19 Signal Processing Based on Hidden Markov Models for Extracting Small Channel Currents

Vikram Krishnamurthy and Shin-Ho Chung

19.1 Introduction

The measurement of ionic currents flowing through single channels in cell membranes has been made possible by the giga-seal patch-clamp technique (Neher and Sakmann, 1976; Hamill et al., 1981). A tight seal between the rim of the electrode tip and the cell membrane drastically reduces the leakage current and extraneous background noise, enabling the resolution of the discrete changes in conductance that occur when single channels open or close. Although the noise from a small patch is much less than that from a whole-cell membrane, signals of interest are often obscured by the noise. Even if the signal frequently emerges from the noise, low-amplitude events such as small subconductance states can remain below the noise level and there may be little evidence of their presence. It is desirable, therefore, to have a method to measure and characterize not only relatively large ionic currents but also much smaller current fluctuations that are obscured by noise.

Extracting the real signal from a limited set of imperfect measurements is a problem that commonly occurs in scientific experiments and techniques have been developed to overcome this difficulty. Following digitization, a single-channel record consists of a sequence of data points. Each data point contains a mixture of the signal and extraneous noise. The challenge is to remove the noise leaving the biological signal untouched. Some of the methods that have been used to do this are linear and nonlinear filtering (Chung and Kennedy, 1991) and transition detectors (Patlak, 1988, 1993; Tyerman et al., 1992; Queyroy and Verdetti, 1992). Although both linear and nonlinear filtering suppress noise and nonlinear filtering produces little distortion of rapid transitions in underlying signals, neither method utilizes all of the knowledge available about the nature of the signal and interfering noise. Using such information improves the probability of recovering the underlying signal accurately. Broadly speaking, this is the strategy used in the hidden Markov models (HMM) processing technique. The HMM processing technique has been fruitfully utilized in electrical engineering in the disciplines of artificial speech recognition and target tracking in defense systems. The technique was then applied for the analysis of single-channel recordings and to extract small channel currents contaminated by random and deterministic noise (Chung et al., 1990, 1991; Krishnamurthy et al., 1991, 1993; Venkataramanan et al., 1998a,b, 2000). With this signal processing method, the underlying parameters of the HMM could be obtained to a remarkable precision despite the extremely poor signal to noise ratio.

The aim of this chapter is to review the construction and use of HMM for estimating the dynamics of ion channel gating. We first provide a brief intuitive explanation and then a rigorous account of the underlying principles of the processing method. We also outline state-of-the-art results in HMM that are the subject of recent research in mathematical statistics and signal processing in electrical engineering. Some of these techniques are relatively new and not yet known in the biophysics community. These include ideas such as estimating the model order of a HMM, jump Markov linear systems (which is a generalization of HMM to deal with digitally filtered Markov chains and correlated noise), and recursive (online) HMM parameter estimation. We refer the reader to Ephraim and Merhav (2002) for a ~~state of the art~~ review of HMM with a stronger mathematical flavor compared to this chapter.

An HMM is an example of a partially observed stochastic dynamical system. Because opening and closing of an ion channel is random, recordings of single-channel current may be modeled probabilistically as a finite-state, random realization of a Markov chain. Since the underlying ion channel current is corrupted by large amounts of thermal, capacitance and other deterministic noise, the underlying state of the dynamical system is only partially observed. HMM and their generalizations are extremely versatile in capturing the response of complex dynamical systems such as ion channels.

19.2 General Description of the HMM Method

19.2.1 Signal Model and Assumptions

To apply a digital signal processing technique based on HMM to records of single-channel currents contaminated by noise, we first make a plausible guess about the origin of the observation sequence and then construct a signal model. It is assumed that the pure single-channel signal, not contaminated by noise, can be represented as a Markov process with the following characteristics.

In this chapter we deal exclusively with *discrete time* HMM. By discrete time, we mean that the noisy ion channel current is observed at discrete time instants $k = 0, 1, 2, \ldots$ after suitable anti-aliasing filtering and sampling. The advantage of using discrete time HMM is that powerful algorithms can be derived with a fairly elementary background in probability (Papoulis and Pillai, 2002) involving manipulation of conditional probability density functions and Bayes' rule. An analogous theory can be developed for continuous time HMM although the mathematical tools are more difficult since they require the use of stochastic differential equations for a unifying treatment of both discrete and continuous time HMM (see, James et al., 1996).

For each discrete time k, the signal s_k is assumed to be at one of the finite number of states, $q_1, q_2, \ldots, q_N$. Each q_i, where $i = 1, 2, \cdots, N$, is called a state of the process and such a process is called an N-state Markov chain. In the context of channel currents, the Markov state, s_k, represents the true conductance level (or current amplitude) uncontaminated by noise at time k. The observed value at time k, y_k, contains the signal, s_k, random noise w_k, and possibly deterministic interferences d_k, such as sinusoidal interferences from electricity mains and baseline drift. We note here that the meaning of the term *state* differs from that adopted in the Colquhoun–Hawkes model of channel dynamics (Colquhoun and Hawkes, 1981), in which *state* refers to a hypothetical, not directly observable, conformation of the channel macromolecule.

We also assume that the probability of the current being at a particular level (state) at time $k + 1$ depends solely on the state at time k and that the transition probabilities are invariant of time k. In other words, the process is construed as a first-order, homogenous Markov chain. The transition probabilities of passing from state level q_i at time k to state level q_j at time $k + 1$ are expressed as

$$a_{ij} = P(s_{k+1} = q_j | s_k = q_i) \tag{19.1}$$

and form a state transition probability matrix $\mathbf{A} = \{a_{ij}\}$. Thus, $\mathbf{A}$ is an $N \times N$ stochastic matrix, with its diagonal elements denoting the probabilities of remaining in the same state at time $k + 1$ as at time k.

Finally, we define the noise, know also as *the probabilistic function* of the Markov chain or the *symbol probability* as $\mathbf{B} = b_i(y_k)$. It is convenient to assume that the noise is Gaussian. In reality, noise superimposed on single-channel currents is not white but tends to be colored. Its spectral power, instead of being flat, rises steeply at high frequencies. This issue was addressed by use of an autoregressive noise model to represent temporal correlation in the background noise contained in patch-clamp recordings and then the algorithm for handling such correlated and state-dependent excess noise was formulated (Venkataramanan et al., 1998a,b; Venkataramanan and Sigworth, 2002). It was demonstrated that the performance of the algorithm was markedly improved when the background noise was modeled realistically.

19.2.2 Example of a Signal Model

Suppose we know that a record contains four current levels but we are not sure of the exact levels nor the exact signal sequence. We can set up an initial model with the following assumed characteristics. First, we make a reasonable guess, and say that the baseline level is 0 pA and that there are three open states at −1, −2, and −3 pA. Second, we assume that the noise is zero-mean Gaussian, with a standard deviation of, say, 0.25 pA. Third, we provide our initial guesses of transition probabilities from one state level at time k to another state level at time $k + 1$. For a four-state Markov chain, these transition probabilities form a 4×4 transition probability matrix. In the example given here, the first entry of the first row of the matrix represents the

Table 19.1 An example of the signal model.

State levels	$q_1 = 0$ pA $q_2 = -1$ pA $q_3 = -2$ pA $q_4 = -3$ pA
Noise characteristics	Gaussian with $\sigma = 0.25$ pA
Transition matrix	$a_{11}\ a_{12}\ a_{13}\ a_{14}$ $a_{21}\ a_{22}\ a_{23}\ a_{24}$ $a_{31}\ a_{32}\ a_{33}\ a_{34}$ $a_{41}\ a_{42}\ a_{43}\ a_{44}$
Initial probability	$\pi_1 = \pi_2 = \pi_3 = \pi_4 = 0.25$

probability that the process remained in the closed state at time $k + 1$ given that it was closed at time k, whereas the second entry of the first row represents the probability of transiting to the first open level at time $k + 1$ given that it was closed at time k. Similarly, the last entry of the last row represents the probability that the process remained in the fourth level, or the -3 pA level, at time $k + 1$ given that it was at this level at time k. Finally, we stipulate that the probability of the signal being at each one of the four levels at time $k = 1$ is 0.25.

These assumptions can be represented as shown in Table 19.1.

We can also stipulate, if needed, that there is AC hum (50 or 60 Hz and its odd harmonics) embedded in the data but, for simplicity, this is not included in this example. We compactly write all these initial guesses as:

$$\lambda^{(0)} = (\mathbf{q}, \mathbf{A}, \mathbf{B}, \pi), \tag{19.2}$$

our first signal model.

19.2.3 Iterative Algorithm for Estimating HMM Parameters

Here we explain in simple terms how the expectation maximization (EM) algorithm can be used to estimate the HMM parameters. A more rigorous formulation together with other numerical algorithms is given in Section 19.5. The signal model is compared with the data. Essentially, the probabilities of all possible pathways between adjacent data points are calculated, both forward and backward, and the true current levels derived from the highest probabilities. Because the initial parameters we have supplied (e.g., the transition probability matrix and conductance levels) are only guesses, there is going to be a mismatch between the model and data. The model is revised so that it will be more consistent with the data. Using the revised model, the observation sequence is compared again with the new model, and the model is again revised. This iterative process continues, as shown in Fig. 19.1.

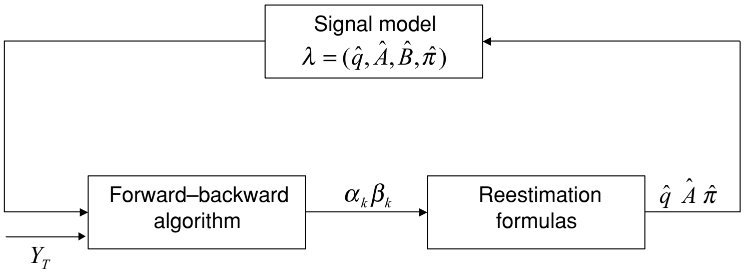
Signal model
$\lambda = (\hat{q}, \hat{A}, \hat{B}, \hat{\pi})$
Forward–backward algorithm
$\alpha_k \beta_k$
Reestimation formulas
$\hat{q}\ \hat{A}\ \hat{\pi}$
Y_T

Fig. 19.1 A block diagram of processing method. On the basis of the initial signal model λ, the observation sequence Y_T is processed, and the forward and backward variables, α_k and β_k are computed for each discrete time k and each Markov state q_i. By using these variables, the parameters of the signal model are revised according to the reestimation formulas. The entire process is repeated many times.

In this iterative process, the HMM processing technique utilizes two mathematical principles. The first is the forward–backward procedure, known also as the *E Step* of the EM algorithm, to be discussed in detail in Section 19.5.2 For each data point k, the algorithm evaluates, using Bayes' rule, the forward and backward variables, α and β. In words, the forward variable $\alpha_k(i)$ is the joint probability of the past and present observations with the present signal in state q_i, given the model λ, and $\beta_k(i)$ is the probability of the future observation given that the present state is q_i and given the model λ. The forward variable is calculated in a forward recursion and the backward variable in a backward recursion.

Then, using the forward and backward variables, the initial model $\lambda^{(0)}$ is reestimated, using the Baum–Welsh reestimation formulas, the *M Step* of the EM algorithm. Estimation formulas stipulate how the model parameters should be revised, given the forward and backward variables. Loosely stated, the *E* step of the EM algorithm maximizes the expectation of the likelihood function if the parameters of the initial model $\lambda^{(0)}$ are replaced by a new model $\lambda^{(1)}$. The same segment of data is now processed with the second signal model, and then the parameters of this model are replaced, again according to the reestimation formulas, in the third signal model $\lambda^{(2)}$. This process is iterated again and again until the difference in the estimates $\lambda^{(n)}$ and $\lambda^{(n+1)}$ in two successive iterations is sufficiently small.

After each iteration, we can compute from the forward variables a numerical value that we call the *likelihood function*—that is, how likely are the model parameters given the data sequence. The closer the model parameters are to the true parameters, the higher the likelihood function. If the likelihood functions were to increase or decrease erratically with successive iterations, this procedure would have been a waste of time. The rationale behind the iterative procedure rests on the elegant reestimation theorem formulated by Baum and colleagues (Baum and Petrie, 1966; Baum et al., 1970; Baum, 1972), which states:

Au: The year "1996" has been changed to "1966" as per the ref. list. Is this OK

$$P(Y_T|\lambda^{n+1} \geq P(Y_T|\lambda^n). \tag{19.3}$$

In words, the probability of the observation sequence Y_T, given the reestimated signal model, is greater than or equal to the probability of Y_T, given the previous signal model. Thus, the signal sequence estimated using the revised model is more consistent with the data than that estimated using the previous signal model. When the iterative procedure converges, then $P(Y_T|\lambda^{n+1}) = P(Y_T|\lambda^n)$, and λ^n is termed the *maximum likelihood estimate* of the HMM. This important theorem, the proof of which is based on Jensen's inequality (Baum and Petrie, 1966), is the core of the HMM processing scheme.

There is a choice of numerical methods of calculating the maximum likelihood estimates, as discussed in Section 19.5.1. One approach is the Newton–Ralphson (NR) algorithm, which, when it converges, does so quadratically and thus rapidly. The EM algorithm, on the other hand, converges linearly, and so convergence can be slow. However, successive iterations with the NR algorithm do not necessarily improve the likelihood function. In contrast, the EM algorithm is simple to implement and has the appealing property that the likelihood function is always improved after each iteration.

19.2.4 Estimating the Number of States

Perhaps the most subjective part of the HMM processing method, like any other parameter estimation scheme, is finding the state dimension—or the number of conductance states in our example—in hidden Markov chain processes. The error in fitting a model to a given set of data decreases with the number of free parameters in the model. Thus, it makes sense, in selecting a model from a set of models with different numbers of parameters, to penalize models having too many parameters. The question of how to penalize HMM for having an excessive number of free parameters is an area of current research (see, for example, Poskitt and Chung, 1996) and one proposed criterion for model-order selection is the compensated likelihood approach (Finesso, 1990; Liu and Narayan, 1994; Rydén, 1995). See Section 19.6.2 for further discussion of this issue.

In practice, however, it is relatively easy to identify the number of states present in the underlying Markov chains. One of the several ways of doing this is by constructing the most likely amplitude histogram. Here we assume that the signal can be represented as a Markov chain with a large number of equally spaced states, say 100 states, and then estimate the most likely signal sequence. We construct an amplitude histogram from the estimated signal sequence under this assumption. After a number of iterations, the maximum likelihood histogram clearly shows prominent peaks, even when no meaningful information can be gleaned from the amplitude histogram of the original record.

Alternatively, we can appeal to the principle of parsimony in deciding the number of conductance states. We measure the goodness of fit by evaluating the likelihood of the model, and weigh this against what is to be gained by increasing the number of parameters, which generally increases the likelihood. Thus, we process the same data segment under the assumption that the underlying signal has a different number of conductance states. If a plot of log likelihood against model order (the number of states) shows a "knee" for a certain model order, we would prefer this model to one of higher order. This approach has been used to determine the number of conductance substates in channel currents activated γ-aminobutyric acd (Gage and Chung, 1994).

19.3 HMM Formulation and Estimation Problems

What follows is a rigorous formulation of the HMM processing techniques. We begin by formalizing the definition of a HMM process.

19.3.1 Definitions

A discrete time HMM process is a stochastic process comprising two ingredients:

1. A *stochastic dynamical system* modeled as an S state discrete time Markov chain s with state space

$$\mathcal{S} = \{1, 2, \ldots, S\}. \tag{19.4}$$

   This Markov chain evolves probabilistically according to the $S \times S$ *transition probability matrix* A. The elements of A are the transition probabilities

$$a_{ij} = \mathbf{P}(s_{k+1} = j | s_k = i), \quad 0 \leq a_{ij} \leq 1, \sum_{j=1}^{S} a_{ij} = 1, \quad i, j \in \{1, \ldots, s\}. \tag{19.5}$$

   The Markov chain s is initialized at time $k = 0$ with

$$\pi_0 = (\pi_0(i),\ i \in \mathcal{S}) = \mathbf{P}(s_0 = i). \tag{19.6}$$

   The Markov chain s models the actual pure, unobserved ion channel current.
2. *Partially observed state*: In an HMM, the Markov chain state (i.e., channel current not contaminated by noise) s is not directly observed. Instead, the observation process y is a noisy corrupted version of s. The observation y is modeled as a random process generated from the conditional probability density (or the probability mass function if y_k is discrete valued)

$$b_i(y_k) = \mathbf{p}(y_k | s_k = i). \tag{19.7}$$

   This conditional probability density is called the *observation likelihood function* in the statistical inference literature. Throughout this chapter we assume that the observation likelihood function $\mathbf{p}$ is parameterized. More precisely, θ denotes the sufficient statistic for the probability density $\mathbf{p}$ by some finite vector θ. For example, as described below, if the observation likelihood b is Gaussian, then θ

comprises the mean and variance since the mean and variance completely specify a Gaussian probability density function.

The above HMM is thus completely modeled by initial probability distribution π_0, the transition probability matrix A, and observation likelihoods b (or equivalently θ). Since we are primarily interested in the evolution and estimation of the HMM over long time scales, the initial distribution π_0 is unimportant. Indeed it can be shown that most HMM forget their initial condition geometrically fast. Most HMM estimation algorithms also forget their initial condition geometrically fast—this is a consequence of "geometric ergodicity" and requires that the transition probability be aperiodic and irreducible (LeGland and Mevel, 2000). To summarize, an HMM is completely parameterized by the model parameter

$$\lambda = (A, \theta). \tag{19.8}$$

In this chapter we are interested in estimating λ given a sequence of N observations of the HMM, where $N > 0$ is a large positive integer denoting the data size (typically several thousand or larger). Denote this N-length HMM observation sequence as

$$Y_N = (y_1, y_2, \ldots, y_N). \tag{19.9}$$

19.3.2 Modeling Ion Channel Current as an HMM

Here we illustrate how to model the noisily observed ion channel current from a patch clamp experiment as an HMM.

A typical trace of the ion channel current measurement from a patch-clamp experiment (after suitable anti-aliasing filtering and sampling) shows that the channel current is a piecewise constant discrete time signal that randomly jumps between two values—zero amperes, which denotes the *closed state* of the channel, and q amperes which denotes the *open* state. Figure 19.2 shows a computer-generated example of a patch-clamp record. To the pure channel current (Fig. 19.2A), noise from various sources is added to mimic the observation sequence, shown in Fig. 19.2B. The *open-state* current level is denoted as q. Sometimes the current recorded from single ion channel dwells on one or more intermediate levels, known as conductance substates. The *pure* ion channel current, uncontaminated by noise, is modeled as the Markov chain s with state space $\mathcal{S} = \{1, 2, 3\}$. These states correspond to the physical ion channel current of

$$\mathbf{q} = (q(1), q(2), q(3))' = \{\mathrm{C}, \mathrm{O}_1, \mathrm{O}_2\} \tag{19.10}$$

corresponding to the physical states of *closed state*, *partially-open state* and *fully-open state*. Subsequently, we will refer to $\mathbf{q}$ as the physical state levels of the Markov chain. When the channel is in the closed state, no currents flow across it. In the open state, the ion channel current has a value of q pA. Figure 19.2 also shows a computer simulated clean ion channel current s.

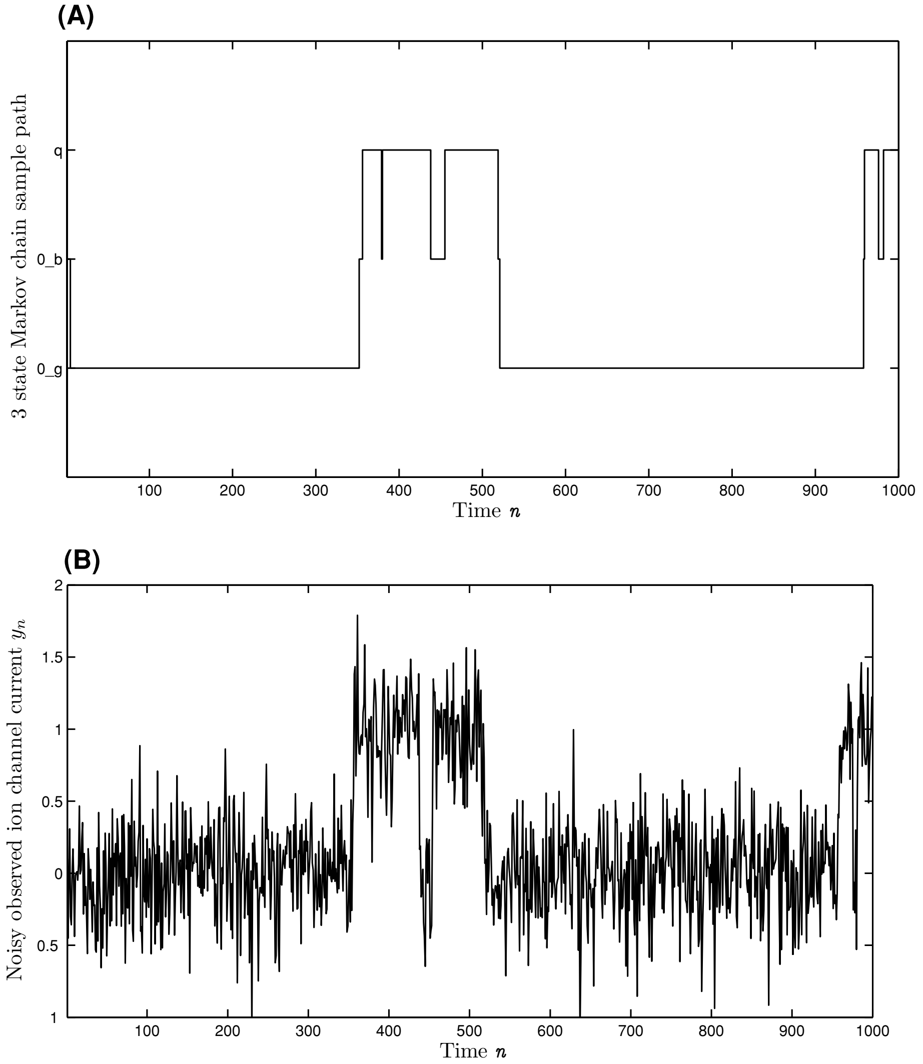

Fig. 19.2 A computer-generated patch-clamp record. To a three-state Markov chain (A), a Gaussian noise was added to mimic a channel current contaminated by amplifier and other noises (B).

The (3×3) transition probability matrix A of the Markov chain s, which governs the probabilistic behavior of the channel current, is given by

$$A = \begin{array}{c|c|c|c|}
 & \text{C} & \text{O}_1 & \text{O}_2 \\
\hline
\text{C} & a_{11} & a_{12} & 0 \\
\hline
\text{O}_1 & a_{21} & a_{22} & a_{23} \\
\hline
\text{O}_2 & 0 & a_{32} & a_{33} \\
\hline
\end{array} \tag{19.11}$$

The elements of A are the transition probabilities $a_{ij} = \mathbf{P}(s_{n+1} = j | s_n = i)$ where $i, j \in \mathcal{S}$ or equivalently the physical state of the ion channel in {C, O_1, O_2}. The zero probabilities in the matrix A, given as an example, state that an ion channel current cannot directly jump from the close state to the fully-open state, or vice versa.

The *observed* noisy ion channel current y from a patch-clamp experiment can be modeled as the Markov chain s corrupted by additive thermal noise with noise variance depending on the state of the ion channel:

$$y_k = q(s_k) + w_k, \quad k = 0, 1, \ldots. \tag{19.12}$$

Here $q(s_k) \in \{\mathrm{C}, \mathrm{O}_1, \mathrm{O}_2\}$, given in (Eq. 19.10), are the physical state levels of the ion channel current, and w_k is a zero mean independent and identically distributed (*iid*) Gaussian scalar noise process with variance σ_w^2. That is, the probability density function of w_k is

$$\mathbf{p}_W(w) = \frac{1}{\sqrt{2\pi}\sigma_w} \exp\left(-\frac{1}{2}\frac{w^2}{\sigma_w^2}\right).$$

In terms of the HMM observation likelihood (Eq. 19.7),

$$b_i(y_k) = \frac{1}{\sqrt{2\pi}\sigma_w} \exp\left(-\frac{1}{2}\frac{(y_k - q(i))^2}{\sigma_w^2}\right). \tag{19.13}$$

In summary, the noisily observed ion channel current y is modeled as a ~~H~~idden Markov model sequence parameterized by the model

$$\lambda = \{A, q, \sigma_w^2\}. \tag{19.14}$$

Our aim is to devise algorithms for estimating λ given an N length sequence of noisy ion channel observations Y_N [defined in (Eq. 19.9)].

Remarks. It is possible to extend the above HMM in several ways.

1. *Higher-order Markov chains*: The model assumption (Eq. 19.5) that s_k depends probabilistically only on its state at the previous time instant, i.e., s_{k-1} means that s is a first-order Markov chain. This assumption can be straightforwardly generalized to higher-order Markov chains with s_k depending probabilistically on the previous Δ time points $s_{k-1}, s_{k-2}, \ldots, s_{k-\Delta}$ where $\Delta \geq 1$. Then define a new first order Markov chain $\bar{s}_k = (s_k, s_{k-1}, \ldots, s_{k-\Delta+1})$ on the state space $\mathcal{S} \times \mathcal{S} \times S$. The HMM processing algorithms presented in this chapter straightforwardly apply to this Markov chain.
2. *Correlated observations*: A critical assumption in constructing an HMM is that the observation process y is "conditionally independent." This means that the

conditional probability density

$$\mathbf{p}(y_k|s_k, \text{past values of } s, y) = \mathbf{p}(y_k|s_k), \tag{19.15}$$

i.e., given s_k, the HMM y_k is independent of the past. This conditional independence holds for the HMM ion channel observations (Eq. 19.12) providing that the corrupting noise w_k is an *iid* process. It is possible to generalize this conditional independence to allow y_k to depend on Δ past values of the observation and state. Such a generalization facilitates dealing with correlated noise. In such a case the observation likelihood of (Eq. 19.7) generalizes to

$$\mathbf{p}(y_k|s_{k-1}, s_{k-2}, \ldots, s_{k-\Delta}, y_{k-1}, y_{k-2}, \ldots, y_{k-\Delta}).$$

Such generalized HMM are widely used in, for example, econometric modeling and fault tolerant systems and are called Markov modulated autoregressive processes or jump Markov autoregressive processes (Krishnamurthy and Rydén, 1998).

3. *State-dependent noise variance*: A generalization of the observation equation (Eq. 19.12) is to model the thermal noise as $w_k(s_k)$, where the noise variance σ_w^2 is a function of the state s_k of the Markov chain. This generalization can easily be incorporated in the HMM.
4. *Additional deterministic interference*: Often the recorded ion channel currents have a deterministic drift and sinusoidal alternating current hum, which corrupts the Markov chain, in addition to the thermal noise. The observation equation (Eq. 19.12) can be formulated as

$$y_k = q(s_k) + w_k + d_k(\theta), \tag{19.16}$$

where $d_k(\theta)$ is the deterministic interference parameterized by some parameter vector θ. For example, in the case of deterministic sinusoidal interference, θ would include the amplitudes, phases and frequencies of the odd harmonics comprising the sinusoidal signal. The observation likelihood (Eq. 19.13) now becomes

$$b_i(y_k) = \frac{1}{\sqrt{2\pi}\sigma_w} \exp\left(-\frac{1}{2}\frac{(y_k - q(i) - d_k(\theta))^2}{\sigma_w^2}\right). \tag{19.17}$$

The HMM parameter is then $\lambda = \{A, q, \theta, \sigma_w^2\}$.
5. *Continuous state space*: The implicit assumption in the above HMM is that the underlying ion channel current s is a finite state process that randomly jumps between a finite number of values according to a Markov chain. It is worthwhile mentioning that there is an equivalently well defined theory for continuous valued states s. For example, if s is represented as a Gaussian continuous-state Markov process, then the Kalman filter and associated parameter estimation algorithms can be used.

19.3.3 Estimation Problems for HMM

Given an N point noisily observed ion channel current sequence Y_N defined in (19.9), there are two HMM estimation problems that are of interest:

Problem 1. *Bayesian state estimation problem.* Compute the optimal state estimate s_k at each time $k = 1, 2, \dots, T$. The term Bayesian reflects the fact that the optimal estimator (defined below) is based on the a posteriori density function of the state and this a posteriori density function is computed via Bayes' rule.

Problem 2. *Maximum likelihood parameter estimation problem*. Compute the model parameter λ that best fits the HMM data Y_N with respect to the maximum likelihood criterion.

In the application to patch-clamp recordings, we are primarily interested in Problem 2 since our ultimate goal is to estimate the model λ that best fits this data. In particular, the transition probability matrix A and state level q yield important information about the kinetics of the ion channel gating. However, Problems 1 and 2 are intimately linked in HMM. Solving Problem 2 involves solving Problem 1 as an intermediate step.

In solving Problem 2, we are looking for the best model λ within the class of models Λ where

$$\Lambda = \{A\},\ q(i) \in [-M, M],\ \sigma_w^2 \in [\sigma_{\min}^2, \sigma_{\max}^2],$$

where A is the transition probability matrices satisfying Eq. 19.5 and M, $\sigma_{\min}^2$ and $\sigma_{\max}^2$ are finite positive constants. Mathematically speaking, for the maximum of a continuous function to exist, the function needs to be defined over a compact set. The above restriction merely restricts the likelihood function to a compact set. Naturally, there are several cost functions that can be used to define the "best" model. The most widely used criterion is the model log likelihood criterion. The log likelihood is more convenient to work with compared to the likelihood. Naturally, since log is a monotone function, maximizing the likelihood is equivalent to maximizing the log likelihood. The log likelihood of a model λ given Y_N is

$$L_N(\lambda) = \log \mathbf{p}(Y_N|\lambda). \tag{19.18}$$

The maximum likelihood estimate (MLE) is defined as the model λ^* that maximizes $L_N(\lambda)$, i.e.,

$$\lambda^* = \arg\max_{\lambda\in\Lambda} L_N(\lambda). \tag{19.19}$$

The log likelihood is the most widely used criterion for estimating HMM because under quite general conditions it has two asymptotic properties that are attractive from a statistician's point of view. First, the MLE is asymptotically consistent, i.e., as $N \to \infty$, the MLE λ_N^* converges to the true model with probability one (Leroux, 1992). Second, the asymptotic error (i.e., as $N \to \infty$) between the true model and the estimate λ_N^* has a Gaussian distribution—this property is called asymptotic normality (Bickel et al., 1998). This further implies that the MLE of an HMM is an asymptotically efficient estimator.

The formulation of the MLE problem is essentially an off-line estimation problem. We collect a batch of observations Y_N and then aim to compute the MLE λ_N^*.

Given that the MLE is a useful parameter estimate, how does one compute the MLE λ_N^* given a block of data Y_N of a HMM? For HMM it is not possible to explicitly solve the maximization problem (Eq. 19.18) and one must resort to a numerical optimization algorithm to compute the MLE. There are two widely used classes of numerical optimization algorithms for computing the MLE, namely the EM algorithm and the NR algorithm. An essential requirement for carrying out any numerical optimization algorithm for optimizing a function (log likelihood in our case) is to be able to evaluate the function at any value. That is, we first need to figure out a way of evaluating the log likelihood $L_N(\lambda)$ for any valid model $\lambda \in \Lambda$. It turns out that evaluating $L_N(\lambda)$ involves solving the state estimation Problem 1.

19.4 Problem 1: Bayesian State Estimation of HMM

At any time $k = 1, 2, \ldots, N$, define the *observation history* of the HMM as

$$Y_k = (y_1, \ldots, y_k). \tag{19.20}$$

The aim is to compute an estimate of the Markov chain s_l at any time $l = 1, 2, \ldots$ given the observation sequence Y_k. More precisely, the aim is to construct a state estimator (function) $\sigma_l(Y_k)$ where $\sigma_l \in \Sigma$ denotes the estimation algorithm and Σ denotes the space of all possible estimation algorithms. By an *optimal state estimator* or *Bayesian state estimator* for an HMM, we shall mean an estimator $\sigma_l^* \in \Sigma$ that minimizes the mean square state estimation error, i.e.,

$$\mathbf{E}\{s_l - \sigma_l^*(Y_k)\}^2 \leq \mathbf{E}\{s_l - \sigma_l(Y_k)\}^2, \quad \sigma_l \in \Sigma.$$

Here $\mathbf{E}\{\cdot\}$ denotes mathematical expectation. Since the metric $\mathbf{E}\{s_l - \sigma_l(Y_k)\}^2$ is simply the variance of the state estimation error, the optimal state estimator is also called the *minimum variance state estimator* or *minimum mean square error (MMSE) state estimator*.

We denote the optimal filtered state estimate as

$$\hat{s}_{l|k} = \sigma_l^*(Y_k). \tag{19.21}$$

The subscript $l|k$ is a reminder that the estimate at time l involves observations up to time k. At first sight it may appear that computing the optimal σ^* to minimize

$\mathbf{E}\{s_k - \sigma(Y_k)\}^2$ is a formidable task. However, by the mean square optimality property of conditional expectations (Jazwinski, 1970) it turns out that the optimal state estimate is

$$\hat{s}_{l|k} = \sigma_l^*(Y_k) = \mathbf{E}\{s_l | Y_k\}. \tag{19.22}$$

In words: *the optimal estimate $\hat{s}_l$ of the state s_l of the HMM at any time l, given the observation history Y_k (Eq. 19.20), is the conditional mean (conditional expectation) of the state s_l given Y_k*. For such a simple result, Eq. 19.22 is quite profound. All of recursive Bayesian estimation, optimal filtering theory merely deals with computing this conditional mean recursively. Indeed the Kalman filter, HMM filter, and particle filter are simply numerical algorithms for computing this conditional mean for different types of partially observed stochastic dynamical systems. The term "Bayesian" reflects the fact that in recursively computing $\hat{s}_{l|k} = \mathbf{E}\{s_l | Y_k\}$, Bayes' rule is used.

Depending on the choice of k and l, there are three types of optimal Bayesian state estimators:

- *Filtering*: If $k = l$, then the estimate $\hat{s}_{k|k}$ is the Bayesian estimate of the state at time k given observations up to time k. Such an estimate is called the filtered state estimate.
- *Prediction*: If $k < l$, then $\hat{s}_{l|k}$ is the Bayesian state estimate at some future time l, given observations up to time k. Such an estimate is called a predicted state estimate.
- *Smoothing*: If $k > l$, then $\hat{s}_{l|k}$ is the Bayesian state estimate of the problem and involves computing the state estimate at some past time l, given the past, present, and future observations up to time k.

In fact to solve Problem 2 (HMM parameter estimation problem), we will require solving the smoothing problem for estimating s_l, $l = 1, 2, \ldots, N$, given the observation sequence Y_N. However, as we show below, the smoothing problem is easily solved once we can solve the filtering problem.

19.4.1 HMM Filtering

The aim here is to derive a real time algorithm for estimating the filtered state estimate $s_{k|k} = \mathbf{E}\{s_K | Y_k\}$. The resulting HMM filter evolves recursively over time k. For notational convenience, we denote $\hat{s}_{k|k} = \hat{s}_k$ for the filtering problem.

Computing the filtered estimate $\hat{s}_k$ can be naturally broken into two steps:

- *Step 1*: Recursively compute the joint probability density $\alpha_k(i)$ defined as

$$\alpha_k(i) = \mathbf{p}(s_k = i, Y_k), \quad k = 1, 2, \ldots, N \tag{19.23}$$

for the HMM Eqs. 19.5 and 19.7. This can be implemented recursively according to the following algorithm:

$$\alpha_{k+1}(j) = b_j(y_{k+1}) \sum_{i=1}^{S} a_{ij}\alpha_k(i), \tag{19.24}$$

initialized by $\alpha_0(i) = \pi_0(i), i = 1, 2, \ldots, S$. The derivation of Eq. 19.24 uses elementary algebra of marginal, conditional probabilities and Bayes' rule as follows:

$$\begin{aligned}\alpha_{k+1}(j) &= \mathbf{p}(s_{k+1} = j, Y_{k+1}) = \mathbf{p}(s_{k+1} = j, y_{k+1}, Y_k)\\ &= \mathbf{p}(y_{k+1}|s_{k+1} = j, Y_k)\mathbf{p}(s_{k+1} = j, Y_k)\\ &= \mathbf{p}(y_{k+1}|s_{k+1} = j, Y_k) \sum_{i=1}^{S} \mathbf{P}(s_{k+1} = j|s_k = i, Y_k)\mathbf{p}(s_k = i, Y_k)\end{aligned} \tag{19.25}$$

Then, using the conditional independence of the observations Eq. 19.15, the Markovian property Eq. 19.5 and noting that $\alpha_k(i) = \mathbf{p}(s_k = i, Y_k)$, directly yields Eq. 19.24.

Recall that $b_j(y_{k+1})$ are the observation likelihoods defined in Eq. 19.7 and are explicitly evaluated in the Gaussian noise case as Eq. 19.13 and deterministic interference case as Eq. 19.17. The above recursion is popularly termed the "Forward algorithm" or the "hidden Markov model state filter."

- *Step 2*: Compute the conditional mean estimate from $\alpha_k(i)$ by summation over all the Markov chain states:

$$\hat{s}_{k+1|k+1} = \mathbf{E}\{s_{k+1}|Y_{k+1}\} = \sum_{i=1}^{S} q(i)\mathbf{p}(s_{k+1}|Y_{k+1}) = \frac{\sum_{i=1}^{S} q(i)\alpha_{k+1}(j)}{\sum_{j=1}^{S} \alpha_{k+1}(j)}. \tag{19.26}$$

It is convenient to express the HMM filter Eq. 19.24 in matrix vector notation. Let $B(y_k)$ denote the $S \times S$ diagonal matrix with (i, i) elements $\mathbf{p}(y_k|s_k = i)$, $i = 1, \ldots, S$. Then the above HMM filter can be conveniently expressed as

$$\alpha_{k+1} = B(y_{k+1})A'\alpha_k, \quad \alpha_0 = \pi_0. \tag{19.27}$$

Here at each time k $\alpha_k = (\alpha_k(1), \ldots, \alpha_k(S))'$ is an S-dimensional column vector with nonnegative elements. Also, applying Step 2 in Eq. 19.26 yields the conditional mean estimate of the HMM filter as

$$\hat{s}_{k+1} = \mathbf{E}\{s_{k+1}|Y_{k+1}\} = \frac{\sum_{i=1}^{S} q(i)\alpha_{k+1}(i)}{\sum_{i=1}^{S}\alpha_{k+1}(i)} = \frac{\alpha_{k+1}}{\mathbf{1}'\alpha_{k+1}} = \frac{B(y_{k+1})A'\alpha_k}{\mathbf{1}'B(y_{k+1})A'\alpha_k}, \tag{19.28}$$

where **1** denotes the S-dimensional vector of ones.

Before one can implement the above equations on a computer, one slight modification is required. It is necessary to scale α_k to prevent numerical underflow. The numerical underflow occurs because α_{k+1} is the product of the transition probabilities and observation likelihood, which are smaller than one in magnitude. Performing the above recursion, thus, leads to all the components of α_k decaying to zero exponentially fast—eventually leading to an underflow error on a computer. Since we are ultimately interested in the normalized filtered density $\mathbf{P}(s_k = i|Y_k)$ and the state estimate $\hat{s}_k = \mathbf{E}\{s_k|Y_k\}$, the underflow problem is straightforwardly remedied by scaling all the elements of α_k by any arbitrary positive number. Since $\hat{s}_k$ involves the ratio of α_k with $\mathbf{1}'\alpha_k$, this scaling factor cancels out in the computation of $\hat{s}_k$ and hence can be chosen arbitrarily. One particularly convenient scaling factor is obtained by normalizing α_k at each iteration. This results in the HMM filter

$$\bar{\alpha}_{k+1} = \frac{B(y_{k+1})A'\bar{\alpha}_k}{\mathbf{1}'B(y_{k+1})A'\bar{\alpha}_k}, \quad \bar{\alpha}_0 = \pi_0. \tag{19.29}$$

$$\hat{s}_{k+1} = \mathbf{q}'\bar{\alpha}_{k+1}, \tag{19.30}$$

where **q** are the physical state levels of the Markov chain as defined in Eq. 19.10. The HMM filter, defined in Eqs. 19.29 and 19.30, is straightforwardly implementable on a computer. The main computational cost is in evaluating $\bar{\alpha}_k$ at each iteration. This requires $O(S^2)$ multiplications at each time k.

19.4.2 HMM Smoothing

So far we have shown how to compute the HMM filtered estimate $\hat{s}_k = \mathbf{E}\{s_k|Y_k\}$. Here we show how to compute the HMM smoothed estimate $\hat{s}_{k|N} = \mathbf{E}\{s_k|Y_N\}$ given a batch of data Y_N.

The smoothing algorithm involves the forward (filtering) recursion given in Eq. 19.23, or equivalently Eq. 19.29, and a backward recursion. Define the smoothed state density as

$$\gamma_{k|N}(i) = \mathbf{P}(s_k = i|Y_N), \quad i = 1, 2, \ldots, S. \tag{19.31}$$

Then similar to Eq. 19.26, the smoothed state estimate is computed as

$$\hat{s}_{k|N} = \mathbf{E}\{s_k|Y_N\} = \sum_{i=1}^{S} q(i)\gamma_{k|N}(i) \tag{19.32}$$

Our task is now to present the forward–backward algorithm for computing $\gamma_{k|N}$. By elementary application of Bayes' rule, this density is computed as

$$\gamma_{k|N}(i) = \frac{\alpha_k(i)\beta_k(i)}{\sum_{i=1}^{S} \alpha_k(i)\beta_k(i)}, \tag{19.33}$$

where α_k is computed via the forward algorithm given in Eq. 19.24, and the backward density $\beta_k(i)$ for $i = 1, 2, \ldots, S$ is defined as

$$\beta_k(i) = \mathbf{p}(Y_{k+1,N}|s_k = i) \tag{19.34}$$

By using a similar argument to Eq. 19.25 it can be shown that β_k can be computed via the backward recursion

$$\beta_k(i) = \sum_{j=1}^{S} \beta_{k+1}(j) a_{ij} b_j(y_{k+1}), \quad k = N, N-1, \ldots 1 \tag{19.35}$$

initialized with $\beta_N(i) = 1, i = 1, 2, \ldots, S$. The above recursion is termed the "Backward algorithm."

In summary, the forward algorithm (Eq. 19.23) together with the backward algorithm (Eq. 19.35) substituted into Eq. 19.33 yields the smoothed density $\gamma_{k|N}(i)$. For N data points, the forward–backward algorithm requires $O(S^2N)$ computations and $O(SN)$ memory.

19.5 Problem 2: HMM Maximum Likelihood Parameter Estimation

Here we present two classes of algorithms for solving Problem 2, i.e., computing the MLE λ^* defined in Eq. 19.19. As mentioned previously, the algorithms use the estimates generated by Problem 1 as an intermediate step.

19.5.1 Newton–Raphson and Related Algorithms

The NR algorithm is a general purpose numerical optimization algorithm that can be used to optimize the likelihood function and thus compute the MLE λ^*. It proceeds iteratively as follows:

- Initialize $\lambda^{(0)} \in \Lambda$.
- For iterations $n = 1, 2, \ldots,$
  - Update parameter estimate as:

$$\lambda^{(n+1)} = \lambda^{(n)} + \left[\nabla_\lambda^2 L(\lambda)\right]^{-1} \nabla_\lambda L(\lambda)\bigg|_{\lambda=\lambda^{(n)}.} \tag{19.36}$$

Here $\nabla_\lambda L(\lambda)$ and $\nabla^2_\lambda L(\lambda)$ denote the first and second derivatives of the likelihood function with respect to the parameter vector λ. The matrix $\nabla^2_\lambda L(\lambda)$ is called the Hessian matrix.

The main advantage of the NR algorithm is that it has a quadratic convergence rate. One of its main disadvantages is that the Hessian $\nabla^2_\lambda L(\lambda^{(n)})$ needs to be evaluated and inverted. Moreover, additional constraints need to be introduced to ensure that the transition probabilities are nonnegative and add up to one, i.e., Eq. 19.5 holds for the estimates obtained from the NR algorithm.

Two variations of the above NR algorithm that avoid this inversion are:

(i) First-order gradient descent: The first-order gradient algorithm is a special case of Eq. 19.36 with the Hessian matrix step size $\left[\nabla^2_\lambda L(\lambda^{(n)})\right]^{-1}$ replaced by a scalar step size of the form $1/n$. Naturally, the convergence rate using a scalar step size is much slower.

(ii) Quasi–Newton–Raphson: The inverse of the Hessian is replaced by a matrix that is easier to compute and invert.

The NR algorithm, Eq. 19.36, requires evaluation of the likelihood function $L(\lambda)$ and its first and second derivatives at $\lambda = \lambda^{(i)}$, $i = 1, \ldots, I$. These can be evaluated in terms of the optimal HMM filter as follows: Consider the HMM filtered density $\alpha_k(i) = \mathbf{p}(s_k = i, Y_k)$ defined in Eq. 19.23 and computed recursively according to Eq. 19.24. At time N (and showing the explicit dependence of α on λ):

$$\alpha^\lambda_N(i) = \mathbf{p}^\lambda(s_N = i, Y_N).$$

The likelihood can then be computed by summing the unnormalized filtered density and time N:

$$L(\lambda) = \mathbf{p}^\lambda(Y_N) = \sum_{i=1}^{S} \alpha^\lambda_N(i). \tag{19.37}$$

Indeed, this is precisely the normalization term in Eq. 19.26.

Consider now evaluating the derivative in Eq. 19.36. Define the sensitivity of the HMM filter as

$$R^\lambda_k(i) = \frac{d}{d\lambda}\alpha^\lambda_k(i), \quad k = 1, \ldots, T.$$

From Eq. 19.37, assuming sufficient regularity to bring the derivative inside the integral

$$\nabla_\lambda L(\lambda) = \sum_{i=1}^{S} R^\lambda_k(i). \tag{19.38}$$

This can be evaluated recursively by differentiating the optimal filter:

$$R^{\lambda}_{k+1}(i) = \big(\nabla_{\lambda}(b^{\lambda}_{j}(y_{k+1})\big)\sum_{j=1}^{S} a^{\lambda}_{ij}\alpha^{\lambda}_{k}(i) + b^{\lambda}_{j}(y_{k+1})\sum_{i=1}^{S}\big(\nabla_{\lambda}a^{\lambda}_{ij}\big)\alpha^{\lambda}_{k}(i)$$
$$+\, b^{\lambda}_{j}(y_{k+1})\sum_{i=1}^{S} a^{\lambda}_{ij}\big)R^{\lambda}_{k}(i).$$

The second-order derivative (Hessian) can be evaluated similarly.

19.5.2 Expectation Maximization Algorithm

The EM algorithm is one of the most widely used numerical methods for computing the ML parameter estimate of a partially observed stochastic dynamical system. The seminal paper by Dempster et al. (1977) formalizes the concept of EM algorithms. Actually, before EM algorithms were formalized in 1977, it was applied in the 1960s by Baum and colleagues (Baum and Petrie, 1966; Baum et al., 1970) to compute the ML parameter estimate of HMM—thus when applied to HMM, the EM algorithm is also called the *Baum–Welch algorithm*.

Similar to the NR algorithm, the EM algorithm is an iterative algorithm. However, instead of directly working on the log likelihood function, the EM algorithm works on an alternative function called the auxiliary or complete likelihood at each iteration. The nice property of the EM algorithm is that by optimizing this auxiliary likelihood at each iteration, the EM algorithm climbs up the surface of the log likelihood, i.e., each iteration yields a model with a better or equal likelihood compared to the previous iteration.

Starting from an initial parameter estimate $\lambda^{(0)}$, the EM algorithm iteratively generates a sequence of estimates $\lambda^{(n)}$, $n = 1, 2, \ldots$ as follows.
Each iteration n consists of two steps:

- *Expectation step*: Evaluate auxiliary (complete) likelihood

$$Q(\lambda^{(n)}, \lambda) = E\{\log \mathbf{p}(X_N, Y_N; \lambda)|Y_N, \lambda^{(n)}\}.$$

The auxiliary likelihood $Q(\lambda^{(n)}, \theta)$ for a HMM can be computed as

$$Q(\lambda^{(n)}, \lambda) = -\frac{N}{2}\ln\sigma_w - \frac{1}{2\sigma_w}\sum_{t=1}^{N}\sum_{i=1}^{S}\mathbf{E}\{(y_k - q(i))^2\}\gamma_k^{\lambda^{(n)}}(i)$$
$$+\sum_{t=1}^{N}\sum_{i=1}^{S}\sum_{j=1}^{S}\gamma_k^{\lambda^{(n)}}(i,j)\log a_{ij},$$

where $\gamma_k^{\lambda^{(n)}}(i) = \mathbf{P}(x_k = q(i)|Y_N; \lambda^{(n)})$ denotes the smoothed state estimate (see, Eq. 19.31) computed using model $\lambda^{(n)}$ via the forward–backward recursions (Eq. 19.33). $\gamma_k^{\lambda^{(n)}}(i, j) = \mathbf{P}(s_k = i, s_{k+1} = j | Y_N; \lambda^{(n)})$ is computed using the forward and backward variables according to the following equation:

$$\gamma_k^{\lambda^{(n)}}(i, j) = \frac{\alpha_k(i) a_{ij} \beta_{k+1}(j) b_j(y_{k+1})}{\sum_i \sum_j \alpha_k(i) a_{ij} \beta_{k+1}(j) b_j(y_{k+1})}. \tag{19.39}$$

- *Maximization step*: Maximize auxiliary (complete) likelihood, i.e, compute

$$\lambda^{(n+1)} = \max_{\lambda} Q(\lambda^{(n)}, \lambda).$$

This maximization is performed by setting $\partial Q / \partial \lambda = 0$, which yields

$$a_{ij} = \frac{\sum_{k=1}^{N} \gamma_k^{\lambda^{(n)}}(i, j)}{\sum_{k=1}^{N} \gamma_k^{\lambda^{(n)}}(i)} = \frac{\mathbf{E}\{\#\text{jumps from } i \text{ to } j | Y_N, \lambda^{(n)}\}}{\mathbf{E}\{\#\text{of visits in } i | Y_N, \lambda^{(n)}\}} \tag{19.40}$$

$$q(i) = \frac{\sum_{k=1}^{N} \gamma_k^{\lambda^{(n)}}(i) y_k}{\sum_{t=1}^{N} \gamma_k^{\lambda^{(n)}}(i)} \tag{19.41}$$

$$\sigma_w^2 = \frac{1}{N} \sum_{k=1}^{N} \sum_{i=1}^{S} \gamma_k^{\lambda^{(n)}}(i) (y_k - q(i))^2. \tag{19.42}$$

19.5.3 Advantages and Disadvantages of EM

The EM Algorithm described above has several advantages compared to the NR algorithm.

- *Monotone property*: The estimate generated in any iteration n is always better or equal to the model in the previous iteration, i.e., $L(\lambda^{(n+1)}) \geq L(\lambda^{(n)})$ with equality holds at a local maximum (see Eq. 19.3). NR does not have monotone property. We refer the reader to Wu (1983) for a rigorous convergence proof of the EM algorithm.
- In many cases, EM is conceptually simpler to apply than NR. For example, the transition probability estimates generated by Eq. 19.40 are automatically nonnegative and add to one. In other words, Eq. 19.5 holds. Similarly, the variance estimate Eq. 19.42 is automatically nonnegative by construction.
- EM is often numerically more robust than NR; inverse of Hessian is not required in EM.
- There are recent variants of the EM that speed up convergence, such as SAGE, AECM (Meng and van Dyk, 1997).

The following are some of the disadvantages of EM Algorithm.

- Typically the convergence of EM can be excruciatingly slow. In comparison, NR often has a quadratic convergence rate, which is much faster than EM. However, with increasing computing power, the slow convergence is usually not a problem for moderately sized HMM.

- NR automatically yields estimates of parameter estimate variance, i.e., the Hessian, whereas EM does not.

19.6 Discussion

In this chapter we have shown how HMMs can be used to model the noisily observed ion channel current. We then described HMM signal processing algorithms for estimating the state and parameters (such as transition probabilities) of the HMM given the noisy observations.

A key advantage of the HMM approach is that it has rigorously provable performance bounds rooted in deep results in mathematical statistics. From a practical point of view, the HMM approach uses all the macroscopic information about the underlying dynamics of the ion channel current to compute the state and parameter estimates: that is, it uses the fact that the underlying ion channel current is piecewise constant, that the gating is approximately Markovian, and that the Markov chain is corrupted by noise with a known distribution. The HMM approach is in contrast to the more ad hoc approach of plotting dwell time histograms, which does not systematically use the above information. In addition, the ML parameter estimate of a HMM is known to be statistically efficient, i.e., it achieves the Cramer–Rao bound (Bickel et al., 1998)—or equivalently for large data lengths N, the resulting ML parameter estimate has the smallest error covariance among the class of asymptotically unbiased parameter estimators.

The conventional approach for processing patch-clamp ion channel current data comprises first eyeballing the noisy ion channel current and rounding off the noisy current to a finite number of values. Then dwell-time histograms are constructed of how long the rounded off process spends in the various states. Such a histogram, plotted on a logarithmic scale, reveals how many exponential functions are needed to fit the observed distributions, or the number of hidden states in the open or closed conformation. The same information can be derived more reliably by adopting the HMM approach. By representing the observed currents as an aggregated Markov chain, the number of hidden states and their transition probability can be directly estimated.

Given the power and elegance of the HMM processing technique, it is not surprising that it has been an active area of research in statistics, electrical engineering, and other areas during the last 15 years. Below we summarize recent developments and extensions.

19.6.1 Recent Developments

The EM algorithm has been the subject of intense research during the last 20 years. We briefly summarize some of the recent developments.

- It has been shown that the EM algorithm can be implemented using a forward step only, i.e., without computation of the backward variable β (Elliott et al., 1995; James et al., 1996). This saves memory requirements, but the computational cost becomes $O(S^4N)$ compared to the forward–backward EM computational cost of $O(S^2N)$ per iteration.
- EM algorithm, like all hill climbing algorithms, converges to a local maximum of the likelihood surface. Thus, one needs to initialize and run EM from several starting points in order to determine the global optimizer of the MLE. During the last 10 years, Markov Chain Monte Carlo methods have been developed that can be combined with the EM algorithm to yield algorithms that converge to the global optimum. For further details, we refer the reader to Liu (2001).

19.6.2 Extensions

Model-order estimation: Throughout this chapter, we have assumed that the model order, i.e., the number of states S of the Markov chain is known. However, in reality there could be several substates when the ion channel is open—and the number of states of the Markov chain may not be known a priori. One way of estimating the number of states of the Markov chain is to introduce a penalized likelihood function

$$L_N(\lambda, S) = L_N(\lambda) + p(S), \tag{19.43}$$

where the penalty function $p(S)$ is a decreasing function of the number of states S of the Markov chain. This function penalizes by choosing a large-dimensional Markov chain. The penalized MLE is then

$$(\lambda^*, S^*) = \arg\max_{\lambda, S} L_N(\lambda, S). \tag{19.44}$$

Conventionally, for model-order estimation, different penalty functions such as the Akaike information criterion (AIC), Bayesian information criterion (BIC), and Minimum description length (MDL) are widely used. Rydén (1995) and Liu and Narayan (1994) present different choices of the penalty function that result in asymptotically consistent model-order estimates.

On-line (recursive) HMM parameter estimation: The algorithms we have proposed so far are off-line. They operate on a batch of data Y_N and assume that there is a fixed underlying model λ that does not change with time. However, in some cases the transition probabilities of the ion channel gating evolve slowly with time. In such cases, it is necessary to devise on-line (recursive) HMM parameter estimation algorithms that operate in real time and adaptively track the slowly time varying parameters of the HMM.

Several such recursive HMM estimators have been proposed (Krishnamurthy and Moore, 1993; Collings et al., 1994; Dey et al., 1994; Krishnamurthy and Yin, 2002). The algorithms are based on applying a stochastic gradient algorithm to either maximize the expected likelihood or the expected prediction error. They are of the form

$$\lambda_{k+1} = \lambda_k + \epsilon \nabla_\lambda e_k(\lambda_k). \tag{19.45}$$

Here, λ_k denotes the HMM parameter estimate at time k, e_k is either the instantaneous log likelihood or prediction error (computed in terms of the forward variable α), ϵ is a step size, and ∇_λ denotes the derivative with respect to the model parameter λ. Choosing ϵ as a small positive constant results in the algorithm tracking slowly time varying parameters. A rigorous weak convergence proof is given by Krishnamurthy and Yin (2002). Such recursive algorithms fall under the ~~G~~eneral class of "stochastic approximation" algorithms and have been the subject of much research during the last 20 years. We refer to Kushner and Yin (1997) for a mathematically rigorous treatment of stochastic approximation algorithms and their convergence.

Jump Markov linear systems: Jump Markov linear systems are a significant generalization of HMM. They permit modeling correlated noise with linear dynamics and also filtered Markov chains. For example, miniature end-plate potentials in a muscle fiber or neuronal cell body recorded with an intracellular electrode comprised of exponentially decaying signals (modeled as a digitally filtered Markov chain) corrupted by noise as follows:

$$\begin{aligned} z_k &= a z_{k-1} + \delta(s_k - s_{k-1}) q(s_k) \\ y_k &= z_k + w_k, \end{aligned}$$

where $\delta(s_k - s_{k-1}) = 1$ if $s_k = s_{k-1}$ and 0 otherwise. Fig. 19.3 shows an example of an exponentially-decaying Markov process embedded in noise, y_k.

For the above model, the conditional independence assumption (Eq. 19.15) does not hold as y_k given s_k depends on the entire history of previous states. The above model is a special case of a jump Markov linear system of the form

$$x_{k+1} = a(s_k)x_k + b(s_k)v_k \tag{19.46}$$

$$y_k = c(s_k)x_k + d(s_k)w_k, \tag{19.47}$$

where x_k is a continuous valued state, s_k is a finite state Markov chain (Eq. 19.5), and v_k and w_k are *iid* noise processes typically assumed to be Gaussian. In such models given the observation sequence $\{y_k\}$, the aim is to construct estimates of the finite state Markov chain s_k and continuous state process x_k. Note that the above dynamical system is a linear system whose parameters $a(s)$, $b(s)$, $c(s)$, $d(s)$ evolve in time according to the realization of the jump Markov chain s—hence the name jump Markov linear system. It is clear that in the special case $a(s) = 1$, $b(s) = 0$, then y_k is a HMM. Also, in the special case s is a 1 state Markov chain

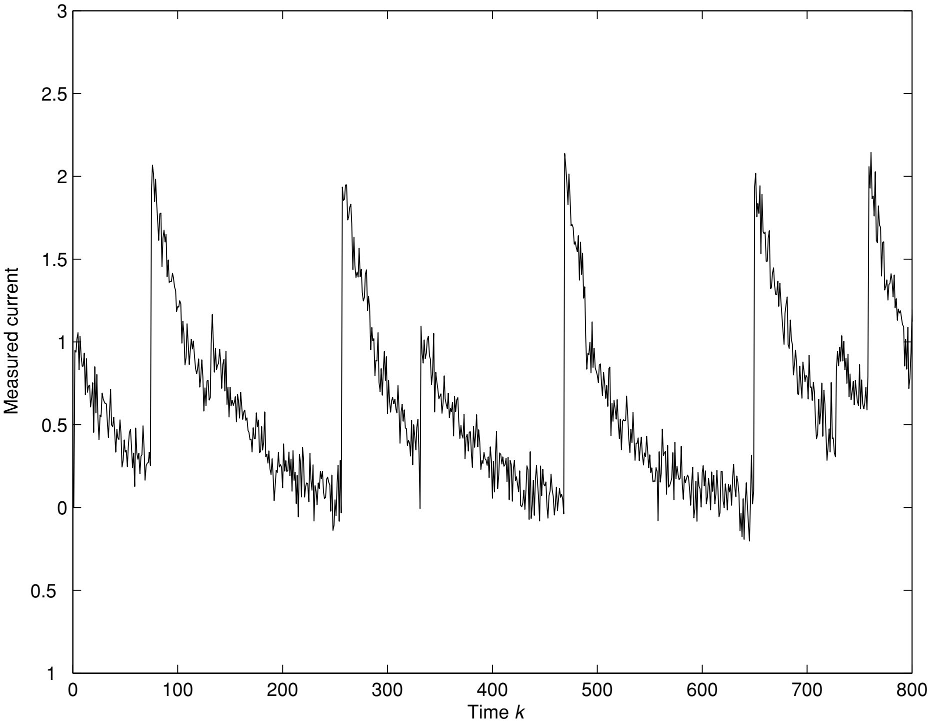

Fig. 19.3 A computer-generated intracellular record. An exponentially-decaying Markov chain, mimicking intracellularly recorded miniature end-plate potentials, is embedded in noise.

(i.e., a constant), the above model becomes a linear state space model. In this special case, if v_k and w_k are Gaussian noise, the conditional mean state estimator of x_k given the observation history is given by the Kalman filter.

Unlike the special cases of the HMM and linear Gaussian state space model, for general jump Markov linear models the optimal Bayesian state estimation problem requires exponential computational complexity (exponential in the data length N). However, there are several high performance non-Bayesian schemes such as maximum a posteriori state estimators which can be used to compute the maximum a posteriori state estimate, rather than the Bayesian conditional mean state estimate (see, Logothetis and Krishnamurthy, 1998). Alternatively, Markov chain Monte Carlo methods can be used to compute approximations of the Bayesian state estimate (see, Ducet et al., 2000, 2001). In particular, Ducet et al. (2001) proposes the so-called "particle filter" which is a sequential Markov chain Monte Carlo algorithm for computing the approximate Bayesian state estimate. Particle filters are widely used in complex Bayesian state estimation problems (Arulampalam et al. 2001).

Automatic control of patch-clamp experiments: Another extension of the basic HMM problem in this chapter is to dynamically control the HMM. For example, it is of interest to dynamically control the patch-clamp experiment to estimate the Nernst potential of the current–voltage curve of an ion channel. The Nernst potential is the applied external potential at which the ion channel current is zero—i.e., it is the applied external potential difference required to maintain electrochemical equilibrium across the ion channel. We refer the reader to Krishnamurthy and Chung (2003) for discrete stochastic optimization-based control algorithms for efficiently estimating the Nernst potential of an ion channel. More generally, the control of HMM with discrete valued observations falls under the class of problems called partially observed Markov decision processes (see for further details, Lovejoy, 1991).

Reference

Arulampalam, S., S. Maskell, N. Gordon, and T. Clapp. 2001. A tutorial on particle filters for on-line non-linear/non-gaussian Bayesian tracking. *IEEE Trans. Signal Proc.* 50:174–188.

Baum, L.E. 1972. An inequality and associated maximization technique in statistical estimation for probabilistic functions of Markov processes. *Inequalities* 3:1–8.

Baum, L.E., and T. Petrie. 1966. Statistical inference for probabilistic functions of finite Markov chain. *Ann. Math. Stat.* 37:1554–1563.

Baum, L.E., T. Petrie, G. Soules, and N. Weiss. 1970. A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains. *Ann. Math. Stat.* 41:164–171.

Bickel, P.J., Y. Ritov, and T. Rydén. 1998. Asymptotic normality of the maximum-likelihood estimator for general hidden Markov models. *Ann. Stat.* 26:1614–1635.

Chung, S.H., and R.A. Kennedy. 1991. Nonlinear forward-backward filtering technique for extracting small biological signals from noise. *J. Neurosci. Methods* 40:71–86.

Chung, S.H., V. Krishnamurthy, and J.B. Moore. 1991. Adaptive processing techniques based on hidden Markov models for characterizing very small channel currents in noise and deterministic interferences. *Phil. Trans. R. Soc. Lond.* B 334:243–284.

Chung, S.H., J.B. Moore, L.G. Xia, L.S. Premkumar, and P. W. Gage. 1990. Characterization of single channel currents using digital signal processing techniques based on hidden Markov models. *Phil. Trans.* ~~*Proc.*~~ *R. Soc. Lond.* B 329:265–285.

Collings, I.B., V. Krishnamurthy, and J.B. Moore. 1994. On-line identification of hidden Markov models via recursive prediction error techniques. *IEEE Trans. Signal Proc.* 42:3535–3539.

Colquhoun, D., and A.G. Hawkes. 1981. On stochastic properties of single ion channels. *Proc. R. Soc. Lond.* B 211:205–235.

Dempster, A.P., N.M. Laird, and D.B. Rubin. 1977. Maximum likelihood from incomplete data via the EM algorithm. *J. R. Stat. Soc.* B 39:1–38.

Dey, S., V. Krishnamurthy, and T. Salmon-Legagneur. 1994. Estimation of Markov modulated time-series via the EM algorithm. *IEEE Signal Proc. Lett.* 1:153–155.

Ducet, A., N. Gordon, and V. Krishnamurthy. 2001. Particle filters for state estimation of jump Markov linear systems. *IEEE Trans. Signal Proc.* 49:613–624.

Ducet, A., A. Logothetis, and V. Krishnamurthy. 2000. Stochastic sampling algorithms for state estimation of jump Markov linear systems. *IEEE Trans. Auto. Control* 45:188–200.

Elliott, R.J., L. Aggoun, and J.B. Moore. 1995. Hidden Markov Models—Estimation and Control. Springer-Verlag, New York.

Ephraim, Y., and N. Merhav. 2002. Hidden Markov processes. *IEEE Trans. Inform. Theory* 48:1518–1569.

Finesso, L. 1990. Consistent estimation of the order of Markov and hidden Markov chains. Ph.D. Dissertation, University of Maryland, MD.

Gage, P.W., and S.H. Chung. 1994. Influence of membrane potential on conductance sublevels of chloride channels activated by GABA. *Proc. R. Soc. Lond.* B 255:167–172.

Hamill, O.P., A. Marty, E. Neher, B. Sakmann, and F.J. Sigworth. 1981. Improved patch-clamp techniques for high-resolution current recording from cells and cell-free membrane patches. *Pflügers Arch.* 391:85–100.

James, M.R., V. Krishnamurthy, and F. LeGland. 1996. Time discretization of continuous-time filters and smoothers for HMM parameter estimation. *IEEE Trans. Inform. Theory* 42:593–605.

Jazwinski, A.H. 1970. Stochastic Processes and Filtering Theory. Academic Press, New Jersey.

Krishnamurthy, V., and S.H. Chung. 2003. Adaptive learning algorithms for Nernst potential and current–voltage curves in nerve cell membrane ion channels. *IEEE Trans. Nanobiosci.* 2:266–278.

Krishnamurthy, V., and J.B. Moore. 1993. On-line estimation of hidden Markov model parameters based on the Kullback-Leibler information measure. *IEEE Trans. Signal Proc.* 41:2557–2573.

Krishnamurthy, V., J.B. Moore, and S.H. Chung. 1991. On hidden fractal model signal processing. *IEEE Trans. Signal Proc.* 24:177–192.

Krishnamurthy, V., J.B. Moore, and S.H. Chung. 1993. Hidden Markov model signal processing in the presence of unknown deterministic interferences. *IEEE Trans. Automatic Control* 38:146–152.

Krishnamurthy, V., and T. Rydén. 1998. Consistent estimation of linear and non-linear autoregressive models with Markov regime. *J. Time Series Anal.* 19:291–308.

Krishnamurthy, V., and G. Yin. 2002. Recursive algorithms for estimation of hidden Markov models and autoregressive models with Markov regime. *IEEE Trans. Inform. Theory* 48:458–476.

Kushner, H.J., and G. Yin. 1997. Stochastic Approximation Algorithms and Applications. Springer-Verlag, New york.

LeGland, F., and L. Mevel. 2000. Exponential forgetting and geometric ergodicity in hidden Markov models. *Math. Controls Signals Syst.* 13:63–93.

Au: Please check the year of publication here.

Leroux, B.G. 1992. Maximum-likelihood estimation for hidden Markov models. *Stochastic Proc. Appl.* 40:127–143.

Liu, J.S. 2001. Monte Carlo Strategies in Scientific Computing. Springer-Verlag, New york.

Liu, C., and P. Narayan. 1994. Order estimation and sequential universal data compression of a hidden Markov source by the method of mixtures. *IEEE Trans. Inform. Theory* 40:1167–1180.

Logothetis, A., and V. Krishnamurthy. 1998. De-interleaving of quantized ~~ar~~ processes with amplitude information. *IEEE Trans. Signal Proc.* 46:1344–1350.

Lovejoy, W.S. 1991. A survey of algorithmic methods for partially observed Markov Decision processes. *Ann. Operations Res.* 28:47–66.

Meng, X.L., and D. van Dyk. 1997. The EM algorithm—an old folk-song sung to a fast new tune. *J. R. Statist. Soc.* B 59:511–567.

Neher, E., and B. Sakmann. 1976. Single-channel currents recorded from membrane of denervated frog muscle fibers. *Nature* 260:799–802.

Papoulis, A., and S. Pillai. 2002. Probability, Random Variables and Stochastic Processes, 4th Ed. McGraw Hill.

Au: Please provide the location of publisher.

Patlak, J.B. 1988. Sodium channel subconductance levels measured with a new variance-mean analysis. *J. Gen. Physiol.* 92:413–430.

Patlak, J.B. 1993. Measuring kinetics of complex single ion channel data using mean-variance histograms. *Biophys. J.* 65:29–42.

Poskitt, D.S., and S.H. Chung. 1996. Markov chain models, time series analysis and extreme value theory. *Adv. Appl. Probab.* 28:405–425.

Queyroy, A., and J. Verdetti. 1992. Cooperative gating of chloride channels subunits in endothelial cells. *Biochim. Biophys. Acta* 1108:159–168.

Rydén, T. 1995. Estimating the order of hidden Markov models. *Statistics* 26:345–354.

Tyerman, D., B.R. Terry, and G.P. Findlay. 1992. Multiple conductances in the large K^+ channel from *Chara corallina* shown by a transient analysis method. *Biophys. J.* 61:736–749.

Venkataramanan, L., R. Kuc, and F.J. Sigworth. 1998a. Identification of hidden Markov models for ion channel currents—Part II: State-dependent excess noise. *IEEE Trans. Signal Proc.* 46:1916–1929.

Venkataramanan, L., R. Kuc, and F.J. Sigworth. 2000. Identification of hidden Markov models for ion channel currents—Part III: Bandlimited, sampled data. *IEEE Trans. Signal Proc.* 48:376–385.

Venkataramanan, L., and F.J. Sigworth. 2002. Applying hidden Markov models to the analysis of single channel activity. *Biophys. J.* 82:1930–1942.

Venkataramanan, L., L.J. Walsh, R. Kuc, and F.J. Sigworth. 1998a. Identification of hidden Markov models for ion channel currents—Part I: Colored background noise. *IEEE Trans. Signal Proc.* 46:1901–1915.

Wu, C. F.J. 1983. On the convergence properties of the EM algorithm. *Ann. Stat.* 11:95–103.