carlo sampling or greedy computations where a portion of all possible sequences $s_{1:N}$ are computed. However, even with greedy computations, there is a fundamental degeneracy in likelihoods constructed from eqs. (8.4) to (8.6) prohibiting the uniqueness of any computed estimator.

Namely, similar to what we saw in section 7.1.3, HMM likelihoods, $p(w_{1:N}|\boldsymbol{\rho}, \boldsymbol{\Pi}, \boldsymbol{\phi})$ or $p(w_{1:N}, s_{1:N}|\boldsymbol{\rho}, \boldsymbol{\Pi}, \boldsymbol{\phi})$, are *invariant to permutations* of the constitutive state labels. That is, relabeling of the constitutive states results in the same value of the likelihood. Consequently, irrespective of how an estimator is obtained, there are always additional $M! - 1$ equally optimal ones leading to $M!$-fold degeneracy.

Example 8.2: State relabeling

To illustrate the likelihood's degeneracy, we consider a simplified HMM containing $N = 3$ time levels and $M = 2$ constitutive states. Further, for clarity, we adopt pedantic notation and let $s_{1:3}, \boldsymbol{\rho}, \boldsymbol{\Pi}, \boldsymbol{\phi}$ stand for the corresponding random variables. Considering realized values for these random variables, invariance of the (marginal) likelihood reads

$$\begin{aligned} &p\left(w_{1:3}\,\middle|\,\boldsymbol{\rho} = (\rho_\alpha, \rho_\beta), \boldsymbol{\Pi} = \begin{pmatrix} \pi_{\alpha\to\alpha} & \pi_{\alpha\to\beta} \\ \pi_{\beta\to\alpha} & \pi_{\beta\to\beta} \end{pmatrix}, \boldsymbol{\phi} = (\phi_\alpha, \phi_\beta)\right) \\ &= p\left(w_{1:3}\,\middle|\,\boldsymbol{\rho} = (\rho_\beta, \rho_\alpha), \boldsymbol{\Pi} = \begin{pmatrix} \pi_{\beta\to\beta} & \pi_{\beta\to\alpha} \\ \pi_{\alpha\to\beta} & \pi_{\alpha\to\alpha} \end{pmatrix}, \boldsymbol{\phi} = (\phi_\beta, \phi_\alpha)\right). \end{aligned}$$

Similarly, invariance of the (joint) likelihood reads

$$\begin{aligned} &p\left(w_{1:3}, s_{1:3} = (\alpha, \beta, \beta)\,\middle|\,\boldsymbol{\rho} = (\rho_\alpha, \rho_\beta), \boldsymbol{\Pi} = \begin{pmatrix} \pi_{\alpha\to\alpha} & \pi_{\alpha\to\beta} \\ \pi_{\beta\to\alpha} & \pi_{\beta\to\beta} \end{pmatrix}, \boldsymbol{\phi} = (\phi_\alpha, \phi_\beta)\right) \\ &= p\left(w_{1:3}, s_{1:3} = (\beta, \alpha, \alpha)\,\middle|\,\boldsymbol{\rho} = (\rho_\beta, \rho_\alpha), \boldsymbol{\Pi} = \begin{pmatrix} \pi_{\beta\to\beta} & \pi_{\beta\to\alpha} \\ \pi_{\alpha\to\beta} & \pi_{\alpha\to\alpha} \end{pmatrix}, \boldsymbol{\phi} = (\phi_\beta, \phi_\alpha)\right). \end{aligned}$$

Both cases are produced by considering every possible permutation of the constitutive states which, for this simple example, are

$$\begin{matrix} \sigma_1 & \sigma_2 \\ \begin{pmatrix} \alpha & \beta \\ \beta & \alpha \end{pmatrix}. \end{matrix}$$

8.4 The Hidden Markov Model in the Bayesian paradigm

A Bayesian formulation provides more modeling flexibility than its frequentist counterpart. Such flexibility is quite useful when modeling dynamical systems. For example, it provides a recipe by which to rigorously back-propagate measurement error into uncertainty over the parameters we seek or even characterize state-spaces in themselves as we will discover later in the context of HMMs within the Bayesian nonparametric paradigm.

In the Bayesian setting, we sample posteriors. Therefore, every question about a system formulated with a Bayesian HMM is answered through the posterior $p(\boldsymbol{\rho}, \boldsymbol{\Pi}, \boldsymbol{\phi}|w_{1:N})$ or the completed posterior $p(s_{1:N}, \boldsymbol{\rho}, \boldsymbol{\Pi}, \boldsymbol{\phi}|w_{1:N})$. The HMM of eqs. (8.4) to (8.6) provides probability distributions only for the passing states $p(s_{1:N}|\boldsymbol{\rho}, \boldsymbol{\Pi})$ and the measurements $p(w_{1:N}|s_{1:N}, \boldsymbol{\phi})$ which do not suffice in fully specifying our posterior. For this reason, a Bayesian HMM requires the specification of additional distributions that supply statistics to the parameters $\boldsymbol{\rho}, \boldsymbol{\Pi}, \boldsymbol{\phi}$. These distributions are our priors and, as anticipated, several reasonable choices can be devised to accommodate a system at hand. Below, we describe suitable choices and subsequently appropriate sampling techniques for a generic Bayesian HMM. We present more specialized versions, tailored to specific cases, in subsequent sections.

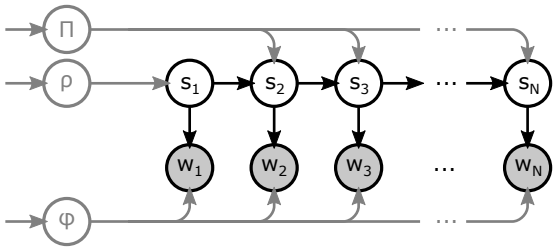

Figure 8.2: Graphical representation of a Bayesian HMM. Here, the parameters $\boldsymbol{\rho}, \boldsymbol{\Pi}$ and ϕ are assumed unknown.

8.4.1 Priors for the HMM

The simplest choice for the initial ρ_{σ_m} and transition $\pi_{\sigma_m \to \sigma_{m'}}$ probabilities are offered by independent priors on $\boldsymbol{\rho}$ and each $\boldsymbol{\pi}_{\sigma_m}$. For instance, draws from Dirichlet distributions

$$\begin{aligned} \boldsymbol{\rho} &\sim \text{Dirichlet}_{\sigma_{1:M}}(\eta \boldsymbol{\zeta}), \\ \boldsymbol{\pi}_{\sigma_m} &\sim \text{Dirichlet}_{\sigma_{1:M}}\left(\alpha_{\sigma_m} \boldsymbol{\beta}_{\sigma_m}\right), \end{aligned}$$

ensure valid probability arrays. In these priors, η and α_{σ_m} are positive scalar constants; while, $\boldsymbol{\zeta} = [\zeta_{\sigma_1}, \cdots, \zeta_{\sigma_M}]$ and $\boldsymbol{\beta}_{\sigma_m} = [\beta_{\sigma_m \to \sigma_1}, \cdots, \beta_{\sigma_m \to \sigma_1}]$ are probability arrays. Due to the conjugacy between the Categorical and Dirichlet distributions, as we see shortly, such prior choices are also computationally favored.

Despite the generality in the priors over the dynamical parameters, a choice for the emission parameters ϕ_{σ_m} depends heavily on the distribution $\mathbb{G}_\phi$ which, in turn, vary widely between systems. Computational tractability is facilitated when we consider iid priors

$$\phi_{\sigma_m} \sim \mathbb{H},$$

under a common, system specific, probability distribution $\mathbb{H}$. Additionally, we see below that the computations involved are greatly simplified if $\mathbb{H}$ is conjugate to $\mathbb{G}_\phi$.

8.4.2 MCMC inference in the Bayesian HMM

With the choices described above, an entire Bayesian HMM forward model is summarized as

$$\begin{aligned} \boldsymbol{\rho} &\sim \text{Dirichlet}_{\sigma_{1:M}}(\eta \boldsymbol{\zeta}), && && (8.31) \\ \boldsymbol{\pi}_{\sigma_m} &\sim \text{Dirichlet}_{\sigma_{1:M}}\left(\alpha_{\sigma_m} \boldsymbol{\beta}_{\sigma_m}\right), && && (8.32) \\ \phi_{\sigma_m} &\sim \mathbb{H}, && && (8.33) \\ s_1 | \boldsymbol{\rho} &\sim \text{Categorical}_{\sigma_{1:M}}(\boldsymbol{\rho}), && && (8.34) \\ s_n | s_{n-1}, \boldsymbol{\Pi} &\sim \text{Categorical}_{\sigma_{1:M}}(\boldsymbol{\pi}_{\sigma_m}), && n = 2:N && (8.35) \\ w_n | s_n, \boldsymbol{\phi} &\sim \mathbb{G}_{\phi_{s_n}}, && n = 1:N && (8.36) \end{aligned}$$

and illustrated in fig. 8.2. Inference on this HMM is more complicated than its non-Bayesian counterpart. Below, we describe two complementary MCMC sampling schemes. One is based on the Gibbs sampler, appropriate for routine applications, and another based on the Metropolis-Hastings sampler, appropriate for demanding applications where mixing of the Gibbs sampler becomes inefficient.

Gibbs sampling

A Gibbs sampling scheme is most efficient when it generates MCMC samples from the HMM's completed posterior $p(s_{1:N}, \boldsymbol{\rho}, \boldsymbol{\Pi}, \boldsymbol{\phi} | w_{1:N})$. In a basic implementation, we iterate between successive updates of $s_{1:N} | \boldsymbol{\rho}, \boldsymbol{\Pi}, \boldsymbol{\phi}, w_{1:N}$

and $\boldsymbol{\rho}, \boldsymbol{\Pi}, \boldsymbol{\phi} | s_{1:N}, w_{1:N}$. Due to the formulation of the HMM, the latter reduces to independent updates for each parameter. Specifically, once $s_{1:N} | \boldsymbol{\rho}, \boldsymbol{\Pi}, \boldsymbol{\phi}, w_{1:N}$ is sampled, we can update the parameters by sampling separately $\boldsymbol{\rho} | s_1$ and $\boldsymbol{\pi}_{\sigma_m} | s_{1:N}$ and $\phi_{\sigma_m} | s_{1:N}, w_{1:N}$ for each σ_m. The entire scheme is summarized in algorithm 8.7.

Algorithm 8.7: Gibbs sampling for Bayesian HMM

Given an initial sample $\boldsymbol{\rho}^{\text{old}}, \boldsymbol{\Pi}^{\text{old}}, \boldsymbol{\phi}^{\text{old}}$, which may be generated from the corresponding priors, MCMC updates are carried out by iterating the following steps:

- Update the state sequence by sampling $s_{1:N}^{\text{new}}$ with *forward filtering backward sampling* based on $\boldsymbol{\rho}^{\text{old}}, \boldsymbol{\Pi}^{\text{old}}, \boldsymbol{\phi}^{\text{old}}$.
- Compute state indices $\mathcal{N}_{\sigma_m}$ and counts $\boldsymbol{d}, \boldsymbol{C}$ based on $s_{1:N}^{\text{new}}$ as described shortly.
- Update the dynamic parameters by sampling from

$$\boldsymbol{\rho}^{\text{new}} \sim \text{Dirichlet}_M \left(\eta \boldsymbol{\zeta} + \boldsymbol{d}^{\text{new}} \right),$$
$$\boldsymbol{\pi}_{\sigma_m}^{\text{new}} \sim \text{Dirichlet}_M \left(\alpha_{\sigma_m} \boldsymbol{\beta}_{\sigma_m} + \boldsymbol{c}_{\sigma_m}^{\text{new}} \right).$$

- Update the emission parameters by sampling $\phi_{\sigma_m}^{\text{new}}$ for each σ_m based on $\mathcal{N}_{\sigma_m}^{\text{new}}$.

Below, we examine the steps involved in this Gibbs scheme in more detail. For clarity, we designate with $\boldsymbol{\rho}^{\text{old}}, \boldsymbol{\Pi}^{\text{old}}, \boldsymbol{\phi}^{\text{old}}$ a sample in the MCMC chain and with $s_{1:N}^{\text{new}}, \boldsymbol{\rho}^{\text{new}}, \boldsymbol{\Pi}^{\text{new}}, \boldsymbol{\phi}^{\text{new}}$ the very next sample.

Updates of the state sequence In the Gibbs sampler, the state sequence is updated by sampling from $p(s_{1:N} | \boldsymbol{\rho}^{\text{old}}, \boldsymbol{\Pi}^{\text{old}}, \boldsymbol{\phi}^{\text{old}}, w_{1:N})$. This distribution is factorized as

$$p(s_{1:N} | \boldsymbol{\rho}^{\text{old}}, \boldsymbol{\Pi}^{\text{old}}, \boldsymbol{\phi}^{\text{old}}, w_{1:N}) = p(s_N | \boldsymbol{\rho}^{\text{old}}, \boldsymbol{\Pi}^{\text{old}}, \boldsymbol{\phi}^{\text{old}}, w_{1:N}) \prod_{n=1}^{N-1} p(s_n | s_{n+1:N}, \boldsymbol{\rho}^{\text{old}}, \boldsymbol{\Pi}^{\text{old}}, \boldsymbol{\phi}^{\text{old}}, w_{1:N})$$

which allows s_N^{new} to be sampled first and subsequently each s_n^{new} to be sampled recursively backwards. We can perform such sampling using the forward terms $\hat{\mathcal{A}}_n(\sigma_m)$ which need to be precomputed through filtering, for example, via algorithm 8.5. As these terms need to be computed under $\boldsymbol{\rho}^{\text{old}}, \boldsymbol{\Pi}^{\text{old}}, \boldsymbol{\phi}^{\text{old}}$ we designate them with $\hat{\mathcal{A}}_n^{\text{old}}(\sigma_m)$. Once every $\hat{\mathcal{A}}_n^{\text{old}}(\sigma_m)$ is computed with a forward recursion, sampling begins with

$$s_N^{\text{new}} \sim \text{Categorical}_{\sigma_{1:M}} \left(\hat{\mathcal{A}}_N^{\text{old}}(\sigma_1), \cdots, \hat{\mathcal{A}}_N^{\text{old}}(\sigma_M) \right) \tag{8.37}$$

and recurses backward based on

$$s_n^{\text{new}} \sim \text{Categorical}_{\sigma_{1:M}} \left(\frac{\hat{\mathcal{A}}_n^{\text{old}}(\sigma_1) \pi_{\sigma_1 \to s_{n+1}^{\text{new}}}^{\text{old}}}{\sum_{m=1}^{M} \hat{\mathcal{A}}_n^{\text{old}}(\sigma_m) \pi_{\sigma_m \to s_{n+1}^{\text{new}}}^{\text{old}}}, \cdots, \frac{\hat{\mathcal{A}}_n^{\text{old}}(\sigma_M) \pi_{\sigma_M \to s_{n+1}^{\text{new}}}^{\text{old}}}{\sum_{m=1}^{M} \hat{\mathcal{A}}_n^{\text{old}}(\sigma_m) \pi_{\sigma_m \to s_{n+1}^{\text{new}}}^{\text{old}}} \right). \tag{!8.38}$$

The entire processes is termed *forward filtering backward sampling* and the steps involved are summarized in algorithm 8.8.

Algorithm 8.8: Forward filtering backward sampling

Given observations $w_{1:N}$ and parameters $\boldsymbol{\rho}, \boldsymbol{\Pi}, \boldsymbol{\phi}$, a state sequence $s_{1:N}$ is sampled as follows:

- Use algorithm 8.5 and $\boldsymbol{\rho}, \boldsymbol{\Pi}, \boldsymbol{\phi}$ to compute the forward terms $\hat{\mathcal{A}}_n(\sigma_m)$.
- At $n = N$ generate

$$s_N \sim \text{Categorical}_{\sigma_{1:M}}\left(\hat{\mathcal{A}}_N(\sigma_1), \cdots, \hat{\mathcal{A}}_N(\sigma_M)\right).$$

- For $n = N-1:1$ generate recursively

$$s_n \sim \text{Categorical}_{\sigma_{1:M}}\left(\frac{\hat{\mathcal{A}}_n(\sigma_1)\pi_{\sigma_1 \to s_{n+1}}}{\sum_{m=1}^M \hat{\mathcal{A}}_n(\sigma_m)\pi_{\sigma_m \to s_{n+1}}}, \cdots, \frac{\hat{\mathcal{A}}_n(\sigma_M)\pi_{\sigma_M \to s_{n+1}}}{\sum_{m=1}^M \hat{\mathcal{A}}_n(\sigma_m)\pi_{\sigma_m \to s_{n+1}}}\right).$$

Updates of the dynamic parameters In the Gibbs sampler, the initial probabilities are updated by sampling from $p(\boldsymbol{\rho}|s_1^{\text{new}})$. Due to conjugacy, this sampling reduces to

$$\boldsymbol{\rho}^{\text{new}} \sim \text{Dirichlet}_{\sigma_{1:M}}\left(\eta\boldsymbol{\zeta} + \boldsymbol{d}^{\text{new}}\right)$$

where $\boldsymbol{d}^{\text{new}} = [d_{\sigma_1}^{\text{new}}, \cdots, d_{\sigma_M}^{\text{new}}]$ is an array of zeroes and ones whose σ_m entry indicates whether $s_1^{\text{new}} = \sigma_m$.

Similarly, the transition probabilities out of each σ_m are updated by sampling from $p(\boldsymbol{\pi}_{\sigma_m}|s_{1:N}^{\text{new}})$. Again, due to conjugacy, this sampling reduces to

$$\boldsymbol{\pi}_{\sigma_m}^{\text{new}} \sim \text{Dirichlet}_{\sigma_{1:M}}\left(\alpha_{\sigma_m}\boldsymbol{\beta}_{\sigma_m} + \boldsymbol{c}_{\sigma_m}^{\text{new}}\right)$$

where $\boldsymbol{c}_{\sigma_m}^{\text{new}} = [c_{\sigma_m \to \sigma_1}^{\text{new}}, \cdots, c_{\sigma_m \to \sigma_M}^{\text{new}}]$ is a vector whose $\sigma_m \to \sigma_{m'}$ entry counts how many times the transition $\sigma_m \to \sigma_{m'}$ occurs in $s_{1:N}^{\text{new}}$.

Note 8.7: Transition count matrix

Bookkeeping in algorithm 8.7 is simpler, if we tabulate the count arrays $\boldsymbol{c}_{\sigma_m}$ into

$$\begin{array}{c} \\ \sigma_1 \\ \sigma_2 \\ \vdots \\ \sigma_M \end{array}\!\!\begin{array}{c} \begin{array}{cccc} \sigma_1 & \sigma_2 & \cdots & \sigma_M \end{array} \\ \begin{bmatrix} c_{\sigma_1\to\sigma_1} & c_{\sigma_1\to\sigma_2} & \cdots & c_{\sigma_1\to\sigma_M} \\ c_{\sigma_2\to\sigma_1} & c_{\sigma_2\to\sigma_2} & \cdots & c_{\sigma_2\to\sigma_M} \\ \vdots & \vdots & \ddots & \vdots \\ c_{\sigma_M\to\sigma_1} & c_{\sigma_M\to\sigma_2} & \cdots & c_{\sigma_M\to\sigma_M} \end{bmatrix} \end{array} = \begin{bmatrix} \boldsymbol{c}_{\sigma_1} \\ \boldsymbol{c}_{\sigma_2} \\ \vdots \\ \boldsymbol{c}_{\sigma_M} \end{bmatrix} = \boldsymbol{C}$$

similar to the tabulation of $\boldsymbol{\Pi}$.

Updates of the observation parameters In the Gibbs sampler, the emission parameters of σ_m are updated by sampling from $p(\phi_{\sigma_m}|s_{1:N}^{\text{new}}, w_{1:N})$. Using Bayes' rule, this distribution factorizes as

$$p(\phi_{\sigma_m}|s_{1:N}^{\text{new}}, w_{1:N}) \propto H(\phi_{\sigma_m}) \prod_{n \in \mathcal{N}_{\sigma_m}^{\text{new}}} G_{\phi_{\sigma_m}}(w_n) \tag{8.39}$$

where $\mathcal{N}_{\sigma_m}^{\text{new}}$ gathers the indices n of the time levels when $s_n^{\text{new}} = \sigma_m$. For arbitrary $\mathbb{H}$ and $\mathbb{G}_\phi$, sampling of $p(\phi_{\sigma_m}|s_{1:N}^{\text{new}}, w_{1:N})$ cannot be performed directly and a *within Gibbs* scheme is required. Nevertheless, as we show in example 8.3, distributions $\mathbb{G}_\phi$ with conjugate priors $\mathbb{H}$ are sampled directly.

Example 8.3: Bayesian HMM with Normal observations

We consider a HMM with state-space $\sigma_{1:M}$ and Normal emission densities

$$G_{\mu,\tau}(w) = \text{Normal}\left(w; \mu, \frac{1}{\tau}\right).$$

Gibbs sampling in this HMM is greatly simplified if we apply the conditionally conjugate prior

$$\mu \sim \text{Normal}\left(\xi, \frac{1}{\psi}\right), \qquad \tau \sim \text{Gamma}(\alpha, \beta).$$

With this choice of prior, a full update of the parameters $\mu_{\sigma_m}, \tau_{\sigma_m}$ is achieved by successively sampling $\mu_{\sigma_m}|\tau_{\sigma_m}^{\text{old}}, s_{1:N}^{\text{new}}, w_{1:N}$ and $\tau_{\sigma_m}^{\text{new}}|\mu_{\sigma_m}^{\text{new}}, s_{1:N}^{\text{new}}, w_{1:N}$. With these choices, the factorization of eq. (8.39) leads to

$$\mu_{\sigma_m}^{\text{new}} \sim \text{Normal}\left(\frac{\psi\xi + \tau^{\text{old}}\sum_{n\in\mathcal{N}_{\sigma_m}^{\text{new}}} w_n}{\psi + \tau^{\text{old}}|\mathcal{N}_{\sigma_m}^{\text{new}}|}, \frac{1}{\psi + \tau^{\text{old}}|\mathcal{N}_{\sigma_m}^{\text{new}}|}\right),$$

$$\tau_{\sigma_m}^{\text{new}} \sim \text{Gamma}\left(\alpha + \frac{1}{2}|\mathcal{N}_{\sigma_m}^{\text{new}}|, \frac{1}{\frac{1}{\beta} + \frac{1}{2}\sum_{n\in\mathcal{N}_{\sigma_m}^{\text{new}}}(w_n - \mu_{\sigma_m}^{\text{new}})^2}\right).$$

Updating each component μ_{σ_m} and τ_{σ_m} *once* per Gibbs iteration yields a valid sampler. However, mixing is better if these samplings are alternated *several* times per iteration. As inner iterations typically require considerably fewer computations than forward filtering backward sampling, for most HMM applications they add little to the sampler's overall computational cost while greatly improving its mixing.

Metropolis-Hastings sampling*

The Gibbs sampler described so far is most often sufficient for HMM applications whenever the total number of time levels with measurements, N, is low or the emission distributions, $\mathbb{G}_{\phi_{\sigma_m}}$, appreciably overlap. However, for long sequences and/or well separated emission distributions, mixing of the Gibbs sampler may become poor. For such cases, an alternative sampler, implemented in algorithm 8.9, that updates $\boldsymbol{\rho}, \boldsymbol{\Pi}, \boldsymbol{\phi}$ while keeping the state sequence $s_{1:N}$ marginalized, is preferable.

Algorithm 8.9: Metropolis-Hastings sampling for Bayesian HMM

Given an initial sample $\boldsymbol{\rho}^{\text{old}}, \boldsymbol{\Pi}^{\text{old}}, \boldsymbol{\phi}^{\text{old}}$, which may be generated from the corresponding priors, MCMC updates are carried out as follows.

- First, compute $\hat{\mathcal{C}}_{1:N}^{\text{old}}$ based on $\boldsymbol{\rho}^{\text{old}}, \boldsymbol{\Pi}^{\text{old}}, \boldsymbol{\phi}^{\text{old}}$ and set $\mathcal{L}^{\text{old}} = \sum_{n=1}^{N} \log \hat{\mathcal{C}}_n^{\text{old}}$.
- Then iterate the steps:
  - Generate proposals $\boldsymbol{\rho}^{\text{prop}}, \boldsymbol{\Pi}^{\text{prop}}, \boldsymbol{\phi}^{\text{prop}}$ based on $\boldsymbol{\rho}^{\text{old}}, \boldsymbol{\Pi}^{\text{old}}, \boldsymbol{\phi}^{\text{old}}$.
  - Compute $\hat{\mathcal{C}}_{1:N}^{\text{prop}}$ based on $\boldsymbol{\rho}^{\text{prop}}, \boldsymbol{\Pi}^{\text{prop}}, \boldsymbol{\phi}^{\text{prop}}$ and set $\mathcal{L}^{\text{prop}} = \sum_{n=1}^{N} \log \hat{\mathcal{C}}_n^{\text{prop}}$.
  - Perform the Metropolis-Hastings acceptance test based on $\{\mathcal{L}^{\text{old}}, \boldsymbol{\rho}^{\text{old}}, \boldsymbol{\Pi}^{\text{old}}, \boldsymbol{\phi}^{\text{old}}\}$ and $\{\mathcal{L}^{\text{prop}}, \boldsymbol{\rho}^{\text{prop}}, \boldsymbol{\Pi}^{\text{prop}}, \boldsymbol{\phi}^{\text{prop}}\}$.
    - If acceptance test *succeeds,* set $\boldsymbol{\rho}^{\text{new}} = \boldsymbol{\rho}^{\text{prop}}, \boldsymbol{\Pi}^{\text{new}} = \boldsymbol{\Pi}^{\text{prop}}, \boldsymbol{\phi}^{\text{new}} = \boldsymbol{\phi}^{\text{prop}}$ and $\mathcal{L}^{\text{new}} = \mathcal{L}^{\text{prop}}$.
    - If acceptance test *fails,* set $\boldsymbol{\rho}^{\text{new}} = \boldsymbol{\rho}^{\text{old}}, \boldsymbol{\Pi}^{\text{new}} = \boldsymbol{\Pi}^{\text{old}}, \boldsymbol{\phi}^{\text{new}} = \boldsymbol{\phi}^{\text{old}}$ and $\mathcal{L}^{\text{new}} = \mathcal{L}^{\text{old}}$.

Such a sampler may be developed based upon the same principles as the generic Metropolis-Hastings sampler of section 5.2.1. In particular, to sample from a HMM's posterior $p(\boldsymbol{\rho}, \boldsymbol{\Pi}, \boldsymbol{\phi}|w_{1:N})$, a Metropolis-Hastings sampler requires selecting a suitable proposal $Q_{\boldsymbol{\rho}^{\text{old}}, \boldsymbol{\Pi}^{\text{old}}, \boldsymbol{\phi}^{\text{old}}}(\boldsymbol{\rho}^{\text{prop}}, \boldsymbol{\Pi}^{\text{prop}}, \boldsymbol{\phi}^{\text{prop}})$. Although such a proposal may attempt

*This is an advanced topic and could be skipped on a first reading.

to update all parameters at once, in general, it is more practical to update one or at most few parameters at a time. This may be achieved by a mixture proposal, for example, of the form

$$\begin{aligned} Q_{\boldsymbol{\rho}^{\text{old}},\boldsymbol{\Pi}^{\text{old}},\boldsymbol{\phi}^{\text{old}}}\left(\boldsymbol{\rho}^{\text{prop}},\boldsymbol{\Pi}^{\text{prop}},\boldsymbol{\phi}^{\text{prop}}\right) = \omega\, Q_{\boldsymbol{\rho}^{\text{old}},\boldsymbol{\Pi}^{\text{old}}}\left(\boldsymbol{\rho}^{\text{prop}},\boldsymbol{\Pi}^{\text{prop}}\right)\delta_{\boldsymbol{\phi}^{\text{old}}}\left(\boldsymbol{\phi}^{\text{prop}}\right) \\ + (1-\omega)\, Q_{\boldsymbol{\phi}^{\text{old}}}\left(\boldsymbol{\phi}^{\text{prop}}\right)\delta_{\boldsymbol{\rho}^{\text{old}},\boldsymbol{\Pi}^{\text{old}}}\left(\boldsymbol{\rho}^{\text{prop}},\boldsymbol{\Pi}^{\text{prop}}\right) \end{aligned}$$

where, with probability ω, only proposals for the dynamical parameters are made; while, with probability $1-\omega$, only proposals for the observational parameters are made. In turn, each of the partial proposals $Q_{\boldsymbol{\rho}^{\text{old}},\boldsymbol{\Pi}^{\text{old}}}\left(\boldsymbol{\rho}^{\text{prop}},\boldsymbol{\Pi}^{\text{prop}}\right)$ and $Q_{\boldsymbol{\phi}^{\text{old}}}\left(\boldsymbol{\phi}^{\text{prop}}\right)$ may consist of further mixtures themselves that propose $\boldsymbol{\rho}^{\text{prop}}, \boldsymbol{\pi}^{\text{prop}}_{\sigma_m}, \phi^{\text{prop}}_{\sigma_m}$ separately.

Note 8.8: Choice of proposals

Generally, $Q_{\boldsymbol{\phi}^{\text{old}}}\left(\boldsymbol{\phi}^{\text{prop}}\right)$ is problem specific; however, for $Q_{\boldsymbol{\rho}^{\text{old}},\boldsymbol{\Pi}^{\text{old}}}\left(\boldsymbol{\rho}^{\text{prop}},\boldsymbol{\Pi}^{\text{prop}}\right)$ we may construct generic proposals by considering products of Dirichlet distributions. For example as

$$Q_{\boldsymbol{\rho}^{\text{old}},\boldsymbol{\Pi}^{\text{old}}}\left(\boldsymbol{\rho}^{\text{prop}},\boldsymbol{\Pi}^{\text{prop}}\right) = \text{Dirichlet}_{\sigma_{1:M}}\left(\boldsymbol{\rho}^{\text{prop}};\kappa\boldsymbol{\rho}^{\text{old}}\right)\prod_{m=1}^{M}\text{Dirichlet}_{\sigma_{1:M}}\left(\boldsymbol{\pi}^{\text{prop}}_{\sigma_m};\lambda\boldsymbol{\pi}^{\text{old}}_{\sigma_m}\right).$$

This choice ensures that the proposed $\boldsymbol{\rho}^{\text{prop}}, \boldsymbol{\Pi}^{\text{old}}$ consist of valid probability arrays and also allows for tuning of the resulting acceptance rate through the values of κ and λ.

Finally, once a proposal $\boldsymbol{\rho}^{\text{prop}}, \boldsymbol{\Pi}^{\text{prop}}, \boldsymbol{\phi}^{\text{prop}}$ is made, either through $Q_{\boldsymbol{\rho}^{\text{old}},\boldsymbol{\Pi}^{\text{old}}}\left(\boldsymbol{\rho}^{\text{prop}},\boldsymbol{\Pi}^{\text{prop}}\right)$ or $Q_{\boldsymbol{\phi}^{\text{old}}}\left(\boldsymbol{\phi}^{\text{prop}}\right)$, an acceptance ratio

$$\begin{aligned} A_{\boldsymbol{\rho}^{\text{old}},\boldsymbol{\Pi}^{\text{old}},\boldsymbol{\phi}^{\text{old}}}\left(\boldsymbol{\rho}^{\text{prop}},\boldsymbol{\Pi}^{\text{prop}},\boldsymbol{\phi}^{\text{prop}}\right) = \frac{p\left(\boldsymbol{\rho}^{\text{prop}},\boldsymbol{\Pi}^{\text{prop}},\boldsymbol{\phi}^{\text{prop}}|w_{1:N}\right)}{p\left(\boldsymbol{\rho}^{\text{old}},\boldsymbol{\Pi}^{\text{old}},\boldsymbol{\phi}^{\text{old}}|w_{1:N}\right)} \\ \times \frac{Q_{\boldsymbol{\rho}^{\text{prop}},\boldsymbol{\Pi}^{\text{prop}},\boldsymbol{\phi}^{\text{prop}}}\left(\boldsymbol{\rho}^{\text{old}},\boldsymbol{\Pi}^{\text{old}},\boldsymbol{\phi}^{\text{old}}\right)}{Q_{\boldsymbol{\rho}^{\text{old}},\boldsymbol{\Pi}^{\text{old}},\boldsymbol{\phi}^{\text{old}}}\left(\boldsymbol{\rho}^{\text{prop}},\boldsymbol{\Pi}^{\text{prop}},\boldsymbol{\phi}^{\text{prop}}\right)} \end{aligned}$$

is computed to complete the Metropolis-Hastings acceptance test. The second ratio depends on the specific choices for the proposals made and can be easily computed. The first ratio arises from the product of the ratio of priors as well as marginal likelihoods

$$\frac{p\left(\boldsymbol{\rho}^{\text{prop}},\boldsymbol{\Pi}^{\text{prop}},\boldsymbol{\phi}^{\text{prop}}|w_{1:N}\right)}{p\left(\boldsymbol{\rho}^{\text{old}},\boldsymbol{\Pi}^{\text{old}},\boldsymbol{\phi}^{\text{old}}|w_{1:N}\right)} = \underbrace{\frac{p\left(w_{1:N}|\boldsymbol{\rho}^{\text{prop}},\boldsymbol{\Pi}^{\text{prop}},\boldsymbol{\phi}^{\text{prop}}\right)}{p\left(w_{1:N}|\boldsymbol{\rho}^{\text{old}},\boldsymbol{\Pi}^{\text{old}},\boldsymbol{\phi}^{\text{old}}\right)}}_{\text{marginal likelihoods}}\underbrace{\frac{p\left(\boldsymbol{\rho}^{\text{prop}},\boldsymbol{\Pi}^{\text{prop}},\boldsymbol{\phi}^{\text{prop}}\right)}{p\left(\boldsymbol{\rho}^{\text{old}},\boldsymbol{\Pi}^{\text{old}},\boldsymbol{\phi}^{\text{old}}\right)}}_{\text{priors}}.$$

The last ratio depends exclusively on the priors and can also be easily computed. The other ratio is formed by the marginal likelihoods which we need to evaluate through eq. (!8.30). In particular, filtering such as algorithm 8.5 needs to be invoked twice: once for $p\left(w_{1:N}|\boldsymbol{\rho}^{\text{old}},\boldsymbol{\Pi}^{\text{old}},\boldsymbol{\phi}^{\text{old}}\right)$ and once for $p\left(w_{1:N}|\boldsymbol{\rho}^{\text{prop}},\boldsymbol{\Pi}^{\text{prop}},\boldsymbol{\phi}^{\text{prop}}\right)$. As filtering makes up the most computationally intensive part, both likelihoods can be retained and updated upon acceptance. By doing so, at the next iteration, we may avoid recomputing $p\left(w_{1:N}|\boldsymbol{\rho}^{\text{old}},\boldsymbol{\Pi}^{\text{old}},\boldsymbol{\phi}^{\text{old}}\right)$ thereby reducing the computational load from two to only *one* filtering operation per iteration which renders this scheme competitive with the earlier Gibbs scheme.

Note 8.9: Sampling the state sequence

If needed, algorithm 8.9 can also sample a state sequence to generate samples from the completed posterior $p(s_{1:N},\boldsymbol{\rho},\boldsymbol{\Pi},\boldsymbol{\phi}|w_{1:N})$ at little additional cost. For instance, following each filtering operation, if instead of only computing marginal likelihoods we also maintain and update every forward term $\hat{\mathcal{A}}_{1:N}(\sigma_m)$, a new state sequence

$s_{1:N}^{\text{new}}$ can be obtained at the end of each Metropolis-Hastings iteration by executing only the backward sampling stage of algorithm 8.8.

8.4.3 Interpretation and label switching*

Similar to the HMM's likelihoods we saw in section 8.3.5, the posteriors (both marginal or completed) of the Bayesian HMM in eqs. (8.31) to (8.36) are invariant to label permutations. In this case, the invariance can be seen in the factorization

$$p\left(\boldsymbol{\rho}, \boldsymbol{\Pi}, \boldsymbol{\phi} | w_{1:N}\right) \propto p\left(w_{1:N} | \boldsymbol{\rho}, \boldsymbol{\Pi}, \boldsymbol{\phi}\right) \\ \times \text{Dirichlet}_{\sigma_{1:M}}\left(\boldsymbol{\rho}; \eta \boldsymbol{\zeta}\right) \prod_{m=1}^{M} \text{Dirichlet}_{\sigma_{1:M}}\left(\boldsymbol{\pi}_{\sigma_m}; \alpha_{\sigma_m} \boldsymbol{\beta}_{\sigma_m}\right) \prod_{m=1}^{M} H(\phi_{\sigma_m})$$

and arises from the invariance of the likelihood as well as of the priors of section 8.4.1 with respect to label permutation.

Note 8.10: Breaking the posterior's invariance

Unlike the frequentist HMM, in a Bayesian HMM we may avoid the posterior's invariance if we assign *label specific* priors on the parameters. For instance, an alternative Bayesian HMM may be constructed with the following prior choices

$$\begin{aligned} \boldsymbol{\rho} &\sim \text{Dirichlet}_{\sigma_{1:M}}\left(\eta(\zeta_1, \cdots, \zeta_M)\right), \\ \boldsymbol{\pi}_{\sigma_m} &\sim \text{Dirichlet}_{\sigma_{1:M}}\left(\alpha_m(\beta_{m\to 1}, \cdots, \beta_{m\to M})\right), \\ \phi_{\sigma_m} &\sim \mathbb{H}_m. \end{aligned}$$

In this version, priors are label specific and, as a result, relabeling of the state-space leads to different posterior values to each one of the $M!$ samples produced by every label permutation. This way, we need not invoke *post hoc* heuristics to resolve identifiability problems.

However, it is better if label specific priors are avoided. This is because priors, informed by state labels, may hinder the mixing of the MCMC samplers applied. For best computational efficiency, it is preferable to use priors that are state, but not label, specific.

The posterior's invariance to label permutations leads to multimodal posteriors. For example, for any MAP estimate $\boldsymbol{\rho}^*, \boldsymbol{\Pi}^*, \boldsymbol{\phi}^*$, there are $M!$ total maximizers produced by the label permutations. Each one of these $M!$ maximizers, under vague priors, is a local mode of the posterior and is associated with a unique labeling. As eqs. (8.31) to (8.36) do not exhibit preference for a particular labeling of the state-space, in general, MCMC samplers produce samples that may use any of them. In fact, a sampler that performs well samples the entire posterior and thus, in the long run, switches between state labels producing samples from all $M!$ posterior modes.

As long as we are interested in deriving estimates that depend only on the constitutive states and not on the particular labeling chosen, the MCMC chains generated are sufficient. For example, if all we care about is quantifying the emission parameters attained at a particular level, we may focus on $p(\phi_{s_n}|w_{1:N})$ alone which, in itself, has only one permutation.

To derive label specific estimates, and therefore to allow for full interpretation of our estimates, we can impose *post hoc* identifiability constraints in terms of the state labels similar to the frequentist HMM of section 8.3.5. For instance, because all $M!$ modes are equally probable, for each MCMC sample computed we can consider all other $M! - 1$ permutations, by forming every possible permutation, and selecting the one that satisfies our constraints. Below we explain the steps involved in more detail.

Suppose that an MCMC sampler has already been employed, and for clarity, we denote with $\boldsymbol{\theta}_k^{(j)} = \{\boldsymbol{s}_k^{(j)}, \boldsymbol{\rho}_k^{(j)}, \boldsymbol{\Pi}_k^{(j)}, \boldsymbol{\phi}_k^{(j)}\}$ and $k = 1$ the values sampled at the j^{th} iteration. As we have $M!$ total permutations, we have $K = M!$ total

*This is an advanced topic and could be skipped on a first reading.

posterior samples which we index with $k = 1 : K$. We use $k > 1$ to denote every other sample value that can be formed by $\boldsymbol{\theta}_1^{(j)}$ through permutations of the state labels. As we mentioned above, due to label invariance, all such samples are equiprobable

$$p\left(\boldsymbol{\theta}_1^{(j)} | w_{1:N}\right) = p\left(\boldsymbol{\theta}_k^{(j)} | w_{1:N}\right).$$

To restore identifiability, it is sufficient to select a single $\boldsymbol{\theta}_k^{(j)}$ out of $\boldsymbol{\theta}_{1:K}^{(j)}$ satisfying our constraints. Since the permutation satisfying the constraints generally may differ from iteration to iteration, we designate it with $k^{(j)}$.

Perhaps the simplest way to impose identifiability relies on an ordering of the emission parameters, if one exists. For instance, provided ϕ_{σ_m} are real scalars, a labeling of the state-space $\sigma'_{1:M}$ may be selected such that it leads to a unique arrangement $\phi_{\sigma'_1} < \cdots < \phi_{\sigma'_M}$ (such as increasing mean signal level). In this simple case, $k^{(j)}$ may be easily identified and $\boldsymbol{\theta}_{k^{(j)}}^{(j)}$ readily found. This strategy, of course, is problem specific and very sensitive to the parameterization of the mother distribution $\mathbb{G}_\phi$ as well as to the imposed arrangement of the emission parameters ϕ. Further, it is unable to handle multivariate emission parameters or parameters that cannot be arranged in a sensible way. Below we describe an alternative strategy with higher computational cost and, although heuristic in nature, is less reliant on parametrizations.

For this strategy, we first need to select a reference point $\hat{\boldsymbol{\theta}}$ against which we can compare $\boldsymbol{\theta}_k^{(j)}$. Subsequently, for each j, out of $\boldsymbol{\theta}_{1:K}^{(j)}$, we select $\boldsymbol{\theta}_{k^{(j)}}^{(j)}$ that yields the best match. The reference $\hat{\boldsymbol{\theta}}$ can be either an *ad hoc* chosen point in the space of $\boldsymbol{s}, \boldsymbol{\rho}, \boldsymbol{\Pi}, \boldsymbol{\phi}$ or the MCMC sample with the highest posterior value. The latter can be readily found *post hoc* among the computed MCMC values $\boldsymbol{\theta}_1^{(j)}$.

Once an appropriate reference $\hat{\boldsymbol{\theta}}$ is selected, the comparison can be based on a dissimilarity function $\mathcal{D}\left(\boldsymbol{\theta}, \boldsymbol{\theta}'\right)$ that we also need to choose. For example, if $\mathcal{D}\left(\boldsymbol{\theta}, \boldsymbol{\theta}'\right)$ is based on the Euclidean distance, then selection from $\boldsymbol{\theta}_{1:K}^{(j)}$ results in finding the k belonging to the same semi-orthant with $\hat{\boldsymbol{\theta}}$. Of course, such k is unique.

Note 8.11: Dissimilarity function

A *dissimilarity function* $\mathcal{D}(\boldsymbol{\theta}, \boldsymbol{\theta}')$ associated to every pair $\boldsymbol{\theta}$ and $\boldsymbol{\theta}'$ yields a positive real scalar quantifying the dissimilarity between $\boldsymbol{\theta}$ and $\boldsymbol{\theta}'$. For example, for two identical samples $\boldsymbol{\theta} = \boldsymbol{\theta}'$, the dissimilarity must be zero; while, for different samples $\boldsymbol{\theta} \neq \boldsymbol{\theta}'$ the dissimilarity must be strictly positive. Solely for restoring identifiability, $\mathcal{D}(\boldsymbol{\theta}, \boldsymbol{\theta}')$ need not be symmetric. For instance, $\mathcal{D}(\boldsymbol{\theta}, \boldsymbol{\theta}')$ and $\mathcal{D}(\boldsymbol{\theta}', \boldsymbol{\theta})$ could attain different values.

A computationally convenient family of $\mathcal{D}(\boldsymbol{\theta}, \boldsymbol{\theta}')$ is offered by those additive over the dissimilarities of the individual state labels

$$\mathcal{D}(\boldsymbol{\theta}, \boldsymbol{\theta}') = \sum_{m=1}^{M} \mathcal{E}_m\left(\boldsymbol{\theta}, \boldsymbol{\theta}'\right)$$

where $\mathcal{E}_m\left(\boldsymbol{\theta}, \boldsymbol{\theta}'\right)$ is a dissimilarity function that compares *only* σ_m of $\boldsymbol{\theta}$ with σ'_m of $\boldsymbol{\theta}'$. In this case, finding the best $\boldsymbol{\theta}_k^{(j)}$ out of $\boldsymbol{\theta}_{1:K}^{(j)}$, reduces to a linear assignment problem, namely to finding the best association between the labeling $\sigma_{1:M}$ employed in $\boldsymbol{\theta}$ and the labeling $\boldsymbol{\sigma}'_{1:M}$ employed in $\boldsymbol{\theta}'$. As such, it can be solved efficiently through the *Hungarian algorithm* without explicitly forming each one of the K samples $\boldsymbol{\theta}_{1:K}^{(j)}$.

8.5 Dynamical variants of the Bayesian HMM*

As we mentioned earlier, the Bayesian HMM affords flexibility otherwise unavailable within the frequentist paradigm. For example, we may consider hierarchical formulations with hyperpriors on $\boldsymbol{\beta}_{\sigma_m}$ and, as we see in the next section, develop a HMM whose state-space $\sigma_{1:M}$ may grow arbitrarily in size. In doing so, such a formulation avoids the pitfalls of having to specify a particular size M to begin with, which is often a serious limitation when studying uncharacterized dynamical systems.

*This is an advanced topic and could be skipped on a first reading.

Before we turn to the study of uncharacterized systems, however, we focus on systems for which M is assumed known. For several such systems, properly tuning the prior on $\boldsymbol{\rho}$ and $\boldsymbol{\Pi}$ is sufficient in introducing flexibility in modeling dynamics. While scenarios we may consider are endless, we restrict ourselves to only to few key cases herein.

8.5.1 Modeling time scales

Earlier, in section 8.4.1, we spoke of priors on transitions probabilities. As we now show, these priors directly impact the induced prior on the escape time which, for some applications, constitutes a more natural quantity with which to work. Here we discuss how Bayesian methods provide us the ability to directly place priors on escape times and thus model timescales.

In particular, on account of the Markov assumption built into the dynamics of the state sequence $s_{1:N}$, once a system modeled by a HMM visits a constitutive state σ_m, it remains for a random number of *additional* steps D_{σ_m} before escaping and selecting another constitutive state. Specifically, in section 2.4.3, we derived the distribution

$$D_{\sigma_m}|\pi_{\sigma_m\to\sigma_m} \sim \text{Geometric}(1-\pi_{\sigma_m\to\sigma_m})$$

which exclusively depends upon the self-transition probability $\pi_{\sigma_m\to\sigma_m}$. Under the prior of section 8.4.1, we immediately obtain $\pi_{\sigma_m\to\sigma_m} \sim \text{Beta}\left(\alpha_{\sigma_m}\beta_{\sigma_m\to\sigma_m}, \alpha_{\sigma_m}(1-\beta_{\sigma_m\to\sigma_m})\right)$, which we may use to derive the induced prior on D_{σ_m}, namely

$$D_{\sigma_m} \sim \text{BetaNegBinomial}\left(1, \alpha_{\sigma_m}\beta_{\sigma_m\to\sigma_m}, \alpha_{\sigma_m}(1-\beta_{\sigma_m\to\sigma_m})\right).$$

This illustrates how the priors applied on an HMM's transition probabilities, in essence, also act as priors on induced timescales. For instance, since the mean of D_{σ_m} is

$$\langle D_{\sigma_m}\rangle = \frac{\alpha_{\sigma_m}\beta_{\sigma_m\to\sigma_m}}{\alpha_{\sigma_m}(1-\beta_{\sigma_m\to\sigma_m})-1}$$

we can tune the hyperparameters α_{σ_m} and $\beta_{\sigma_m\to\sigma_m}$ to influence priors on dwell periods selecting an *a priori* desired duration. For example, if a duration $\langle D_{\sigma_m}\rangle$ is specified, setting

$$\alpha_{\sigma_m} = \frac{\langle D_{\sigma_m}\rangle}{(1-\beta_{\sigma_m\to\sigma_m})\langle D_{\sigma_m}\rangle - \beta_{\sigma_m\to\sigma_m}}$$

provides a recipe for adjusting the values of α_{σ_m} that allows for state specific time scales.

Note 8.12: The sticky HMM

One way of influencing the *same* timescale across all constitutive states in a Bayesian HMM proceeds via setting every $\alpha_{\sigma_m} = \alpha$ equal and reparametrizating $\boldsymbol{\beta}_{\sigma_m}$ as

$$\boldsymbol{\beta}_{\sigma_m} = (1-c)\boldsymbol{B} + c\boldsymbol{D}_{\sigma_m}$$

where c is a scalar selected between 0 and 1; $\boldsymbol{B} = [B_{\sigma_1}, \cdots, B_{\sigma_M}]$ is a probability array; while, $\boldsymbol{D}_{\sigma_m} = [D_{\sigma_m\to\sigma_1}, \cdots, D_{\sigma_m\to\sigma_M}]$ is a probability array specific to each constitutive state σ_m. The latter can be used to separate self-transitions by setting $D_{\sigma_m\to\sigma_m} = 1$ and $D_{\sigma_m\to\sigma_{m'}} = 0$. The resulting prior on $\boldsymbol{\Pi}$ now takes the form

$$\boldsymbol{\pi}_{\sigma_m} \sim \text{Dirichlet}_{\sigma_{1:M}}\left(\alpha(1-c)\boldsymbol{B} + \alpha c\boldsymbol{D}_{\sigma_m}\right).$$

With this prior, self-transitions over the entire state-space are reinforced with only a limited number of hyperparameters $\alpha, c, \boldsymbol{B}$. Due to its ability to reinforce self-transitions and long dwells, this prior is termed *sticky*. Under the sticky prior, the induced dwell durations are

$$\langle D_{\sigma_m}\rangle = \frac{c+(1-c)B_{\sigma_m}}{(1-c)\left(1-B_{\sigma_m}\right)-\frac{1}{\alpha}}$$

which become uniform over $\sigma_{1:M}$ by setting $B_{\sigma_m} = 1/M$.

8.5.2 Modeling equilibrium

Provided every $\beta_{\sigma_m \to \sigma_{m'}}$ is non-zero, the prior on $\mathbf{\Pi}$ ensures that the transition probabilities $\pi_{\sigma_m \to \sigma_{m'}}$ in a Bayesian HMM are strictly positive. This, in turn, ensures that transitions between any pair of constitutive states are possible in all resulting $s_{1:N}$. Therefore, a system modeled by such a HMM is ergodic, *i.e.*, may explore the entire state-space. Such systems, if allowed to evolve for sufficiently long time, may reach equilibrium.

For a dynamical system at equilibrium, initialization, and kinetics are interrelated. In particular, ρ_{σ_m} and $\pi_{\sigma_m \to \sigma_{m'}}$ satisfy the balance condition

$$\rho_{\sigma_m} = \sum_{m'=1}^{M} \rho_{\sigma_{m'}} \pi_{\sigma_{m'} \to \sigma_m}.$$

For a dynamical system at equilibrium, we can use this condition to express $\boldsymbol{\rho}$ in terms of $\mathbf{\Pi}$, suggesting that at equilibrium $\boldsymbol{\rho}$ is a dependent parameter. Accordingly, to model a system at equilibrium, we need to place priors only on $\mathbf{\Pi}$. For example

$$\begin{aligned}
\boldsymbol{\pi}_{\sigma_m} &\sim \text{Dirichlet}_{\sigma_{1:M}}\left(\alpha_{\sigma_m}\boldsymbol{\beta}_{\sigma_m}\right), \\
\phi_{\sigma_m} &\sim \mathbb{H}, \\
s_1|\mathbf{\Pi} &\sim \text{Categorical}_{\boldsymbol{\sigma}_{1:M}}\left(\boldsymbol{\rho}_{\mathbf{\Pi}}\right), \\
s_n|s_{n-1}, \mathbf{\Pi} &\sim \text{Categorical}_{\boldsymbol{\sigma}_{1:M}}\left(\boldsymbol{\pi}_{\sigma_m}\right), && n = 2:N \\
w_n|s_n, \boldsymbol{\phi} &\sim \mathbb{G}_{\phi_{s_n}}, && n = 1:N
\end{aligned}$$

in which the initial probability array $\boldsymbol{\rho}_{\mathbf{\Pi}}$ is now dictated by the balance condition.

Although the prior on $\mathbf{\Pi}$ is the same as seen before, due to its implicit effect on $\boldsymbol{\rho}_{\mathbf{\Pi}}$, it is no longer conjugate to $s_{1:N}|\mathbf{\Pi}$. Consequently, we cannot use the Gibbs sampler of algorithm 8.7 to obtain MCMC samples $\boldsymbol{\pi}_{\sigma_m}|s_{1:N}$ and inference is only possible by means of a Metropolis-Hastings sampler such as an appropriately adjusted algorithm 8.9.

8.5.3 Modeling reversible systems

The prior on $\mathbf{\Pi}$ we just discussed enforces equilibrium on the HMM which is somewhat stronger than simply ensuring reversibility of the kinetics irrespective of equilibrium being reached by the time of the first measurement. To model a reversible dynamical system, that may not necessarily have reached equilibrium before the measurement's onset, we need to consider independent priors on $\boldsymbol{\rho}$ and $\mathbf{\Pi}$. In such case, $\boldsymbol{\rho} \sim \text{Dirichlet}_{\sigma_{1:M}}(\eta\boldsymbol{\zeta})$ remains an appropriate choice; however, ensuring reversible kinetics requires fundamentally different choices for $\mathbf{\Pi}$.

Note 8.13: A reversible HMM

One way to ensure a reversible $\mathbf{\Pi}$ is to reparametrize the transition probabilities as

$$\pi_{\sigma_m \to \sigma_{m'}} = \frac{\lambda_{\sigma_m \leftrightarrow \sigma_{m'}}}{\sum_{m''=1}^{M} \lambda_{\sigma_m \leftrightarrow \sigma_{m''}}}.$$

Reversibility is ensured by requiring that the new parameters be pairwise symmetric

$$\lambda_{\sigma_m \leftrightarrow \sigma_{m'}} = \lambda_{\sigma_{m'} \leftrightarrow \sigma_m}.$$

On account of symmetry, in the new parametrization, we need only $M(M+1)/2$ priors that we may select independently. For instance

$$\lambda_{\sigma_m \leftrightarrow \sigma_{m'}} \sim \text{Gamma}\left(fE_{\sigma_m}E_{\sigma_{m'}}, 1\right)$$

where f and $E_{\sigma_1}, \cdots, E_{\sigma_M}$ are hyperparameters controlling how tightly each constitutive state couples to the others.

The kinetic scheme induced by the symmetric prior of $\lambda_{\sigma_m \leftrightarrow \sigma_{m'}}$ leads to the following reparametrized equilibrium distribution

$$\boldsymbol{\rho}_* = \left[\frac{\sum_{m=1}^{M} \lambda_{\sigma_1 \leftrightarrow \sigma_m}}{\sum_{m=1}^{M} \sum_{m'=1}^{M} \lambda_{\sigma_{m'} \leftrightarrow \sigma_m}}, \cdots, \frac{\sum_{m=1}^{M} \lambda_{\sigma_M \leftrightarrow \sigma_m}}{\sum_{m=1}^{M} \sum_{m'=1}^{M} \lambda_{\sigma_{m'} \leftrightarrow \sigma_m}} \right].$$

As a result, whenever equilibrium needs to be imposed as a stronger condition to reversibility, we may proceed by setting the initial probabilities $\boldsymbol{\rho}$ equal to $\boldsymbol{\rho}_*$.

8.5.4 Modeling kinetic schemes

Unlike the ergodic HMM where the system may evolve to and from any constitutive state, some physical scenarios require that some transitions be prohibited. For example, modeling irreversible chemical reactions such as *photo-bleaching* where molecules undergo a chemical change rendering them unable to fluoresce at a designated wavelength.

From the modeling perspective, we can take advantage of the flexibility allowed by the hyperparameters $\beta_{\sigma_m \to \sigma_{m'}}$ to model kinetic schemes. Under the prior of section 8.4.1, a transition probability $\pi_{\sigma_m \to \sigma_{m'}}$ is zero only when the corresponding $\beta_{\sigma_m \to \sigma_{m'}}$ is zero. Essentially, to ensure that the system modeled cannot undergo some transitions, or undergoes other transitions into a certain order, we need to properly set the sparsity pattern of

$$\begin{array}{c} \\ \sigma_1 \\ \sigma_2 \\ \vdots \\ \sigma_M \end{array} \begin{array}{c} \begin{array}{cccc} \sigma_1 & \sigma_2 & \cdots & \sigma_M \end{array} \\ \begin{bmatrix} \beta_{\sigma_1 \to \sigma_1} & \beta_{\sigma_1 \to \sigma_2} & \cdots & \beta_{\sigma_1 \to \sigma_M} \\ \beta_{\sigma_2 \to \sigma_1} & \beta_{\sigma_2 \to \sigma_2} & \cdots & \beta_{\sigma_2 \to \sigma_M} \\ \vdots & \vdots & \ddots & \vdots \\ \beta_{\sigma_M \to \sigma_1} & \beta_{\sigma_M \to \sigma_2} & \cdots & \beta_{\sigma_M \to \sigma_M} \end{bmatrix} \end{array} = \begin{bmatrix} \boldsymbol{\beta}_{\sigma_1} \\ \boldsymbol{\beta}_{\sigma_2} \\ \vdots \\ \boldsymbol{\beta}_{\sigma_M} \end{bmatrix}.$$

Example 8.4: A left-to-right HMM

To model a system, such as an idealized molecular motor with no reverse stepping, where returning to previous constitutive states are *prohibited*, we may use a left-to-right structure of the form

$$\begin{array}{c} \\ \sigma_1 \\ \sigma_2 \\ \sigma_3 \\ \sigma_4 \\ \sigma_5 \end{array} \begin{array}{c} \begin{array}{ccccc} \sigma_1 & \sigma_2 & \sigma_3 & \sigma_4 & \sigma_5 \end{array} \\ \begin{bmatrix} 1/5 & 1/5 & 1/5 & 1/5 & 1/5 \\ 0 & 1/4 & 1/4 & 1/4 & 1/4 \\ 0 & 0 & 1/3 & 1/3 & 1/3 \\ 0 & 0 & 0 & 1/2 & 1/2 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \end{array} = \begin{bmatrix} \boldsymbol{\beta}_{\sigma_1} \\ \boldsymbol{\beta}_{\sigma_2} \\ \boldsymbol{\beta}_{\sigma_3} \\ \boldsymbol{\beta}_{\sigma_4} \\ \boldsymbol{\beta}_{\sigma_5} \end{bmatrix}$$

where, for simplicity, we have chosen a state-space of size $M = 5$. Observed for a sufficiently long period, $N \gg 1$, a system modeled as such eventually reaches σ_5. Since the prior imposed on $\boldsymbol{\pi}_{\sigma_5}$ is deterministic, allowing only for $\boldsymbol{\pi}_{\sigma_5} = [0, 0, 0, 0, 1]$, this model is then equivalent to modeling absorbing dynamics at the boundary.

8.5.5 Modeling factorial dynamics

Occasionally the underlying system of interest consists of multiple components that evolve *independently*. Although in such systems each component follows its own dynamics, it may be possible that the entire system is assessed only through a common observation. That is, all components may give rise a single collective observation.

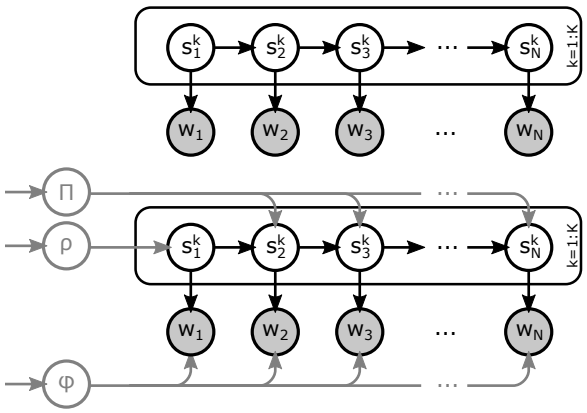

Figure 8.3: Graphical representation of a factorial HMM. The top panel shows the basic structure; while, the lower panel indicated also dependencies on the parameters.

Example 8.5: Photo-blinking

Imagine that we observe a specimen consisting of K fluorescent probes undergoing photo-blinking. That is, each probe switches between a bright and a dark states, which we may model by σ_1 and σ_2, respectively. Under idealized conditions, *i.e.*, when probes are faraway from each other, we assume that each probe switches between σ_1 and σ_2 independently. We can readily model such a scenario with

$$s_n^k | s_{n-1}^k, \mathbf{\Pi} \sim \text{Categorical}_{\sigma_{1:2}}(\boldsymbol{\pi}_{s_n^k}).$$

Here, s_n^k is the state, termed *photo-state*, of the k^{th} probe at the time of the n^{th} assessment and $\mathbf{\Pi}$ gathers the transition probabilities $\boldsymbol{\pi}_{\sigma_1}$ and $\boldsymbol{\pi}_{\sigma_2}$.

When a probe is bright, it emits photons with a rate $\mu_{\sigma_1} > 0$. However, when the probe is dark, it emits no photons, which we model with a rate $\mu_{\sigma_2} = 0$. Since photon emissions from all probes are additive, in total, our specimen emits photons with a rate that combines contributions from all probes. As such, the photon emission rate driving the n^{th} assessment is $\sum_{k=1}^{K} \mu_{s_n^k}$. Considering a detector with exposure time τ_{exp}, this leads to an emission distribution

$$w_n | s_n^{1:K} \sim \text{Poisson}\left(\tau_{\text{exp}} \sum_{k=1}^{K} \mu_{s_n^k}\right),$$

where w_n denotes the net amount of photon detections at the n^{th} time level.

Naturally, such a system may be formulated by a generalization of the HMM as follows

$$\begin{aligned}
s_1^k | \boldsymbol{\rho} &\sim \text{Categorical}_{\sigma_{1:M}}(\boldsymbol{\rho}), && && k = 1:K \\
s_n^k | s_{n-1}^k, \mathbf{\Pi} &\sim \text{Categorical}_{\sigma_{1:M}}(\boldsymbol{\pi}_{s_n^k}), && n = 2:N, && k = 1:K \\
w_n | s_n^{1:K}, \boldsymbol{\phi} &\sim \mathbb{G}_{\sum_{k=1}^{K} \phi_{s_n^k}}, && n = 1:N.
\end{aligned}$$

In this model, each component is modeled by its own state s_n^k. However, at each time level, observations are coupled by a common emission distribution $\mathbb{G}_{\sum_{k=1}^{K} \phi_{s_n^k}}$. This formulation is termed *factorial hidden Markov model* and it is depicted graphically in fig. 8.3. Inference in this system relies on the posterior $p\left(s_{1:N}^{1:K}, \boldsymbol{\rho}, \mathbf{\Pi}, \boldsymbol{\phi} | w_{1:N}\right)$ and follows the same filtering and smoothing algorithms seen earlier.

8.6 The infinite Hidden Markov Model*

In the previous section, we saw how a Bayesian HMM constructed around a fixed state-space $\sigma_{1:M}$ can identify the characteristics of each constitutive state σ_m, for example dynamical and observational parameters represented by $\boldsymbol{\rho}, \boldsymbol{\Pi}$ and $\boldsymbol{\phi}$, respectively. These characteristics are captured in the posterior $p(\boldsymbol{\rho}, \boldsymbol{\Pi}, \boldsymbol{\phi}|w_{1:N})$ which, for the models presented so far, inevitably depends upon the size, M, of the state-space employed.

In practice, we often need to study dynamical systems whose state-space is uncharacterized. In this case, our knowledge of the system at hand does not allow us to specify a unique M. Indeed, despite the generality and elegance of our formulations, the dependence of our posteriors upon M is a limiting factor. Luckily, extensions of the Bayesian formulation are possible resulting in posterior distributions independent of M which may remain unspecified or arbitrarily large.

In particular, by building upon the Bayesian HMM and using appropriate hyperpriors, described shortly, we may develop a HMM version whose state-space is infinite. Such a formulation remains valid and may be applied even when our primary goal is to identify the characteristics of the constitutive states visited by the system while the total number of available states is unknown.

Note 8.14: Dynamics on infinite state-spaces

With an infinite state-space, our system has access to infinite constitutive states. Specifically, each time the system departs from a passing state s_n it may escape to infinitely many σ_m. Provided that the system has already visited only a finite number of them, this means that, at every transition, the systems can always explore new states that will be visited for the first time. In principle, such a system may be allowed to visit an unvisited state every time it transitions. Although such scenarios may arise, for example such as birth processes of example 2.4, most often we are interested in studying systems that frequently or sporadically *revisit* states. For the latter systems, the number of constitutive states visited during the time course of our measurements, which is finite, is drastically lower that the total number of observations.

As we mentioned, the posterior $p(\boldsymbol{\rho}, \boldsymbol{\Pi}, \boldsymbol{\phi}|w_{1:N})$ of the model in eqs. (8.31) to (8.36) depends upon M. Such dependency signifies that with a different number of constitutive states available, different choices of kinetic $\boldsymbol{\rho}, \boldsymbol{\Pi}$ and emission $\boldsymbol{\phi}$ parameters are assigned under the measurements $w_{1:N}$. To eliminate such dependence on M, we need to be able to reinforce state revisiting in an infinite state model. This can be achieved by properly selecting the priors on the initial and transition probabilities $\boldsymbol{\rho}$ and $\boldsymbol{\Pi}$.

One way to do so is to consider placing a common prior among all constitutive states. In this case, setting η and all α_{σ_m} equal. For simplicity, we denote the latter with α. Also, we may set all elements of the arrays $\boldsymbol{\zeta}$ and $\boldsymbol{\beta}_{\sigma_m}$ equal and denote them with $\boldsymbol{\beta} = [\beta_{\sigma_1}, \cdots, \beta_{\sigma_M}]$. Under this common prior, constitutive states with high β_{σ_m} generally receive more transitions into them than constitutive states with low β_{σ_m}.

Of course, for an uncharacterized system, we cannot identify beforehand how often the constitutive states are visited or even which of them are visited more often than others. Thus, in principle, the prior $\boldsymbol{\beta}$ is unknown too and we need to estimate it in parallel with other quantities of interest. For this reason, we place a hyperprior on $\boldsymbol{\beta}$ and, as $\boldsymbol{\beta}$ is a probability array, the most natural choice for it is also a Dirichlet distribution. This leads to the following hierarchical Dirichlet formulation

$$\boldsymbol{\beta} \sim \text{Dirichlet}_{\sigma_{1:M}}(\gamma\boldsymbol{\xi}), \tag{8.40}$$

$$\boldsymbol{\rho}|\boldsymbol{\beta} \sim \text{Dirichlet}_{\sigma_{1:M}}(\alpha\boldsymbol{\beta}), \tag{8.41}$$

$$\boldsymbol{\pi}_{\sigma_m}|\boldsymbol{\beta} \sim \text{Dirichlet}_{\sigma_{1:M}}(\alpha\boldsymbol{\beta}), \tag{8.42}$$

where γ is a positive scalar and $\boldsymbol{\xi}$ a probability array. As our system is uncharacterized, at this stage, as we cannot distinguish among the constitutive states, we need to ensure symmetry of $\boldsymbol{\beta}$, which we may achieve through

$$\boldsymbol{\xi} = \left[\frac{1}{M}, \cdots, \frac{1}{M}\right].$$

*This is an advanced topic and could be skipped on a first reading.

As anticipated, the hierarchical prior of eqs. (8.40) and (8.41), when combined with the HMM's kinetics and emissions

$$\begin{aligned} \phi_{\sigma_m} &\sim \mathbb{H}, \\ s_1|\boldsymbol{\rho} &\sim \text{Categorical}_{\boldsymbol{\sigma}_{1:M}}(\boldsymbol{\rho}), \\ s_n|s_{n-1}, \boldsymbol{\Pi} &\sim \text{Categorical}_{\boldsymbol{\sigma}_{1:M}}(\boldsymbol{\pi}_{\sigma_m}), && n = 2:N \\ w_n|s_n, \boldsymbol{\phi} &\sim \mathbb{G}_{\phi_{s_n}}, && n = 1:N \end{aligned}$$

results in a posterior $p(\boldsymbol{\rho}, \boldsymbol{\Pi}, \boldsymbol{\phi}|w_{1:N})$ that converges in the limit $M \to \infty$. Consequently, so long as M is sufficiently large, the HMM above provides estimates independent of the particular M values chosen.

Computational inference on this model can be based on appropriate modifications of the Gibbs or Metropolis-Hastings samplers of algorithms 8.7 and 8.9. The modifications for the latter are straightforward and, for this reason, here we focus only on a presentation of the Gibbs sampler which targets the completed posterior $p(s_{1:N}, \boldsymbol{\beta}, \boldsymbol{\rho}, \boldsymbol{\Pi}, \boldsymbol{\phi}|w_{1:N})$. For this target, only an additional step to update $\boldsymbol{\beta}$ is required of algorithm 8.7. This update needs to sample $\boldsymbol{\beta}$ from its full conditional $p(\boldsymbol{\beta}|s_{1:N}, \boldsymbol{\rho}, \boldsymbol{\Pi}, \boldsymbol{\phi}, w_{1:N})$ which reduces to $p(\boldsymbol{\beta}|\boldsymbol{\rho}, \boldsymbol{\Pi})$. However, since eq. (8.40) is not conjugate with eqs. (8.41) and (8.42), a Metropolis-Hastings step is necessary.

Note 8.15: iHMM

The description and the associated computational schemes we presented in this section rely on a finite approximation of the *infinite hidden Markov model* (iHMM). Formally, the latter is the model achieved in the limiting case $M = \infty$ and entails a truly infinite state-space $\sigma_{1:\infty}$. In this limit, a detailed description of the corresponding generative model involves the Dirichlet and hierarchical Dirichlet processes. It is also possible to carry out our computational inference on the exact iHMM instead of relying on finite approximations. For example, it is possible to carry out MCMC sampling involving an infinite state-space by completing the posterior

$$p(s_{1:N}, \boldsymbol{\beta}, \boldsymbol{\rho}, \boldsymbol{\Pi}|w_{1:N}) = \int du_{1:N}\; p(u_{1:N}, s_{1:N}, \boldsymbol{\beta}, \boldsymbol{\rho}, \boldsymbol{\Pi}|w_{1:N})$$

with auxiliary slice variables $u_{1:N}$ as we developed in example 5.14. The resulting sampler gives rise to *beam sampling* schemes.

8.7 *A case study in fluorescence spectroscopy**

Favoring simplicity, so far we focused on problems where observations depend directly on the underlying hidden states or, as we might call them, on first order HMM. To help illustrate why the methods presented here are more general than what first appears, we describe a case study involving dynamics in continuous time that necessarily leads to a second order HMM as observations occur precisely at jump times. In this case study, we introduce an auxiliary variable method, inspired from section 5.4.3, in order reduce the second order HMM to a first order HMM for which the algorithms provided in this chapter hold. We also demonstrate how to discretize time in order to incorporate continuous time observations. This treatment here is necessary for observations occurring at jump times. More complex models with continuous dynamics and observations at arbitrary times are dealt with in chapter 10.

8.7.1 Time resolved spectroscopy

We start by considering an important class of experiments that does not probe the state of the dynamical system of interest but rather jumps in the system's trajectory. For instance, *time-resolved* spectroscopic experiments collect individual photons and report on their detection time. Since the detected photons stem from the probed physical system, they are emitted precisely when the system (an atom or, more typically, molecule) jumps across energy

*This is an advanced topic and could be skipped on a first reading.

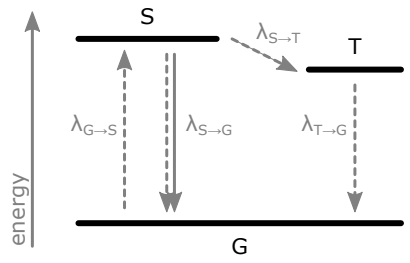

Figure 8.4: Jablonski diagram of a fluorophore possessing three energy states: G, S, and T. Arrows indicate Markovian transitions at the rates shown. Solid and dashed arrows distinguish between detectable and non-detectable transitions.

levels. Since the time a photon needs to reach the detector in such experiments is insignificant, the recorded photon detection times report upon transitions between, rather than instantaneous, states of the system.

In this case study, we consider a fluorescent molecule, *i.e.*, a *fluorophore*, with three energy states labeled with G, S, T. Respectively, these are: the fluorophore's *ground state* (state with the lowest energy); the first excited *singlet state* (state with the highest energy); and the first excited *triplet state* (state with intermediate energy). These are typically depicted schematically, in increasing energy order, using a *Jablonski diagram* like fig. 8.4.

During an experiment, while residing in G, a fluorophore absorbs energy at a random time and undergoes a transition $G \to S$. Subsequently, after residing for a short period in S, the fluorophore undergoes either an $S \to G$ or an $S \to T$ transition. If in T, the fluorophore may only undergo a $T \to G$ transition. All such transitions are denoted with arrows in fig. 8.4. Terminating at G, the fluorophore is re-excited and the same cycle repeats until the conclusion of the experiment. Physical Chemistry often models dwells in each one of the three states as memoryless. This leads to a kinetic scheme fully determined by the transition rates $\lambda_{G\to S}, \lambda_{S\to G}, \lambda_{S\to T}, \lambda_{T\to G}$ also shown in fig. 8.4.

Of interest is often the mean dwell time in the excited state S, *i.e.*, the so called fluorescence lifetime, which helps in characterizing the fluorophore. This is because lifetime is often unique to each molecule or alternative chemical forms of a molecule (assuming that, within error, lifetimes are sufficiently well-separated that they can be distinguished).

On the theoretical front, what makes this set-up challenging to analyze is the fact that photons are emitted and detected only whenever the fluorophore undergoes the transition $S \to G$; while, in a typical experiment, all other transitions are either *non-radiative* or emit photons not otherwise detected. The situation is even more complicated due to the fact that, even when the fluorophore undergoes $S \to G$ transitions, photons may not always be emitted or may not always be detected. Here we formulate this system and show how the general framework for HMMs can be used to estimate transition rates and eventually, through them, the fluorescence lifetime.

8.7.2 Discretization of time

For clarity, we consider an experiment that starts at time $T_{\min}$ and concludes at time $T_{\max}$. Further, we use T_k, with indices $k = 1 : K$, to denote the reported photon detection times which we arrange in ascending order, *i.e.*, $T_{k-1} < T_k$.

To operate within the HMM framework, we must first discretize time. For this, we break the experiment's time course into a total of N hypothetical windows separated by the time levels

$$t_n = T_{\min} + \frac{n}{N}\left(T_{\max} - T_{\min}\right), \qquad n = 0 : N.$$

These time levels define N windows which we successively index by $n = 1 : N$. Specifically, our n^{th} window spans the time interval between t_{n-1} and t_n.

8.7.3 Formulation of the dynamics

Following notation we first introduce in section 2.3, we denote with $\mathcal{S}(t)$ the passing state at time t of our fluorophore. Due to memorylessness, the trajectory $\mathcal{S}(\cdot)$ is a Markov jump process with state-space G, S, and T and its transition rate matrix is given by

$$\boldsymbol{\Lambda} = \begin{bmatrix} 0 & \lambda_{G\to S} & \lambda_{G\to T} \\ \lambda_{S\to G} & 0 & \lambda_{S\to T} \\ \lambda_{T\to G} & \lambda_{T\to S} & 0 \end{bmatrix}.$$

Our end goal is to estimate the unknown entries of $\boldsymbol{\Lambda}$. To do so, we do not need the full trajectory $\mathcal{S}(\cdot)$. Instead, we focus on the passing states only at the time levels t_n which are already sufficient to link $\boldsymbol{\Lambda}$ with our measurements. Accordingly, for each time level, we consider the corresponding passing state

$$s_n = \mathcal{S}(t_n), \qquad\qquad n = 0 : N.$$

As the underlying trajectory is a Markov jump process, we can easily deduce the transition rules of our dynamical model

$$s_n | s_{n-1} \sim \text{Categorical}_{G,S,T}\left(\boldsymbol{\pi}_{s_{n-1}}\right), \qquad\qquad n = 1 : N.$$

According to eq. (2.15), the transition probabilities stem from the rows of the propagator

$$\boldsymbol{\Pi} = \begin{bmatrix} \boldsymbol{\pi}_G \\ \boldsymbol{\pi}_S \\ \boldsymbol{\pi}_T \end{bmatrix} = \begin{bmatrix} \pi_{G\to G} & \pi_{G\to S} & \pi_{G\to T} \\ \pi_{S\to G} & \pi_{S\to S} & \pi_{S\to T} \\ \pi_{T\to G} & \pi_{T\to S} & \pi_{T\to T} \end{bmatrix} = \boldsymbol{Q}^{t_{n-1}\to t_n} = \mathbf{exp}\left(\frac{T_{\max} - T_{\min}}{N}\boldsymbol{G}\right) \tag{8.43}$$

that corresponds to the generator $\boldsymbol{G}$ of the rate matrix $\boldsymbol{\Lambda}$. As with every dynamical system seen so far, the kinetic model does not specify the initial conditions. Consequently, we need to model the initialization rule separately

$$s_0 \sim \text{Categorical}_{G,S,T}(\boldsymbol{\rho})$$

with appropriate initial probabilities $\boldsymbol{\rho} = [\rho_G, \rho_S, \rho_T]$ that may pr may not be related to $\boldsymbol{\Lambda}$ depending upon the specifics of the experiment.

8.7.4 Formulation of the measurements

The most convenient way to model the photon detection times is to consider a set of observation variables $w_{1:N}$, where each one of our windows is associated with its own w_n. We encode the photon detection times $T_{1:K}$ by setting $w_n = 1$ when at least one photon is detected and setting $w_n = 0$ when no photon is detected during our n^{th} window.

Note 8.16: Observations

If we use N_k to denote the window that encodes the k^{th} photon detection time, T_k, we see that

$$N_k = \left\lceil N\frac{T_k - T_{\min}}{T_{\max} - T_{\min}} \right\rceil, \qquad\qquad k = 1 : K$$

where $\lceil x \rceil$ is the ceiling function, *i.e.*, the smallest index that is larger than x.

When we attempt to model our photon detections with a low N, our windows may be large and misleadingly, some of them, may absorb more than one photon detection. However, as N grows large, and our windows correspondingly shrink, the photon detections times $T_{1:K}$ are encoded in different, well separated, windows. For

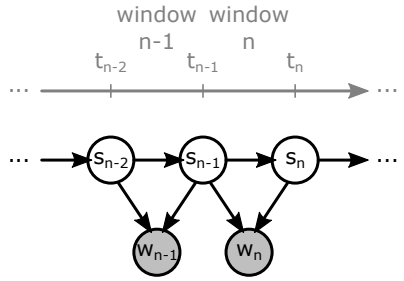

Figure 8.5: A HMM for fluorescence spectroscopy representing time resolved measurements, $T_{1:K}$, by observation variables w_n that are linked to the passing states s_n of the underlying fluorophore. By contrast to the observation variables $w_{1:N}$ measured in an experiment, the passing states $s_{0:N}$ remain hidden.

sufficiently large N, our observation variables $w_{1:N}$ follow the pattern

$$\underbrace{0,\cdots,0}_{\substack{\text{windows}\\1:N_1-1}},\overbrace{1}^{T_1},\underbrace{0,\cdots,0}_{\substack{\text{windows}\\N_1+1:N_2-1}},\overbrace{1}^{T_2},\underbrace{0,\cdots,0}_{\substack{\text{windows}\\N_2+1:N_3-1}},\overbrace{1}^{T_3},0\cdots\qquad\cdots 0,\overbrace{1}^{T_K},\underbrace{0,\cdots,0}_{\substack{\text{windows}\\N_K+1:N}}.$$

On account of this pattern, our observation sequence $w_{1:N}$ contains *no successive* windows with $w_n = 1$. By contrast, it contains multiple successive windows with $w_n = 0$.

Under the variables $w_{1:N}$, it is straightforward to model our assessment rules by

$$w_n|s_{n-1}, s_n \sim \text{Bernoulli}\left(\beta_{s_{n-1}\to s_n}\right), \qquad n = 1:N$$

and, as we have nine possible pairs $s_{n-1} \to s_n$, we need to specify nine different Bernoulli weights. To a good approximation, these are given by

$$\begin{aligned}
&\beta_{G\to G} \approx 0, && \beta_{G\to S} \approx 0, && \beta_{G\to T} \approx 0,\\
&\beta_{S\to G} \approx \eta, && \beta_{S\to S} \approx 0, && \beta_{S\to T} \approx 0,\\
&\beta_{T\to G} \approx 0, && \beta_{T\to S} \approx 0, && \beta_{T\to T} \approx 0,
\end{aligned}$$

where η is the fraction of detectable transitions $S \to G$ to total transitions $S \to G$. To be clear, all approximately zero terms above become strictly zero in the limit that N tends to infinity.

Our approximations on $\beta_{s_{n-1}\to s_n}$ improve and eventually become exact as $N \to \infty$ for which our hypothetical windows become so narrow that they accommodate no more than one transition. For this reason, our end goal is to devise a training method for our model supporting this limit. Put differently, our strategy is to derive a set of training equations on which we can formally reach the $N \to \infty$ limit.

8.7.5 Modeling overview

In summary, the model of time resolved fluorescence spectroscopy developed so far reads

$$\begin{aligned}
s_0 &\sim \text{Categorical}_{G,S,T}(\boldsymbol{\rho}),\\
s_n|s_{n-1} &\sim \text{Categorical}_{G,S,T}\left(\boldsymbol{\pi}_{s_{n-1}}\right), && n = 1:N\\
w_n|s_{n-1}, s_n &\sim \text{Bernoulli}\left(\beta_{s_{n-1}\to s_n}\right), && n = 1:N
\end{aligned}$$

and is depicted graphically in fig. 8.5. An immediate challenge that we face is that each observation variable $w_n|s_{n-1}, s_n$ depends on *two*, rather than one, hidden states. On account of this almost imperceptible difference, with dramatic theoretical consequences, none of the basic algorithms developed in section 8.3 apply.

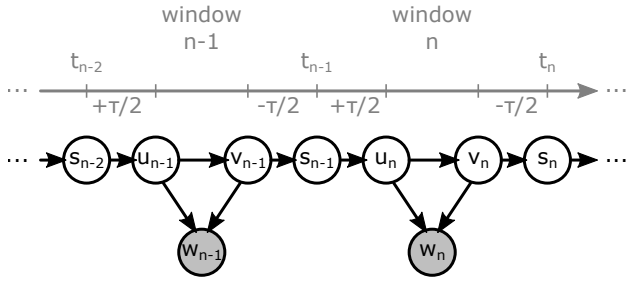

Figure 8.6: A HMM, augmented with additional hidden states $u_{1:N}, v_{1:N}$, is used to decouple successive passing states $s_{0:N}$ from their respective observations $w_{1:N}$.

8.7.6 Reformulation

To continue, we must reformulate our model in such a way that it becomes similar to the HMM devised earlier. Namely, we need to transform it such that each observation is associated with *only one* hidden state.

One way to achieve a transformation is to consider a positive time period τ, but otherwise sufficiently small, $\tau < (T_{\max} - T_{\min})/N$. With the aid of τ, we introduce two auxiliary variables. That is, we consider two additional passing states per time level

$$u_n = \mathcal{S}\left(t_{n-1} + \frac{\tau}{2}\right), \qquad v_n = \mathcal{S}\left(t_n - \frac{\tau}{2}\right), \qquad n = 1:N.$$

From these new states, u_n occurs near the very beginning and v_n occurs near the very end of their respective window. Due to memorylessness, we can exactly represent the dynamics of our system

$$\begin{aligned} u_n|s_{n-1} &\sim \mathsf{Categorical}_{G,S,T}\left(\boldsymbol{\psi}'_{s_{n-1}}\right), && n = 1:N \\ v_n|u_n &\sim \mathsf{Categorical}_{G,S,T}\left(\boldsymbol{\pi}'_{u_n}\right), && n = 1:N \\ s_n|v_n &\sim \mathsf{Categorical}_{G,S,T}\left(\boldsymbol{\psi}'_{v_n}\right), && n = 1:N. \end{aligned}$$

The new transition probabilities are obtained through the rows of the propagators

$$\boldsymbol{\Psi}' = \begin{bmatrix} \boldsymbol{\psi}'_G \\ \boldsymbol{\psi}'_S \\ \boldsymbol{\psi}'_T \end{bmatrix} = \begin{bmatrix} \psi'_{G\to G} & \psi'_{G\to S} & \psi'_{G\to T} \\ \psi'_{S\to G} & \psi'_{S\to S} & \psi'_{S\to T} \\ \psi'_{T\to G} & \psi'_{T\to S} & \psi'_{T\to T} \end{bmatrix} = \boldsymbol{Q}^{t_{n-1}\to t_{n-1}+\frac{\tau}{2}} = \boldsymbol{Q}^{t_n-\frac{\tau}{2}\to t_n} = \mathbf{exp}\left(\frac{\tau}{2}\boldsymbol{G}\right),$$

$$\boldsymbol{\Pi}' = \begin{bmatrix} \boldsymbol{\pi}'_G \\ \boldsymbol{\pi}'_S \\ \boldsymbol{\pi}'_T \end{bmatrix} = \begin{bmatrix} \pi'_{G\to G} & \pi'_{G\to S} & \pi'_{G\to T} \\ \pi'_{S\to G} & \pi'_{S\to S} & \pi'_{S\to T} \\ \pi'_{T\to G} & \pi'_{T\to S} & \pi'_{T\to T} \end{bmatrix} = \boldsymbol{Q}^{t_{n-1}+\frac{\tau}{2}\to t_n-\frac{\tau}{2}} = \mathbf{exp}\left(\left(\frac{T_{\max}-T_{\min}}{N} - \tau\right)\boldsymbol{G}\right).$$

Taking advantage of the new states, and provided τ is sufficiently small, we can introduce another approximation to the observations

$$\beta_{s_{n-1}\to s_n} \approx \beta_{u_n\to v_n}, \qquad n = 1:N.$$

This approximation becomes exact as $\tau \to 0^+$ at which u_n and v_n essentially merge with s_{n-1} and s_n, respectively. Of course, since $\tau < (T_{\max} - T_{\min})/N$, this limiting condition does not introduce further restrictions in our formulation since it is already fulfilled under $N \to \infty$.

Gathering everything together, our reformulated Markov model that leverages auxiliary variables reads

$$
\begin{aligned}
s_0 &\sim \text{Categorical}_{G,S,T}(\boldsymbol{\rho}), && \\
u_n|s_{n-1} &\sim \text{Categorical}_{G,S,T}\left(\boldsymbol{\psi}'_{s_{n-1}}\right), && n = 1:N \\
v_n|u_n &\sim \text{Categorical}_{G,S,T}\left(\boldsymbol{\pi}'_{u_n}\right), && n = 1:N \\
s_n|v_n &\sim \text{Categorical}_{G,S,T}\left(\boldsymbol{\psi}'_{v_n}\right), && n = 1:N \\
w_n|u_n, v_n &\sim \text{Bernoulli}\left(\beta_{u_n \to v_n}\right), && n = 1:N
\end{aligned}
$$

and it is depicted graphically in fig. 8.6. Now, because the states $s_{0:N}$ are no longer directly associated with observations, we can afford to discard them through marginalization, which leads to an equivalent, but somewhat simpler, model

$$
\begin{aligned}
u_1 &\sim \text{Categorical}_{G,S,T}(\boldsymbol{\rho}'), && \\
v_1|u_1 &\sim \text{Categorical}_{G,S,T}\left(\boldsymbol{\pi}'_{u_1}\right), && \\
u_n|v_{n-1} &\sim \text{Categorical}_{G,S,T}\left(\boldsymbol{\psi}''_{v_{n-1}}\right), && n = 2:N \\
v_n|u_n &\sim \text{Categorical}_{G,S,T}\left(\boldsymbol{\pi}'_{u_n}\right), && n = 2:N \\
w_n|u_n, v_n &\sim \text{Bernoulli}\left(\beta_{u_n \to v_n}\right), && n = 1:N
\end{aligned}
$$

which we depict graphically in the left panel of fig. 8.7. Marginalization implies that, in this model, the initial probabilities are given by

$$
\boldsymbol{\rho}' = \begin{bmatrix} \rho'_G & \rho'_S & \rho'_T \end{bmatrix} = \boldsymbol{\rho}\boldsymbol{\Phi}' = \boldsymbol{\rho}\exp\left(\frac{\tau}{2}\boldsymbol{G}\right)
$$

and the transition probabilities by the rows of

$$
\boldsymbol{\Psi}'' = \begin{bmatrix} \boldsymbol{\psi}''_G \\ \boldsymbol{\psi}''_S \\ \boldsymbol{\psi}''_T \end{bmatrix} = \begin{bmatrix} \psi''_{G\to G} & \psi''_{G\to S} & \psi''_{G\to T} \\ \psi''_{S\to G} & \psi''_{S\to S} & \psi''_{S\to T} \\ \psi''_{T\to G} & \psi''_{T\to S} & \psi''_{T\to T} \end{bmatrix} = \boldsymbol{\Psi}'\boldsymbol{\Psi}' = \exp\left(\tau\boldsymbol{G}\right).
$$

Note 8.17: HMM order reduction

The last version of our model represents a conventional HMM as introduced in section 8.2. To make the correspondence clearer, we consider super-states $\xi_n = (u_n, v_n)$, depicted graphically on the right panel of fig. 8.7, and rewrite the model in the equivalent form

$$
\begin{aligned}
\xi_1 &\sim \text{Categorical}_{\chi_{1:9}}(\boldsymbol{r}), && \\
\xi_n|\xi_{n-1} &\sim \text{Categorical}_{\chi_{1:9}}\left(\boldsymbol{P}_{\xi_{n-1}}\right), && n = 2:N \\
w_n|\xi_n &\sim \text{Bernoulli}\left(\beta_{\xi_n}\right), && n = 1:N.
\end{aligned}
$$

The initial, $\boldsymbol{r}$, and transition, $\boldsymbol{P}_\xi$, probabilities are determined according to $\boldsymbol{\rho}', \boldsymbol{\Pi}', \boldsymbol{\Psi}''$. In particular, these are

$$
\begin{aligned}
r_{\xi_1} = p(\xi_1) &= p(u_1, v_1) \\
&= p(v_1|u_1)p(u_1) = \pi'_{u_1\to v_1}\rho'_{u_1}, \\
P_{\xi_{n-1}\to\xi_n} = p(\xi_n|\xi_{n-1}) &= p(u_n, v_n|u_{n-1}, v_{n-1}) \\
&= p(v_n|u_n, u_{n-1}, v_{n-1})p(u_n|u_{n-1}, v_{n-1}) \\
&= p(v_n|u_n)p(u_n|v_{n-1}) = \pi'_{u_n\to v_n}\psi''_{v_{n-1}\to u_n}.
\end{aligned}
$$

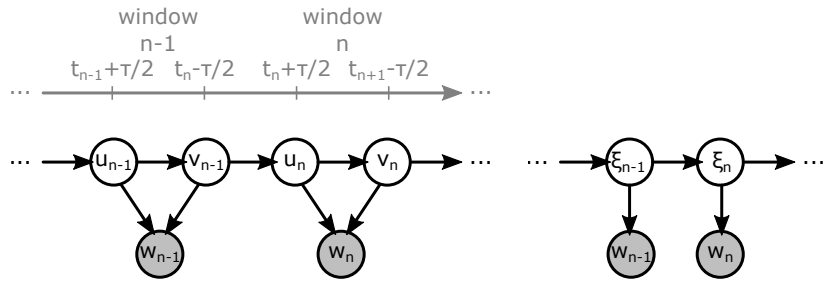

Figure 8.7: Left: A modified HMM with two decoupled passing states per observation. Right: An equivalent HMM with one passing state per observation.

As each super-state is formed by a pair of G, S, and T, our new state-space consists of

$$\begin{aligned} \chi_1 &= GG, & \chi_2 &= GS, & \chi_3 &= GT, \\ \chi_4 &= SG, & \chi_5 &= SS, & \chi_6 &= ST, \\ \chi_7 &= TG, & \chi_8 &= TS, & \chi_9 &= TT, \end{aligned}$$

and, because each constitutive super-state is *derived* from G, S, and T, similar to example 2.8, we follow the common convention and order $\chi_{1:9}$ lexicographically.

8.7.7 Computational training

Via auxiliary variables, we have re-formulated our second order Markov model problem in order to make it amenable to a similar training strategy as the conventional HMM of sections 8.3 and 8.4. In its final version, the unknown parameters are still those of the initial problem, namely the entries of $\boldsymbol{\Lambda}$ and potentially $\boldsymbol{\rho}, \eta$. The likelihood of our model, formally given by $p(w_{1:N}|\boldsymbol{\Lambda}, \boldsymbol{\rho}, \eta)$, can be computed according to eq. (8.7) by completion with the terminal states

$$p\left(w_{1:N}|\boldsymbol{\Lambda}, \boldsymbol{\rho}, \eta\right) = \sum_{u_N, v_N} p\left(w_{1:N}, u_N, v_N|\boldsymbol{\Lambda}, \boldsymbol{\rho}, \eta\right) = \sum_{u_N, v_N} \mathcal{A}_N(u_N, v_N).$$

In turn, the terms of the filter $\mathcal{A}_N(u_N, v_N)$ can be computed by forward filtering. Nevertheless, because N needs to be large, such that our approximate observation representation holds, naive filtering with algorithm 8.1 is impractical. Additionally, even if we were able to perform the filtering recursion in algorithm 8.1 for excessively large N, directly training our model suffers from the approximations induced by having a non-zero τ and finite N. Now we show how to eliminate such approximations altogether and derive a tractable version of the filtering algorithm that carries over the limit $N \to \infty$.

Limit $\tau \to 0^+$

As all of our propagators depend continuously on τ, we can formally apply the $\tau \to 0^+$ limit. Specifically, note 2.18 implies that

$$\mathbf{exp}\left(\frac{\tau}{2}\boldsymbol{G}\right) \to \mathbb{1}, \qquad \mathbf{exp}\left(\tau \boldsymbol{G}\right) \to \mathbb{1}, \qquad \mathbf{exp}\left(\left(\frac{T_{\max} - T_{\min}}{N} - \tau\right)\boldsymbol{G}\right) \to \boldsymbol{\Pi}.$$

Here, $\mathbb{1}$ is the identity matrix of size three. In this limit, we can safely replace our model with the limiting one

$$\begin{aligned}
u_1 &\sim \text{Categorical}_{G,S,T}(\boldsymbol{\rho}), \\
v_1|u_1 &\sim \text{Categorical}_{G,S,T}\left(\boldsymbol{\pi}_{u_1}\right), \\
u_n|v_{n-1} &\sim \text{Categorical}_{G,S,T}\left(\mathbb{1}_{v_{n-1}}\right), && n = 2:N \\
v_n|u_n &\sim \text{Categorical}_{G,S,T}\left(\boldsymbol{\pi}_{u_n}\right), && n = 2:N \\
w_n|u_n, v_n &\sim \text{Bernoulli}\left(\beta_{u_n \to v_n}\right), && n = 1:N
\end{aligned}$$

thereby relaxing any approximation mediated by τ.

Marginal likelihood

Having relaxed the dependency of the model on τ, we now show how to apply forward filtering, *i.e.*, algorithm 8.1. To make our calculations more transparent, we adopt the super-state formalism over super-state $\xi_n = (u_n, v_n)$ of note 8.17 and show how to recursively compute the forward terms of the filter which, in this case, read $\mathcal{A}_n(u_n, v_n) = \mathcal{A}_n(\xi_n)$. Further, to maintain the notation to a minimum, we follow note 8.5 and gather our forward terms in row arrays

$$\mathbb{A}_n = \begin{bmatrix} \mathcal{A}_n(\chi_1) & \mathcal{A}_n(\chi_2) & \mathcal{A}_n(\chi_3) & \mathcal{A}_n(\chi_4) & \mathcal{A}_n(\chi_5) & \mathcal{A}_n(\chi_6) & \mathcal{A}_n(\chi_7) & \mathcal{A}_n(\chi_8) & \mathcal{A}_n(\chi_9) \end{bmatrix}.$$

With this convention, the computation of the (marginal) likelihood, L, reduces to

$$L = \mathbb{A}_N \boldsymbol{\Sigma}, \qquad \boldsymbol{\Sigma} = \boldsymbol{\sigma} \otimes \boldsymbol{\sigma}, \qquad \boldsymbol{\sigma} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

where, to be clear, $\boldsymbol{\Sigma}$ is simply a row vector populated by ones.

Note 8.18: Vectorization of r and P

According to note 8.17, the model in section 8.7.7, leads to the tabulations

$$\boldsymbol{r} = \begin{bmatrix} \rho_G\pi_{G\to G} & \rho_G\pi_{G\to S} & \rho_G\pi_{G\to T} & \rho_S\pi_{S\to G} & \rho_S\pi_{S\to S} & \rho_S\pi_{S\to T} & \rho_T\pi_{T\to G} & \rho_T\pi_{T\to S} & \rho_T\pi_{T\to T} \end{bmatrix},$$

$$\boldsymbol{P} = \begin{bmatrix} \pi_{G\to G} & \pi_{G\to S} & \pi_{G\to T} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \pi_{S\to G} & \pi_{S\to S} & \pi_{S\to T} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \pi_{T\to G} & \pi_{T\to S} & \pi_{T\to T} \\ \pi_{G\to G} & \pi_{G\to S} & \pi_{G\to T} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \pi_{S\to G} & \pi_{S\to S} & \pi_{S\to T} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \pi_{T\to G} & \pi_{T\to S} & \pi_{T\to T} \\ \pi_{G\to G} & \pi_{G\to S} & \pi_{G\to T} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \pi_{S\to G} & \pi_{S\to S} & \pi_{S\to T} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \pi_{T\to G} & \pi_{T\to S} & \pi_{T\to T} \end{bmatrix}.$$

Adopting array operations, both are vectorized as

$$\begin{aligned}
\boldsymbol{r} &= \left(\boldsymbol{\rho} \otimes \boldsymbol{\sigma}^t\right) \odot \left(\boldsymbol{a}_G^t \boldsymbol{\Pi} \boldsymbol{B}_G + \boldsymbol{a}_S^t \boldsymbol{\Pi} \boldsymbol{B}_S + \boldsymbol{a}_T^t \boldsymbol{\Pi} \boldsymbol{B}_T\right), \\
\boldsymbol{P} &= \left(\boldsymbol{\sigma} \otimes \boldsymbol{I} \otimes \boldsymbol{\sigma}^t\right) \odot \left(\boldsymbol{A}_G^t \boldsymbol{\Pi} \boldsymbol{B}_G + \boldsymbol{A}_S^t \boldsymbol{\Pi} \boldsymbol{B}_S + \boldsymbol{A}_T^t \boldsymbol{\Pi} \boldsymbol{B}_T\right)
\end{aligned}$$

where $\otimes, \odot$ denote the Kronecker and Hadamard product, respectively, and the auxiliary arrays are

$$\boldsymbol{a}_G = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \quad \boldsymbol{A}_G = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad \boldsymbol{B}_G = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix},$$

$$\boldsymbol{a}_S = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \quad \boldsymbol{A}_S = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad \boldsymbol{B}_S = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix},$$

$$\boldsymbol{a}_T = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \qquad \boldsymbol{A}_T = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}, \qquad \boldsymbol{B}_T = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

From eq. (!8.9), we see that the filtering updates follow the recursion

$$\mathcal{A}_n(\xi_n) = \sum_{\xi_{n-1}} \text{Bernoulli}(w_n; \beta_{\xi_n}) P_{\xi_{n-1} \to \xi_n} \mathcal{A}_{n-1}(\xi_{n-1}), \qquad n = 2:N$$

which we can vectorize as

$$\mathbb{A}_n = \mathbb{A}_{n-1} \boldsymbol{P}_{w_n}, \qquad n = 2:N.$$

Note 8.19: Vectorization of P_0 and P_1

The matrices, $\boldsymbol{P}_0$ and $\boldsymbol{P}_1$, required in the filtering updates are tabulated in

$$\boldsymbol{P}_0 = \begin{bmatrix} \pi_{G\to G} & \pi_{G\to S} & \pi_{G\to T} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \zeta_0\pi_{S\to G} & \pi_{S\to S} & \pi_{S\to T} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \pi_{T\to G} & \pi_{T\to S} & \pi_{T\to T} \\ \pi_{G\to G} & \pi_{G\to S} & \pi_{G\to T} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \zeta_0\pi_{S\to G} & \pi_{S\to S} & \pi_{S\to T} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \pi_{T\to G} & \pi_{T\to S} & \pi_{T\to T} \\ \pi_{G\to G} & \pi_{G\to S} & \pi_{G\to T} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \zeta_0\pi_{S\to G} & \pi_{S\to S} & \pi_{S\to T} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \pi_{T\to G} & \pi_{T\to S} & \pi_{T\to T} \end{bmatrix},$$

$$\boldsymbol{P}_1 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \zeta_1\pi_{S\to G} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \zeta_1\pi_{S\to G} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \zeta_1\pi_{S\to G} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

with $\zeta_0 = 1 - \eta$ and $\zeta_1 = \eta$. Similar to $\boldsymbol{P}$, these are vectorized by

$$\boldsymbol{P}_0 = \left(\boldsymbol{\sigma} \otimes \boldsymbol{I} \otimes \boldsymbol{\sigma}^t\right) \odot \left(\boldsymbol{A}_G^t \boldsymbol{\Pi}_0 \boldsymbol{B}_G + \boldsymbol{A}_S^t \boldsymbol{\Pi}_0 \boldsymbol{B}_S + \boldsymbol{A}_T^t \boldsymbol{\Pi}_0 \boldsymbol{B}_T\right),$$
$$\boldsymbol{P}_1 = \left(\boldsymbol{\sigma} \otimes \boldsymbol{I} \otimes \boldsymbol{\sigma}^t\right) \odot \left(\boldsymbol{A}_G^t \boldsymbol{\Pi}_1 \boldsymbol{B}_G + \boldsymbol{A}_S^t \boldsymbol{\Pi}_1 \boldsymbol{B}_S + \boldsymbol{A}_T^t \boldsymbol{\Pi}_1 \boldsymbol{B}_T\right).$$

In $\boldsymbol{P}_0$ and $\boldsymbol{P}_1$, we use $\boldsymbol{\Pi}_0$ and $\boldsymbol{\Pi}_1$ to discriminate between detection-less and detection-full pseudo-propagators

$$\boldsymbol{\Pi}_0 = \boldsymbol{Z}_0 \odot \boldsymbol{\Pi}, \qquad \boldsymbol{\Pi}_1 = \boldsymbol{Z}_1 \odot \boldsymbol{\Pi}$$

where with $\boldsymbol{Z}_0$ and $\boldsymbol{Z}_1$, termed "masks", we encode detection-less and detection-full transitions

$$\boldsymbol{Z}_0 = \begin{bmatrix} 1 & 1 & 1 \\ \zeta_0 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}, \qquad \boldsymbol{Z}_1 = \begin{bmatrix} 0 & 0 & 0 \\ \zeta_1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}.$$

As $\boldsymbol{Z}_0$ and $\boldsymbol{Z}_1$ encode our observation rules, *i.e.,* encode time windows with either 0 or 1 detections, the pseudo-propagators are related by $\boldsymbol{\Pi} = \boldsymbol{\Pi}_0 + \boldsymbol{\Pi}_1$. Additionally, because photons are emitted only when our system *jumps* across constitutive states, the diagonal entires in $\boldsymbol{Z}_0$ are all one. By contrast, the diagonal entries in $\boldsymbol{Z}_1$ are all zero.

Finally, according to eq. (!8.10), the filter is initialized with $\mathcal{A}_1(\xi_1) = \text{Bernoulli}(w_1; \beta_{\xi_1}) r_{\xi_1}$ which, in vectorized form reads

$$\mathbb{A}_1 = \left(\boldsymbol{\rho} \otimes \boldsymbol{\sigma}^t\right) \odot \left(\boldsymbol{a}_G^t \boldsymbol{\Pi}_{w_1} \boldsymbol{B}_G + \boldsymbol{a}_S^t \boldsymbol{\Pi}_{w_1} \boldsymbol{B}_S + \boldsymbol{a}_T^t \boldsymbol{\Pi}_{w_1} \boldsymbol{B}_T\right).$$

Note 8.20: Vectorization

With the aid of two operators

$$\mathbb{L}\left(\boldsymbol{C}\right) = \left(\boldsymbol{\rho}\boldsymbol{C}\right) \otimes \boldsymbol{\sigma}^t, \qquad \mathbb{D}\left(\boldsymbol{C}\right) = \boldsymbol{a}_G^t \boldsymbol{C} \boldsymbol{B}_G + \boldsymbol{a}_S^t \boldsymbol{C} \boldsymbol{B}_S + \boldsymbol{a}_T^t \boldsymbol{C} \boldsymbol{B}_T$$

defined over the 3×3 matrices $\boldsymbol{C}$; the initial forward term takes a much simpler form

$$\mathbb{A}_1 = \mathbb{L}\left(\boldsymbol{I}\right) \odot \mathbb{D}\left(\boldsymbol{\Pi}_{w_1}\right).$$

By induction, we can now show that the forward variables, $\mathbb{A}_{1:N}$, satisfy an important relationship

$$\mathbb{A}_n = \mathbb{L}\left(\boldsymbol{I}\boldsymbol{\Pi}_{w_1} \cdots \boldsymbol{\Pi}_{w_{n-1}}\right) \odot \mathbb{D}\left(\boldsymbol{\Pi}_{w_n}\right), \qquad n = 1 : N.$$

Accordingly, the (marginal) likelihood is given by

$$L = \left[\mathbb{L}\left(\boldsymbol{\Pi}_{w_1} \cdots \boldsymbol{\Pi}_{w_{N-1}}\right) \odot \mathbb{D}\left(\boldsymbol{\Pi}_{w_N}\right)\right] \boldsymbol{\Sigma} = \mathbb{L}\left(\boldsymbol{\Pi}_{w_1} \cdots \boldsymbol{\Pi}_{w_{N-1}}\right) \left[\mathbb{D}\left(\boldsymbol{\Pi}_{w_N}\right)\right]^t = \boldsymbol{\rho}\, \boldsymbol{\Pi}_{w_1} \cdots \boldsymbol{\Pi}_{w_N} \boldsymbol{\sigma}.$$

Limit $N \to \infty$

According to note 8.16, the product of the pseudo-propagators in our likelihood takes the form

$$\begin{aligned}
\boldsymbol{\Pi}_{w_1} \cdots \boldsymbol{\Pi}_{w_N} &= \overbrace{\boldsymbol{\Pi}_0 \cdots \boldsymbol{\Pi}_0}^{\substack{\text{windows} \\ 1:N_1-1}} \boldsymbol{\Pi}_1 \overbrace{\boldsymbol{\Pi}_0 \cdots \boldsymbol{\Pi}_0}^{\substack{\text{windows} \\ N_1+1:N_2-1}} \boldsymbol{\Pi}_1 \boldsymbol{\Pi}_0 \cdots \qquad \cdots \boldsymbol{\Pi}_0 \boldsymbol{\Pi}_1 \overbrace{\boldsymbol{\Pi}_0 \cdots \boldsymbol{\Pi}_0}^{\substack{\text{windows} \\ N_K+1:N}} \\
&= \boldsymbol{\Pi}_0^{N_1-1} \boldsymbol{\Pi}_1 \boldsymbol{\Pi}_0^{N_2-N_1-1} \boldsymbol{\Pi}_1 \cdots \qquad \cdots \boldsymbol{\Pi}_0^{N_K-N_{K-1}-1} \boldsymbol{\Pi}_1 \boldsymbol{\Pi}_0^{N-N_K}.
\end{aligned}$$

Note 8.21: Asymptotics

Considering the limit $N \to \infty$, from eq. (8.43), we see

$$\boldsymbol{\Pi} = \boldsymbol{I} + \frac{T_{\max} - T_{\min}}{N} \boldsymbol{G} + \mathcal{O}\left(\frac{1}{N^2}\right),$$

which we may also use to approximate the pseudo-propagators. Specifically, as $\boldsymbol{\Pi}_0 = \boldsymbol{\Pi} \odot \boldsymbol{Z}_0$ and $\boldsymbol{\Pi}_1 = \boldsymbol{\Pi} \odot \boldsymbol{Z}_1$, we readily derive

$$\begin{aligned}
\boldsymbol{\Pi}_0 &= \boldsymbol{I} + \frac{T_{\max} - T_{\min}}{N} \boldsymbol{G}_0 + \mathcal{O}\left(\frac{1}{N^2}\right) = \mathbf{exp}\left(\frac{T_{\max} - T_{\min}}{N} \boldsymbol{G}_0\right) + \mathcal{O}\left(\frac{1}{N^2}\right), \\
\boldsymbol{\Pi}_1 &= \frac{T_{\max} - T_{\min}}{N} \boldsymbol{G}_1 + \mathcal{O}\left(\frac{1}{N^2}\right)
\end{aligned}$$

where $\boldsymbol{G}_0 = \boldsymbol{G} \odot \boldsymbol{Z}_0$ and $\boldsymbol{G}_1 = \boldsymbol{G} \odot \boldsymbol{Z}_1$.

Additionally, according to the definition of N_k in note 8.16, we have

$$N_k \frac{T_{\max} - T_{\min}}{N} = \frac{T_{\max} - T_{\min}}{N} + \mathcal{O}\left(\frac{1}{N^2}\right), \qquad k = 1 : K.$$

Putting everything together, we obtain an asymptotic expression of our likelihood

$$L = \left(\frac{T_{\max} - T_{\min}}{N}\right)^K \ell + \mathcal{O}\left(\frac{1}{N^{K+1}}\right). \tag{8.44}$$

where ℓ is *independent* of N. Specifically, ℓ is given by

$$\ell = \boldsymbol{\rho}\,\mathbf{exp}\left(g_0\boldsymbol{G}_0\right)\boldsymbol{G}_1\mathbf{exp}\left(g_1\boldsymbol{G}_0\right)\boldsymbol{G}_1\cdots\boldsymbol{G}_1\mathbf{exp}\left(g_{K-1}\boldsymbol{G}_0\right)\boldsymbol{G}_1\mathbf{exp}\left(g_K\boldsymbol{G}_0\right)\boldsymbol{\sigma}. \tag{8.45}$$

As we can see, ℓ depends only on $\boldsymbol{\Lambda}, \boldsymbol{\rho}, \eta$ and the successive time lags

$$g_0 = T_1 - T_{\min}, \qquad g_1 = T_2 - T_1, \qquad \cdots \qquad g_{K-1} = T_K - T_{K-1}, \qquad g_K = T_{\max} - T_K.$$

8.7.8 Bayesian considerations

From eq. (8.44), it becomes clear that the unknown parameters in our formulation enter the model's likelihood in a complicated way rendering it pointless to seek training through the Baum-Welch method of section 8.3.3 simply because closed form expressions do not follow from the derivatives in the M-step. Similarly, as conjugate priors are unavailable, Bayesian training like those in section 8.4.1 is also not possible.

A viable training strategy, however, is through a Metropolis-Hastings MCMC scheme where, under non-conjugate prior assignments, proposals are drawn and subsequently accepted or rejected according to the (marginal) posterior. As this strategy is quite general, here we consider a wider problem, where the unknown parameters may include not only entries of the transition rate matrix $\boldsymbol{\Lambda}$, but also initial probabilities $\boldsymbol{\rho}$ and observation parameter η.

For clarity, we gather the unknown parameters in $\boldsymbol{\theta}$ and, to stress their dependence, we denote with $\ell\left(\boldsymbol{\theta}\right)$ the product in eq. (8.45). With this formalism, our priors, which need to be specified, are encoded in $p(\boldsymbol{\theta})$ and our likelihood is given, only asymptotically, by

$$p(w_{1:N}|\boldsymbol{\theta}) = \left(\frac{T_{\max} - T_{\min}}{N}\right)^K \ell\left(\boldsymbol{\theta}\right) + \mathcal{O}\left(\frac{1}{N^{K+1}}\right).$$

As in section 5.2.1, using an appropriate Metropolis-Hastings proposal $q(\boldsymbol{\theta}^{\text{prop}}|\boldsymbol{\theta}^{\text{old}})$, we arrive at the acceptance ratio, eq. (5.8), of the form

$$A_N\left(\boldsymbol{\theta}^{\text{prop}}|\boldsymbol{\theta}^{\text{old}}\right) = \frac{p\left(w_{1:N}|\boldsymbol{\theta}^{\text{prop}}\right)}{p\left(w_{1:N}|\boldsymbol{\theta}^{\text{old}}\right)}\frac{p\left(\boldsymbol{\theta}^{\text{prop}}\right)}{p\left(\boldsymbol{\theta}^{\text{old}}\right)}\frac{q\left(\boldsymbol{\theta}^{\text{old}}|\boldsymbol{\theta}^{\text{prop}}\right)}{q\left(\boldsymbol{\theta}^{\text{prop}}|\boldsymbol{\theta}^{\text{old}}\right)}.$$

For any finite choice of N, this ratio is intractable. However, the limiting case $N \to \infty$ leads to

$$A_\infty\left(\boldsymbol{\theta}^{\text{prop}}|\boldsymbol{\theta}^{\text{old}}\right) = \frac{\ell\left(\boldsymbol{\theta}^{\text{prop}}\right)}{\ell\left(\boldsymbol{\theta}^{\text{old}}\right)}\frac{p\left(\boldsymbol{\theta}^{\text{prop}}\right)}{p\left(\boldsymbol{\theta}^{\text{old}}\right)}\frac{q\left(\boldsymbol{\theta}^{\text{old}}|\boldsymbol{\theta}^{\text{prop}}\right)}{q\left(\boldsymbol{\theta}^{\text{prop}}|\boldsymbol{\theta}^{\text{old}}\right)}$$

which we can readily evaluate numerically.

Example 8.6: Bayesian fluorescence spectroscopy

In the most general case, the unknowns in a typical problem of interest in fluorescence spectroscopy may include: all transition rates $\lambda_{G\to S}, \lambda_{S\to G}, \lambda_{S\to T}, \lambda_{T\to G}$, all initial probabilities ρ_G, ρ_S, ρ_T, as well as η. Convenient prior choices then include

$$\lambda_{G\to S} \sim \text{Gamma}\left(2, \frac{\lambda_{\text{ref}}}{2}\right), \qquad \lambda_{S\to G} \sim \text{Gamma}\left(2, \frac{\lambda_{\text{ref}}}{2}\right),$$
$$\lambda_{S\to T} \sim \text{Gamma}\left(2, \frac{\lambda_{\text{ref}}}{2}\right), \qquad \lambda_{T\to G} \sim \text{Gamma}\left(2, \frac{\lambda_{\text{ref}}}{2}\right),$$
$$\boldsymbol{\rho} \sim \text{Dirichlet}_3\left(\frac{1}{3}, \frac{1}{3}, \frac{1}{3}\right), \qquad \eta \sim \text{Beta}\left(1, 1\right).$$

In these priors, the hyperparameters may be adjusted to incorporate prior confidence on certain values and λ_{ref} can be used to set *a priori* appropriate timescales.

For numerical stability, it is preferable to use unitless priors

$$\tilde{\lambda}_{G\to S} \sim \text{Gamma}\left(2, \frac{1}{2}\right), \qquad \tilde{\lambda}_{S\to G} \sim \text{Gamma}\left(2, \frac{1}{2}\right)$$
$$\tilde{\lambda}_{S\to T} \sim \text{Gamma}\left(2, \frac{1}{2}\right), \qquad \tilde{\lambda}_{T\to G} \sim \text{Gamma}\left(2, \frac{1}{2}\right)$$

and implement the timescale through eq. (8.45) cast in the form

$$\ell = \boldsymbol{\rho}\,\mathbf{exp}\left(\tilde{g}_0\tilde{\boldsymbol{G}}_0\right)\tilde{\boldsymbol{G}}_1\mathbf{exp}\left(\tilde{g}_1\tilde{\boldsymbol{G}}_0\right)\tilde{\boldsymbol{G}}_1\cdots\tilde{\boldsymbol{G}}_1\mathbf{exp}\left(\tilde{g}_{K-1}\tilde{\boldsymbol{G}}_0\right)\tilde{\boldsymbol{G}}_1\mathbf{exp}\left(\tilde{g}_K\tilde{\boldsymbol{G}}_0\right)\boldsymbol{\sigma}.$$

with $\tilde{g}_k = g_k\lambda_{\text{ref}}$ and where $\tilde{\boldsymbol{G}}$ is the coinciding $\boldsymbol{G}$ constructed from $\tilde{\boldsymbol{\Lambda}}$. Numerical stability can be further increased if the fastest timescale is separated and evaluated analytically. In particular, if $\tilde{\lambda}_{\text{fast}}$ denotes the fastest rate, $\tilde{\boldsymbol{G}}_0$ can be replaced by $\tilde{\boldsymbol{\Gamma}}_0 - \tilde{\lambda}_{\text{fast}}\mathbb{1}$. This way, ℓ results in

$$\ell = e^{-\tilde{\lambda}_{\text{fast}}\lambda_{\text{ref}}(T_{\max}-T_{\min})}\boldsymbol{\rho}\,\mathbf{exp}\left(\tilde{g}_0\tilde{\boldsymbol{\Gamma}}_0\right)\tilde{\boldsymbol{G}}_1\mathbf{exp}\left(\tilde{g}_1\tilde{\boldsymbol{\Gamma}}_0\right)\tilde{\boldsymbol{G}}_1\cdots\tilde{\boldsymbol{G}}_1\mathbf{exp}\left(\tilde{g}_{K-1}\tilde{\boldsymbol{\Gamma}}_0\right)\tilde{\boldsymbol{G}}_1\mathbf{exp}\left(\tilde{g}_K\tilde{\boldsymbol{\Gamma}}_0\right)\boldsymbol{\sigma}.$$

8.8 Exercise problems

Exercise 8.1: EM for Poisson HMM

Adapt the Baum-Welch algorithm to train a HMM with Poisson emissions. For concreteness, consider the model

$$\begin{aligned} s_1|\boldsymbol{\rho} &\sim \text{Categorical}_{\sigma_{1:M}}(\boldsymbol{\rho}), \\ s_n|s_{n-1},\boldsymbol{\Pi} &\sim \text{Categorical}_{\sigma_{1:M}}(\boldsymbol{\pi}_{\sigma_m}), && n = 2:N \\ w_n|s_n,\boldsymbol{\phi} &\sim \text{Poisson}(\phi_{s_n}), && n = 1:N. \end{aligned}$$

Compare your parameters $\boldsymbol{\rho}, \boldsymbol{\Pi}$ and $\boldsymbol{\phi}$ estimated with Baum-Welch with the ground truth you used to generate your synthetic data.

Exercise 8.2: Implementing Viterbi

Generate observations $w_{1:N}$ using ancestral sampling for a simple HMM with two states. Assume known kinetic and emission parameters. Implement the Viterbi algorithm, algorithm 8.3, to find the sequence $s^\sharp_{1:N}$. Compare your $s^\sharp_{1:N}$ to ground truth.

Exercise 8.3: Bayesian model for Poisson HMM

Consider the same model as in exercise 8.1 and provide a Bayesian formulation that estimates all unknown model parameters. Make your own choices for the priors and briefly justify your choices. Histogram your MCMC samples and indicate the ground truth.

Exercise 8.4: HMMs with common parameters

Consider a total of Q independent HMMs whose dynamics and observations are influenced by the same $\boldsymbol{\rho}, \boldsymbol{\Pi}, \boldsymbol{\phi}$. This scenario is typical of experiments where we try to estimate $\boldsymbol{\rho}, \boldsymbol{\Pi}, \boldsymbol{\phi}$ from a number of short traces. Here we need to consider a joint likelihood over all traces and apply a common prior over the parameters. For each trace, we have

$$\begin{aligned} s^q_1|\boldsymbol{\rho} &\sim \text{Categorical}_{\sigma_{1:M}}(\boldsymbol{\rho}), && q = 1:Q \\ s^q_n|s^q_{n-1},\boldsymbol{\Pi} &\sim \text{Categorical}_{\sigma_{1:M}}\left(\boldsymbol{\pi}_{s^q_{n-1}}\right), && n = 2:N, \quad q = 1:Q \end{aligned}$$

$$w_n^q | s_n^q, \boldsymbol{\phi} \sim \mathbb{G}_{\phi_{s_n^q}}, \qquad n = 1:N, \qquad k = 1:Q.$$

1. Adapt the Baum-Welch algorithm to train the resulting model.
2. Develop a Bayesian model that estimates all model parameters, represent your model graphically, and describe a MCMC sampling scheme.

Exercise 8.5: A sticky HMM

Here we provide a Bayesian model that estimates all dynamic parameters for a HMM with two states and Normal emissions. We assume that the emission parameters are known and our only goal is to estimate dynamical parameters.

Start by generating synthetic data and assume that your escape probabilities coincide with escape rates of the same order of magnitude for each state.

Next perform inference using a sticky HMM. As you implement the sticky HMM, consider three cases: one where the hyperparameters of note 8.12 are tuned to approximately match the dwell time of each state in your synthetic data; and two more cases where the hyperparameters under- and overestimate the dwell by an order of magnitude.

For all three cases, histogram your MCMC samples for your kinetic parameters and indicate the ground truth in your histogram.

Exercise 8.6: The iHMM

Here we consider the iHMM of section 8.6.

1. Generate synthetic data with $M = 3$ states using the usual ancestral sampling scheme of a HMM model. Assume a Normal emission model with, for simplicity, the same variance in each state.
2. Implement the Gibbs sampler proposed in section 8.6 to sample kinetic parameters and mean levels of the emission distributions.
3. Repeat the above steps for $M = 10$ and $M = 50$.
4. Use your MCMC samples and histogram the fraction of time spent in each state. Compare your results to the ground truth and the mean expected time derived from your prior.

Project 8.1: De-drifting a trace in HMM analysis

In experimental techniques, such as *force spectroscopy*, the apparatus collecting data drifts over time giving rise to an apparently low frequency undulation added on top of the signal. In force spectroscopy, the slow drift of an optical trap holding a micron-sized bead corrupts our assessment of its position used as a microscopic measure of force impinged upon the bead. Often, this force can be imparted by a molecule undergoing transitions in a discrete state-space through a dual optical trap setup (*e.g.*, Comstock *et al.* Ultrahigh-resolution optical trap with single-fluorophore sensitivity. Nat. Meth. 8:335, 2011).

To learn properties of a system free from the corruption introduced by drift, we consider a HMM with two states in the presence of drift, $d(\cdot)$, captured by the following generative model

$$\begin{aligned}
d(\cdot) \sim& \text{GaussianP}(\mu_{\text{drift}}(\cdot), C_{\text{drift}}(\cdot,\cdot)),\\
s_1 | \boldsymbol{\rho} \sim& \text{Categorical}_{\sigma_{1:2}}(\boldsymbol{\rho}),\\
s_{n+1} | s_n, \boldsymbol{\Pi} \sim& \text{Categorical}_{\sigma_{1:2}}(\boldsymbol{\pi}_{\sigma_m}),\\
w_n | s_n, d(\cdot) \sim& \text{Normal}\left(\mu_{s_n} + d(t_n), v\right).
\end{aligned}$$

1. Simulate about a 10^3 point trajectory using the familiar squared exponential $C_{\text{drift}}(\cdot,\cdot)$ with prefactor equal to 2 and length scale equal to 500 times the time step size. Set $M = 2$, $\mu_{\text{drift}} = 0$, $\pi_{\sigma_1 \to \sigma_1} = \pi_{\sigma_2 \to \sigma_2} = 0.9$, $\pi_{\sigma_1 \to \sigma_2} = \pi_{\sigma_2 \to \sigma_1} = 0.1$, μ_{σ_1}=-5, μ_{σ_2}=5, $v = 1$ (in rescaled "unitless" units). That is, drift should occur on a slow timescale as compared to other time scales of the problem.
2. Place appropriate priors and implement a MCMC sampling scheme to estimate $d(\cdot), \boldsymbol{\Pi}$ under known v. As a prior on $d(\cdot)$ use a GaussianP with a squared exponential $C_{\text{drift}}(\cdot,\cdot)$ whose parameters are close to what was used to generate the data.

3. Plot various samples of your $d(\cdot)$ and compare to ground truth. Also, histogram your values for $\mathbf{\Pi}$ and compare with ground truth.

Project 8.2: A Bayesian HMM for raw FRET measurements

In fluorescent experiments relying on *Förster resonance energy transfer* (FRET) measurements, we typically obtain two scalar measurements, w_n^D and w_n^A, at each time level t_n. These are the number of photons emitted by a fluorescent label (called a fluorophore) designated as *donor* and the number of photons emitted by a second fluorophore designated as *acceptor*, respectively.

The donor and acceptor can be located on two ends of a molecule. When the donor and acceptor move close to one another, as a molecule collapses on itself or folds, energy can be transferred from a donor (typically directly excited by a laser light) to an acceptor. As such, the origin of the photons (whether higher energy photons from the donor or lower energy photons from the acceptor) report back on the conformational state of a molecule.

As individual photons are emitted by the fluorophores independently, the raw measurements are described by

$$w_n^D|s_n \sim \text{Poisson}\left(\mu_{s_n}^D\right), \qquad w_n^A|s_n \sim \text{Poisson}\left(\mu_{s_n}^A\right), \qquad n = 1:N$$

where s_n is the conformational state of the molecule attached to the two fluorophores and the state dependent parameters $\mu_{\sigma_1}^D, \cdots, \mu_{\sigma_M}^D$ and $\mu_{\sigma_1}^A, \cdots, \mu_{\sigma_M}^A$ are the corresponding average photon emissions per unit time.

1. Set up a Bayesian HMM for the analysis of measurements $w_{1:N}^D$ and $w_{1:N}^A$, generated from synthetic data, from the donor and acceptor channels. Typical values are $N = 1000$, $M = 3$ and $\mu_{\sigma_m}^D, \mu_{\sigma_m}^A$ in the range 100–1000 photons/s.
2. Describe an MCMC sampling scheme for the model posterior in part 1.
3. Implement the MCMC sampling scheme of step 2.
4. Verify, using synthetic data, that your implementation of step 3 generates samples with the correct statistics.

In FRET experiments, a common issue is the *crossover* of photons into the wrong photon detector due to spectral overlap. Crossover is generally given as a matrix of probabilities

$$C = \begin{bmatrix} c_{D\to D} & c_{D\to A} \\ c_{A\to D} & c_{A\to A} \end{bmatrix}$$

where, for example, $c_{D\to A}$ is the probability of a donor photon detected in the acceptor channel. Due to conservation, these probabilities satisfy $c_{D\to D} + c_{D\to A} = 1$ and $c_{A\to D} + c_{A\to A} = 1$. Typical values are $c_{D\to A}, c_{A\to D}$ in the range 5–15%.

5. Show that with crossover, the measurements are described by

$$w_n^D|s_n \sim \text{Poisson}\left(c_{D\to D}\mu_{s_n}^D + c_{A\to D}\mu_{s_n}^A\right),$$
$$w_n^A|s_n \sim \text{Poisson}\left(c_{D\to A}\mu_{s_n}^D + c_{A\to A}\mu_{s_n}^A\right).$$

6. Modify the Bayesian model of step 1 to incorporate crossover, assuming known crossover probabilities, and implement and verify your MCMC.

Project 8.3: A Bayesian HMM for FRET efficiency measurements

In a FRET experiment like in project 8.2, most often w_n^D and w_n^A are combined into a single scalar quantity

$$\epsilon_n = \frac{w_n^A}{w_n^A + w_n^D}$$

which is termed the (apparent) *FRET efficiency*. In this case, the observation model takes a simpler form

$$\epsilon_n|s_n \sim \mathbb{G}_{\phi_{\sigma_m}}$$

where $\phi_{\sigma_m} = (\mu_{\sigma_m}^D, \mu_{\sigma_m}^A)$. In general, the probability density $G_\phi(\epsilon)$ is analytically intractable. However, provided *all* emission levels are high enough, we can safely use the approximations

$$\text{Poisson}\left(w^D; \mu^D\right) \approx \text{Gamma}\left(w^D; \mu^D, 1\right), \qquad \text{Poisson}\left(w^A; \mu^A\right) \approx \text{Gamma}\left(w^A; \mu^A, 1\right).$$

1. Consider these approximations and derive an analytic formula for the resulting emission density $G_\phi(\epsilon)$.
2. Set up a Bayesian HMM for the analysis of apparent FRET efficiencies $\epsilon_{1:N}$.
3. Describe an MCMC sampling scheme for the posterior of the model in step 2.
4. Implement the MCMC sampling scheme of step 3.
5. Verify, using synthetic data, that your implementation of step 4 generates samples with the correct statistics. As in project 8.3, typical values are $N = 1000$, $M = 3$ and $\mu^D_{\sigma_m}, \mu^A_{\sigma_m}$ in the range 100–1000 photons/s.

Additional Reading

C Bishop. Pattern recognition and machine learning. Springer, 2006.
O Cappé, E Moulines, T Rydén. Inference in hidden Markov models. Springer, 2005.
S Särkkä. Bayesian filtering and smoothing. Cambridge University Press, 2013.
LR Rabiner and B Juang. An introduction to hidden Markov models. IEEE ASSP Magazine. 3:4, 1986.
LR Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. Proc. IEEE. 77:257, 1989.
M Beal, Z Ghahramani, C Rasmussen. The infinite hidden Markov model. NIPS, 2001.
Z Ghahramani, M Jordan. Factorial hidden Markov models. NIPS, 1995.
J van Gael, Y Teh, Z Ghahramani. The infinite factorial hidden Markov model. NIPS, 2008.
EB Fox, EB Sudderth, M Jordan, AS Willsky. A sticky HDP-HMM with application to speaker diarization. Ann. Appl. Stat. 5:1020, 2011.
J van Gael, Y Saatci, TW Teh, Z Ghahramani. Beam sampling for the infinite hidden Markov model. Proc. 25th Intl. Conf. Mach. Learn. 1088, 2008.
A Saurabh, M Safar, I Sgouralis, M Fazel, S Pressé. Single photon smFRET. I. Theory and conceptual basis. bioRxiv. 2022.07.20.500887, 2022.
IV Gopich, A Szabo. Theory of the statistics of kinetic transitions with application to single-molecule enzyme catalysis. J. Chem. Phys. 124:154712, 2006).
IV Gopich, A Szabo. Theory of the energy transfer efficiency and fluorescence lifetime distribution in single-molecule FRET. Proc. Natl. Acad. Sc. 109:7747, 2012.
IV Gopich, A Szabo. Theory of photon statistics in single-molecule Förster resonance energy transfer. J. Chem. Phys. 122:014707, 2005.
SA McKinney, C Joo, T Ha. Analysis of single-molecule FRET trajectories using hidden Markov modeling. Biophys. J., 91:1941, 2006.
H Mazal, G Haran. Single-molecule FRET methods to study the dynamics of proteins at work. Current Op. Biomed. Eng. 12:8, 2019.
B Schuler, and H Hofmann. Single-molecule spectroscopy of protein folding dynamics–expanding scope and timescales. Current Op. Struct. Bio. 23:36, 2013.
Z Kilic, I Sgouralis, S Pressé. Residence Time Analysis of RNA Polymerase Transcription Dynamics: A Bayesian Sticky HMM Approach. Biophys. J. 120:1665, 2021.
I Sgouralis, S Madaan, F Djutanta, R Kha, R Hariadi, S Pressé. A Bayesian nonparametric approach to single molecule Förster resonance energy transfer. J. Phys. Chem. B. 123:675, 2019.
I Sgouralis, S Pressé. ICON: an adaptation of infinite HMMs for time traces with drift. Biophys. J. 112:2117, 2017.
I Sgouralis, S Pressé. An introduction to infinite HMMs for single molecule data analysis. Biophys. J. 112:2021, 2017.