Statistical analysis of neural data:
Discrete-space hidden Markov models

Liam Paninski
Department of Statistics and Center for Theoretical Neuroscience
Columbia University
http://www.stat.columbia.edu/∼liam

April 3, 2009

Contents

1 Discrete-time, homogeneous hidden Markov models (HMMs)

One key application of the "hidden data" framework we developed in the last chapter is to the hidden Markov model (HMM). HMMs have widespread applications in time-series analysis, notably in speech processing, bioinformatics, and control theory, and we will describe a wide variety of applications in neuroscience in the next three chapters. As usual, before diving into the details of these applications, first we must develop some basic analytical tools.

Hidden Markov models are appropriate for describing time-series data that are produced from a system that transitions in a random, Markovian manner between some number of states: these transitions are not observed (hidden), hence the name HMM. In particular, an HMM is described by two random variables at every point in time t: the hidden state q_t and the observed emission y_t (see Fig. 1). For clarity, in this chapter we will begin with the simplest case, in which: 1) time is measured in discrete steps 2) the state variable q_t can take on one of N discrete states $1, \ldots, N$; and 3) the transition and emission probabilities (defined more explicitly below) are independent of time. In this case, we say that $(Q, Y) = (\{q_1, q_2, \ldots, q_t\}, \{y_1, y_2, \ldots, y_t\})$ forms a homogeneous, discrete-time Markov chain. (We will see that each of these assumptions may be relaxed significantly.) We will discuss a number of examples shortly.

More precisely, a homogeneous, discrete-time Markov chain has the following two characteristics: first,

$$p(q_t|q_{t-1}, \ldots, q_1) = p(q_t|q_{t-1}), \tag{1}$$

that is, the future is independent of past given the present (this is a Markov assumption), and second,

$$\alpha_{nm} \equiv p(q_t = m|q_{t-1} = n) = p(q_s = m|q_{s-1} = n), \qquad \forall t, s \in (1, \ldots, T), \tag{2}$$

i.e., the probability of transitioning from state n to state m is constant (homogeneous) as a function of time t. All homogeneous, discrete-time Markov chains can then be completely described by matrices α with the properties that

$$0 \leq \alpha_{nm} \tag{3}$$

and

$$\sum_{m=1}^{N} \alpha_{nm} = 1, \tag{4}$$

since each row is a discrete probability distribution.

In an HMM, the sequence of states, $Q \equiv q_1, \ldots, q_T$, is assumed to evolve only with reference to itself, but not with reference to the sequence of emissions, $Y \equiv y_1, \ldots, y_T$. That is, the next state is independent of the previous emissions given the previous state,

$$p(q_t|q_{t-1}, y_{t-1}, \ldots, y_1) = p(q_t|q_{t-1}) \tag{5}$$

Conversely, the probability distribution of the emission variable does depend on the current state, but does not depend on any previous (or future) state or emission given the current state (another Markov assumption),

$$p(y_t|q_t, q_{t-1}, \ldots, q_1, y_{t-1}, \ldots, y_1) = p(y_t|q_t) \tag{6}$$

[0] Thanks to Sean Escola for his help preparing these notes; the sections on multistate GLM are adapted directly from (Escola and Paninski, 2008).

The traditional HMM framework also assumes that the emission probability distributions, like the transition probability distributions, are time-homogeneous,

$$\eta_{nk} = p(y_t = k|q_t = n) = p(y_s = k|q_s = n), \qquad \forall t, s \in (1, \ldots, T) \tag{7}$$

though again, we will see that the last two assumptions can be relaxed significantly. The η matrices have the same constraints as the α matrices,

$$0 \leq \eta_{nk} \tag{8}$$

and

$$\sum_{k=1}^{K} \eta_{nk} = 1 \tag{9}$$

for a system with K discrete emission classes $1, \ldots, K$.

The dependency structure encapsulated in the Markov and time-homogeneity assumptions (Eqs. 1, 2, 6, and 7) are illustrated in the graphical model shown in Fig. 1. The following factorized distribution over the sequence of states and the sequence of emissions is the full probabilistic description of an HMM:

$$\log p(Q, Y) = \log \left(p(q_1) \prod_{t=2}^{T} p(q_t|q_{t-1}) \prod_{t=1}^{T} p(y_t|q_t) \right) \tag{10}$$

or, more concretely,

$$\log p(Q, Y|\alpha, \eta, \vec{\pi}) = \log \pi_{q_0} + \sum_{t=2}^{T} \log \alpha_{q_{t-1}q_t} + \sum_{t=1}^{T} \log \eta_{q_t y_t} \tag{11}$$

where the $N \times N$ α matrix and the $N \times K$ η matrix are as defined above, and the N-element π vector is the initial state distribution (i.e. $\pi_n \equiv p(q_0 = n)$).

As usual, we would like to infer the parameters of the model α, η, and π (or collectively $\theta = (\alpha, \eta, \pi)$) from the observed data, e.g. by maximizing the log-likelihood. In an HMM the sequence of states Q is unknown (hidden) and must be integrated out of the complete log-likelihood equation to yield the incomplete log-likelihood:

$$L(\theta|Y) = \log \sum_{Q} p(Q, Y|\theta) = \log \sum_{Q} \left(\pi_{q_0} \prod_{t=2}^{T} \alpha_{q_{t-1}q_t} \prod_{t=1}^{T} \eta_{q_t y_t} \right) \tag{12}$$

The sum in Eq. 12 is over all possible paths along the hidden Markov chain during the course of the time-series, and thus calculating the likelihood would appear to have an exponential computational time-complexity ($O(N^T)$, since the number of possible paths Q is N^T). Luckily, it is possible to derive a recursive algorithm for computing this likelihood (allowing us to evaluate the likelihood in linear instead of exponential time); moreover, each iteration of the EM algorithm in this setting may also be performed in linear time. Before we get to the details of these computations, however, let's discuss a few examples.

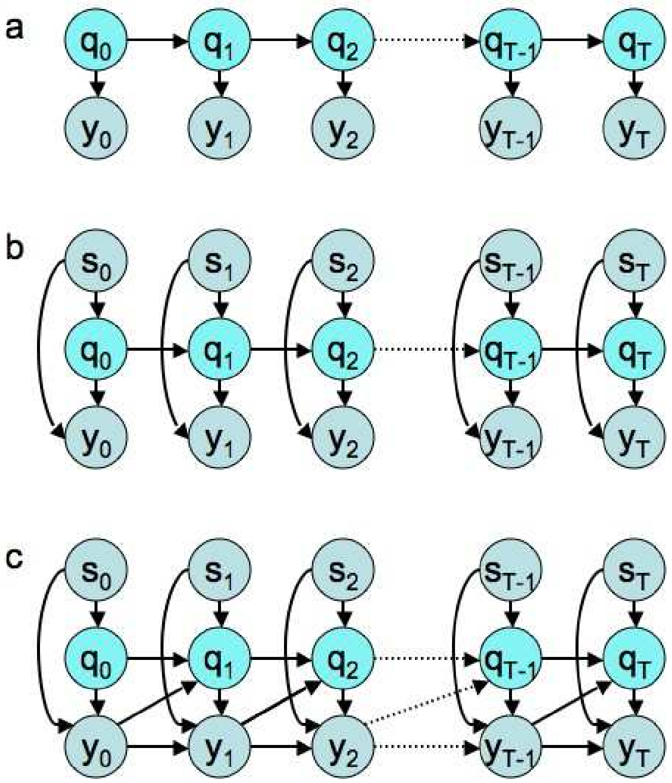

Figure 1: Schematic illustration of several of the HMMs discussed here. In each case, nodes colored aqua are hidden; light blue nodes are observed. We represent the models in directed graphical model form: an arrow from one node x to another node y indicates that x is a "parent" of y: parents and children satisfy the probabilistic relationship $p(x_1, x_2, \ldots x_N) = \prod_i p(x_i | parents\ of\ x_i)$. See (Jordan, 1999) for more details. **A**: Basic HMM structure. **B**: An HMM in which the output y_t and the next hidden state q_{t+1} depend on both q_t and the observed side-information s_t (e.g., this side information could be the observed stimulus, $s(t) = \vec{x}(t)$). **C**: An autoregressive HMM, in which y_t and q_{t+1} depend on the hidden state q_t, the side information s_t, and the past observed output y_{t-1}.

1.1 Example: the switching Poisson model is a simple model for spike trains which flip between a few distinct firing states

In the switching Poisson model, we imagine that the neuron whose spike train we are observing is a Poisson process whose rate λ is flipping randomly between one of N states (we might think of each of these distinct firing rate states as corresponding to different attentive or behavioral states (Rubin, 2003; MacLean et al., 2005; Bezdudnaya et al., 2006))[1]. In particular, if $\lambda(t)$ forms a Markov chain, then we can identify $q_t = \lambda(t)$, and y_t as the observed spike count in some small bin at time t,

$$y_t \sim Poiss(\lambda(t)dt).$$

[1] This kind of point process model — in which the conditional intensity function is itself a random variable — is known as a "doubly stochastic process," or "Cox process" (Cox, 1955; Snyder and Miller, 1991; Sahani, 1999; Moeller and Waagepetersen, 2004).

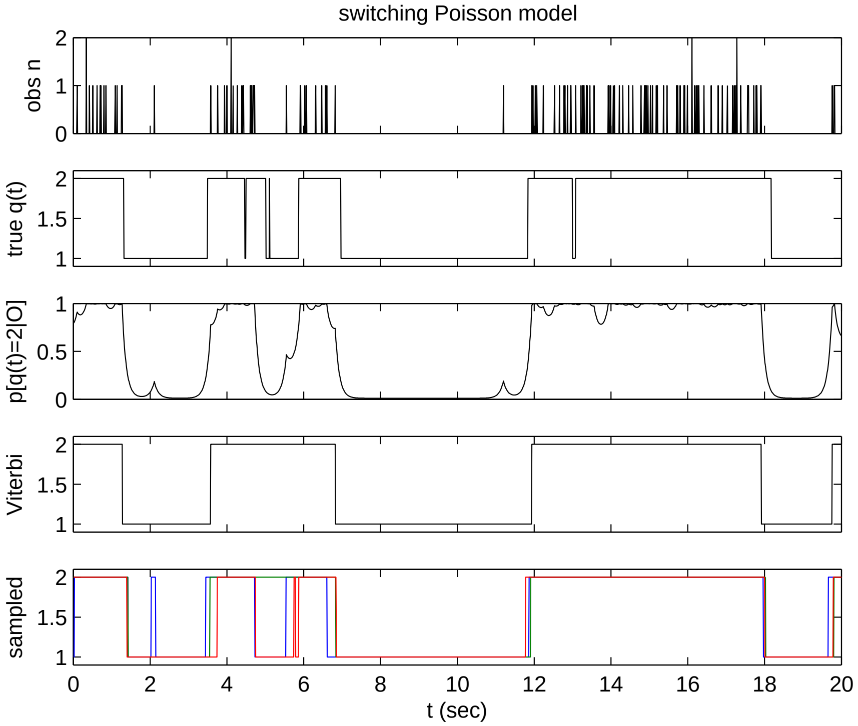

Figure 2: Illustration of the switching Poisson model (simulated data). **A**: The observed discretized spike train $y(t)$. **B**: The true underlying hidden state sequence Q. **C**: The computed forward-backward probabilities $p[q_t = 2|Y]$. **D**: The Viterbi path (most likely path Q given observed sequence Y). **E**: Three i.i.d. samples from $p(Q|Y)$, computed using the forward-backward method. The firing rates here were $\lambda_1 = 0.5$ and $\lambda_2 = 10$ Hz, while the transition rate from $q = 1$ to $q = 2$ was 1 Hz (symmetric transitions were used for simplicity). Note that both the Viterbi path $\arg\max_Q P(Q|Y)$ and the conditional mean path $E(Q|Y)$ track the true Q fairly well; a few quick jumps in the true Q are missed due to the smoothing properties of these algorithms under the low transition rates used here.

This gives us a nice simple HMM that we can then fit to data. See Fig. 2 for an illustration.

A slight generalization of this model was developed by (Gat et al., 1997) (see also (Jones et al., 2007; Kemere et al., 2008) for more recent examples): in this case, the spike trains of multiple neurons were observed simultaneously. Each individual spike train i was modeled as an independent Poisson process with rate $\lambda_i(t)$, and it was assumed that the vector of firing rates $\vec{\lambda}(t)$ was itself a Markov chain (and in particular, this vector could take on only some finite number N of values). (Gat et al., 1997) fit this model to data recorded from the frontal cortex of an awake, behaving monkey and were then able to analyze changes in the firing rate state that were correlated with experimentally-observable behavioral parameters (e.g., reaction time). We will discuss further generalizations of this model below.

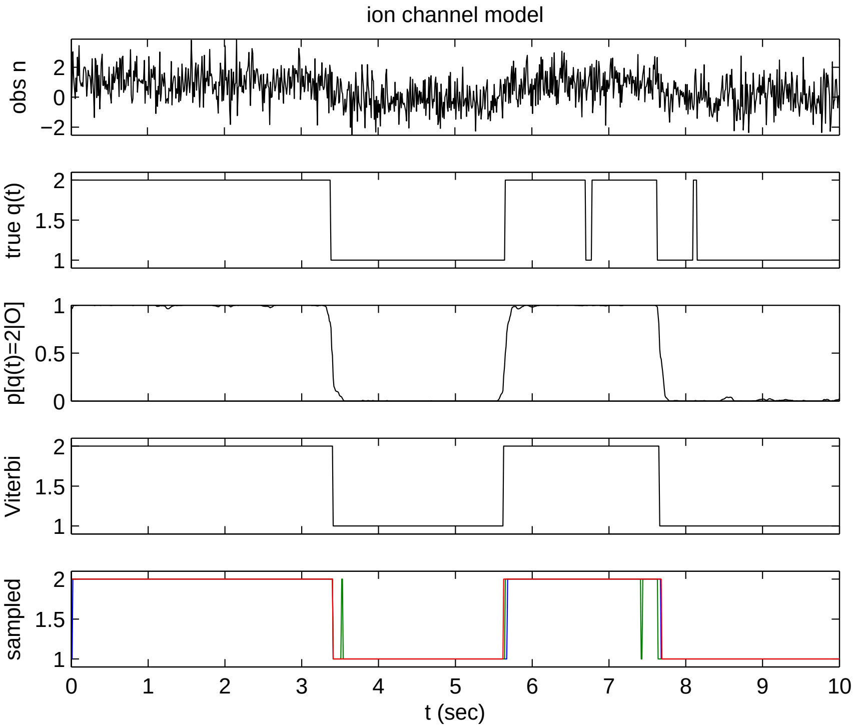

Figure 3: Illustration of the ion channel HMM (simulated data). **A**: The observed conductance sequence $y(t)$. **B**: The true underlying hidden state sequence Q. **C**: The computed forward-backward probabilities $p[q_t = 2|Y]$. **D**: The Viterbi path (most likely path Q given observed sequence Y). **E**: Three i.i.d. samples from $p(Q|Y)$, computed using the forward-backward method. The mean conductances here were $\mu_1 = 0$ and $\mu_2 = 1$, with $\sigma_1 = \sigma_2 = 1$, while the transition rate from $q = 1$ to $q = 2$ was 1 Hz (symmetric transitions were used for simplicity).

1.2 Example: ion channels are often modeled as HMMs

The standard model for ion channels is as follows (Colquhoun and Hawkes, 1982; Hawkes, 2004): the channel has some finite number N of stable configurations, and the channel will flip randomly between these states. The biophysical properties of the channel (e.g., conductivity to a given ion species, or sensitivity to a given drug or voltage perturbation) depend on the configuration state. It is generally very difficult to observe the configuration of the ion channel directly (these shape measurements are typically done via crystallographic techniques, which preclude dynamic measurements of state), but we may be able to make detailed, high-frequency observations of the conductivity of the channel (via single-channel patch clamp techniques (Sakmann and Neher, 1995)). Here it is natural to label the configuration at time t as the hidden state q_t, and the conductivity as the observation y_t. It is often reasonable to model the emission probabilities as

$$y_t|q_t \sim \mathcal{N}(\mu(q_t), \sigma^2(q_t)),$$

i.e., Gaussian with a state-dependent mean and variance (Fredkin and Rice, 1992; de Gunst et al., 2001). See Fig. 3 for an illustration.

1.3 Computing the likelihood in an HMM; forward and backward probabilities

[2] As mentioned above, the key idea for computing the likelihood $p(Y|\theta)$ efficiently is to make use of the special structure of the HMM to perform the necessary marginalizations recursively. In particular,

$$\begin{aligned} p(Y|\theta) &= \sum_Q p(Q, Y|\theta) \\ &= \sum_{q_1=1}^K \sum_{q_2=1}^K \dots \sum_{q_T=1}^K p(q_1) \left(\prod_{t=2}^T p(q_t|q_{t-1}) \right) \left(\prod_{t=1}^T p(y_t|q_t) \right). \end{aligned}$$

The main thing to notice is that we can rearrange the sums here to make the computations much more efficient:

$$p(Y|\theta) = \\ \sum_{q_T} p(y_T|q_T) \sum_{q_{T-1}} p(q_T|q_{T-1}) p(y_{T-1}|q_{T-1}) \sum_{q_{T-2}} \dots \sum_{q_2} p(q_3|q_2) p(y_2|q_2) \sum_{q_1} p(q_2|q_1) p(y_1|q_1) p(q_1); \quad (13)$$

this formula can easily be computed recursively. In particular, if we define the "forward" probabilities as

$$a_n(t) \equiv p(y_1, \dots, y_t, q_t = n|\theta), \quad (14)$$

then we may compute these forward probabilities recursively by

$$a_n(1) = \pi_n \eta_{ny_1} \quad (15)$$

and

$$a_n(t) = \left(\sum_{m=1}^N a_m(t-1) \alpha_{mn} \right) \eta_{ny_t}, \ t > 1, \quad (16)$$

which involves $O(NT)$ computation instead of $O(T^N)$. Marginalizing over the hidden state in the final forward probabilities yields the likelihood,

$$L(\theta|Y) = \log \sum_{n=1}^N a_n(T). \quad (17)$$

(Note that all we really have done here is implement equation (13).)

Alternately, we may introduce the "backward" probabilities

$$b_n(t) \equiv p(y_{t+1}, \dots, y_T|q_t = n, \theta). \quad (18)$$

These can also be computed recursively by

$$b_n(T) = 1 \quad (19)$$

and

$$b_n(t) = \sum_{m=1}^N \alpha_{nm} \eta_{my_{t+1}} b_m(t+1) \quad (20)$$

[2] Much of the following couple sections is directly adapted from (Rabiner, 1989).

which again requires linear time-complexity in T. (We may obtain the likelihood, again, by marginalizing

$$L(\theta|Y) = \log \sum_n p(y_2, \dots, y_T|q_1 = n, \theta) p(q_1 = n|\theta) p(y_1|q_1 = n, \theta) = \log \sum_n b_n(1) \pi_n \eta_{ny_1},$$

but typically we just use the forward probabilities and formula (17) for simplicity.)

1.4 Maximizing and sampling from the state sequence given observed outputs

Before we get to the EM algorithm for optimizing the likelihood of the HMM parameters, it is worth discussing two more important related problems. First, how do we determine the most likely state sequence Q given the observation Y,

$$\arg\max_Q p(Q|Y)?$$

Again, while at first blush it would appear that we might need to search over all possible paths Q to solve this problem (a task whose complexity grows exponentially in T), it turns out to be possible to perform this optimization recursively and quite efficiently (Viterbi, 1967). This recursive method is a special case of the "dynamic programming" technique due to Bellman (Bellman, 1957): since the likelihood $p(Q, Y) = p(q_1, q_2, \dots, q_t, y_1, y_2, \dots y_t)$ depends on Q and Y only through the "local" terms $p(y_t|q_t)$ and $p(q_t|q_{t-1})$, we may solve for the optimal Q by using (cheap) local inductive computations instead of (exponentially expensive) global search. The key idea is that, to solve for the optimal path up to time step $t+1$, we only need to keep track of the state q_t of the optimal path at time t, along with the likelihood of this best path up to time t, but given these two pieces of information we do not need to keep track of the state at previous times $t-1, t-2$, etc.

To make this more explicit, define the intermediate quantity

$$\delta_t(n) = \max_{q_1, q_2, \cdots, q_{t-1}} p(q_1, q_2, \dots, q_{t-1}, q_t = n, y_1, y_2, \dots y_t);$$

this is proportional to the likelihood along the optimal path up until time t which is constrained to end on state i at time t. Now it is straightforward to derive the induction

$$\delta_{t+1}(m) = \left(\max_n \delta_t(n) \alpha_{nm} \right) \eta_{my_{t+1}};$$

i.e., given a list of most likely paths that end at state n at time t, we may easily compute the most likely paths that end at any other arbitrary state m at time $t+1$ (this is the easy local computation we referred to above). If we initialize

$$\delta_1(n) = \pi_n \eta_{ny_1},$$

run the recursion forward to $t = T$, and then maximize the final likelihood value

$$\max_n \delta_T(n),$$

then it is easy to backtrack iteratively from time $t = T$ to $t = 1$, selecting the state which maximized $\delta_n(t)$ given the optimal state at each time $t+1$. This provides us with a path which optimizes the full likelihood $p(Q|Y)$ (the "Viterbi" path, after (Viterbi, 1967), who proposed this inductive algorithm); the total complexity of the algorithm is $O(N^2T)$. Finally, note that the optimization at each time step may have a nonunique solution (and therefore the optimal path may be nonunique), but nevertheless the algorithm is guaranteed to find a globally optimal solution; moreover, it is easy to modify the algorithm to collect the top k paths for some $k > 1$, simply by keeping track of the k best paths at each time t, instead of just the best path.

The above discussion defines optimality in terms of the likelihood $p(Q|Y)$. If instead we use a different definition of optimality — e.g., if we choose the path that maximizes the expectation of the number of correctly predicted states, i.e., the expectation of the cost function,

$$C(Q, Q') \equiv \sum_t 1(q_t \neq q'_t),$$

then we find that the optimal path is instead

$$q_t = \arg\max_n p(q_t = n|Y, \theta).$$

Thus we choose q_t to maximize the marginal probabilities $p(q_t = n|Y, \theta)$ instead of the full joint probability $p(Q|Y, \theta)$, where the marginals $p(q_t = n|Y, \theta)$ may be computed as

$$p(q_t = n|Y, \theta) = \frac{p(q_t = n, Y|\theta)}{p(Y|\theta)} = \frac{p(q_t = n, Y_{1,t}|\theta)p(Y_{t+1,T}|q_t = n, \theta)}{p(Y|\theta)} = \frac{a_n(t)b_n(t)}{p(Y|\theta)}, \quad (21)$$

where we have already discussed how to compute the likelihood $p(Y|\theta) = \sum_{n=1}^N a_n(T)$; the first equality here is by Bayes, the second is by the Markov property of the model, and the third is by definition of $a_n(t)$ and $b_n(t)$.

Of course, these two solutions to the problem of optimizing the path are not the same in general, since the latter solution cares only about individual states, while the Viterbi path cares about sequences (and therefore about the transitions between states as well). For example, it is possible to construct examples for which the optimal individual-sense path is actually impossible (because one of the transitions is illegal, i.e., $\alpha_{nm} = 0$) and therefore least likely in the full-sequence sense.

What if we want to sample from the posterior $p(Q|Y)$ instead of computing the optimal path? This may be done easily using an alternate version of the forward-backward method in which we sweep forward first and then sample backwards (c.f. the approach above, in which the forward and backward steps may be computed completely independently of each other; we will discuss this alternate coupled forward-backward method in much more detail in the context of state-space models in the next chapter). We begin by computing the forward probabilties $a_n(t)$. Then, to construct samples from Q, we recurse backwards: for each desired sample path initialize q_T by drawing a sample from the distribution

$$q_T \sim p(q_T = n|Y) = \frac{a_n(T)}{\sum_n a_n(T)},$$

then for $T > t > 0$, sample backwards,

$$\begin{aligned} q_t &\sim p(q_t | Q_{t+1:T}, Y) \\ &= p(q_t | q_{t+1}, Y) \\ &= \frac{1}{Z} p(q_t, q_{t+1}, Y) \\ &= \frac{1}{Z} p(q_{t+1} | q_t) p(q_t | Y) \\ &= \frac{1}{Z} \alpha_{q_t q_{t+1}} a_{q_t}(t). \end{aligned}$$

Thus sampling on each time step t simply requires that we draw independently from a simple discrete distribution, proportional to the product in the last line. Once this product has been computed, this sampling can be done using standard methods. A nice feature of this method is that the forward probabilities $a_n(t)$ may be precomputed for all t via a single forward step; this only has to be done once, no matter how many sample paths are required. Putting the samples together, for $0 < t \leq T$, clearly gives a sample from $p(Q|Y)$, as desired.

Instead of computing the forwards probabilities and then sampling backwards we could, of course, compute the backwards probabilities and then sample forwards. Again, the key is just to note that we can draw a sample path from $p(Q|Y)$ by recursively sampling from $p(q_{t+1}|Q_{1:t}, Y)$, and the latter can be written in the simple form

$$p(q_{t+1}|Q_{1:t}, Y) = p(q_{t+1}|q_t, Y_{t+1:T}) = \frac{1}{Z} p(q_{t+1}|q_t) p(Y_{t+1:T}|q_{t+1}).$$

This formulation makes explicit a very important point: an HMM conditioned on its outputs remains a Markov chain, but with transition probabilities proportional to $p(q_{t+1}|q_t)p(Y_{t+1:T}|q_{t+1})$ instead of the original $p(q_{t+1}|q_t)$. We will see a number of applications of this insight below.

See Figures 2 and 3 for illustrations of these computations, as applied to the switching Poisson and ion channel models discussed above[3].

[3] Figures 2 and 3 were created using Kevin Murphy's Matlab HMM toolbox, www.cs.ubc.ca/∼murphyk/Software/HMM/hmm.html.

1.5 Expectation-Maximization algorithm for the HMM

Deriving the EM algorithm for the HMM parameters is fairly straightforward[4]. As usual, we compute the expected complete log-likelihood:

$$\begin{aligned}
\langle \log p(Q, Y|\theta) \rangle &= \left\langle \log \pi_{q_0} + \sum_{t=2}^{T} \log \alpha_{q_{t-1}q_t} + \sum_{t=1}^{T} \log \eta_{q_t y_t} \right\rangle_{p(Q|Y,\hat{\theta}^{(i)})} \\
&= \left\langle \log \pi_{q_0} \right\rangle_{p(Q|Y,\hat{\theta}^{(i)})} + \sum_{t=2}^{T} \left\langle \log \alpha_{q_{t-1}q_t} \right\rangle_{p(Q|Y,\hat{\theta}^{(i)})} + \sum_{t=1}^{T} \left\langle \log \eta_{q_t y_t} \right\rangle_{p(Q|Y,\hat{\theta}^{(i)})} \\
&= \left\langle \log \pi_{q_0} \right\rangle_{p(q_0|Y,\hat{\theta}^{(i)})} + \sum_{t=2}^{T} \left\langle \log \alpha_{q_{t-1}q_t} \right\rangle_{p(q_{t-1},q_t|Y,\hat{\theta}^{(i)})} + \sum_{t=1}^{T} \left\langle \log \eta_{q_t y_t} \right\rangle_{p(q_t|Y,\hat{\theta}^{(i)})} \\
&= \sum_{n=1}^{N} p(q_0{=}n|Y, \hat{\theta}^{(i)}) \log \pi_n + \sum_{t=2}^{T} \sum_{n=1}^{N} \sum_{m=1}^{N} p(q_{t-1}{=}n, q_t{=}m|Y, \hat{\theta}^{(i)}) \log \alpha_{nm} \\
&\quad + \sum_{t=1}^{T} \sum_{n=1}^{N} p(q_t{=}n|Y, \hat{\theta}^{(i)}) \log \eta_{n y_t}. \qquad (22)
\end{aligned}$$

From equation (22) it is clear that, to perform the M-step (or, recalling the connection between the gradient of the expected complete log-likelihood and the gradient of the log-marginal likelihood (Salakhutdinov et al., 2003), to compute the gradient $\nabla_\theta \log p(Q|\theta)$), we need only compute the single and pairwise marginal distributions $p(q_t|Y, \hat{\theta}^{(i)})$ and $p(q_{t-1}, q_t|Y, \hat{\theta}^{(i)})$, given respectively by equation (21) and

$$\begin{aligned}
p(q_t = n, q_{t+1} = m|Y, \theta) &= \frac{p(q_t = n, q_{t+1} = m, Y|\theta)}{p(Y|\theta)} \\
&= \frac{p(q_t = n, Y_{1,t}|\theta) p(q_{t+1} = m|q_t = n, \theta) p(Y_{t+1,T}|q_{t+1} = m, \theta)}{p(Y|\theta)} \\
&= \frac{a_n(t) \alpha_{nm} \eta_{m y_{t+1}} b_m(t + 1)}{p(Y|\theta)}, \qquad (23)
\end{aligned}$$

where the derivation follows that of equation (21) closely.

In the simple case of static α and η matrices in a time-homogeneous HMM, it is possible to derive analytic solutions for the next parameter setting $\hat{\theta}_{i+1}$ in each M-step. More generally, ascent techniques can be employed to maximize equation (22), as we will describe at length below. However, the analytic solution for the next parameter setting of the initial state distribution π is useful in the general case. This is found by introducing a Lagrange multiplier

[4] The EM algorithm for HMMs is often called the "Baum-Welch" algorithm, after (Baum et al., 1970).

φ to guarantee that $\sum_n \pi_n = 1$, and then setting the gradient of equation (22) to zero.

$$\begin{aligned}
0 &= \frac{\partial}{\partial \pi_n}\Big(\langle \log p(Q,Y|\theta)\rangle_{p(Q|Y,\hat{\theta}^{(i)})} - \varphi\left(\textstyle\sum_n \pi_n - 1\right)\Big) \\
&= \frac{\partial}{\partial \pi_n}\Bigg(\sum_{n=1}^{N} p(q_0 = n|Y,\hat{\theta}^{(i)})\log \pi_n + \sum_{t=2}^{T}\sum_{n=1}^{N}\sum_{m=1}^{N} p(q_{t-1}=n, q_t = m|Y,\hat{\theta}^{(i)})\log \alpha_{nm} \\
&\qquad + \sum_{t=1}^{T}\sum_{n=1}^{N} p(q_t = n|Y,\hat{\theta}^{(i)})\log \eta_{ny_t} - \varphi\left(\textstyle\sum_n \pi_n - 1\right)\Bigg) \\
&= p(q_1{=}n|Y,\hat{\theta}^{(i)})\frac{1}{\pi_n} - \varphi \\
\implies &\quad \pi_n = \frac{p(q_1{=}n|Y,\hat{\theta}^{(i)})}{\varphi}. \qquad (24)
\end{aligned}$$

Since π and $p(q_1 = n|Y,\hat{\theta}^{(i)})$ must both sum to one, we have the update

$$\hat{\pi}_n^{(i+1)} = p(q_1{=}n|Y,\hat{\theta}^{(i)}). \qquad (25)$$

A similar derivation establishes the following M-step updates for the transition and emission matrices:

$$\hat{\eta}_{nm}^{(i+1)} = \frac{\sum_{t=1}^{T} 1(y_t = m)p(q_t = n|\hat{\theta}^{(i)},Y)}{\sum_{t=1}^{T} p(q_t = n|\hat{\theta}^{(i)},Y)}$$

and

$$\hat{\alpha}_{nm}^{(i+1)} = \frac{\sum_{t=1}^{T-1} p(q_t = n, q_{t+1} = m|\hat{\theta}^{(i)},Y)}{\sum_{t=1}^{T-1} p(q_t = n|\hat{\theta}^{(i)},Y)}.$$

Again, all of the necessary ingredients here have been computed in the E-step, as described above; moreover, as in our previous applications of the EM formalism, the solutions for each of the three parameters π, η, and α have natural interpretations as weighted versions of the fully-observed maximum likelihood estimates; that is, we replace observations of the hidden states q_t in the fully-observed setting with the "pseudo-observations" $p(q_t = n|\theta, Y)$, $p(q_t = n, q_{t+1} = m|\theta, Y)$, and so on. For example, the usual sample size T is replaced by the sum of weights $\sum_{t=1}^{T} p(q_t = n|\hat{\theta}^{(i)}, Y)$.

It is also straightforward to incorporate more special structure into the model in these EM updates. As an example, recall the multineuronal switching Poisson model introduced by (Gat et al., 1997; Jones et al., 2007; Kemere et al., 2008). In this case, the emission probability $p(y_t|q_t)$ was assumed to have a special form:

$$p(y_t|q_t) = p(n_1(t), n_2(t), \ldots n_C(t)|\lambda_1(t), \lambda_2(t), \ldots, \lambda_C(t)) = \prod_{j=1}^{C} \frac{e^{\lambda_j(t)dt}(\lambda_j(t)dt)^{n_j(t)}}{n_j(t)!},$$

where $n_j(t)$ is the spike count from neuron j at time t and C is the total number of observed cells. Here the updates for π and α remain the same, but instead of updating a full matrix η encoding the conditional probabilities of any possible combination of vector spike counts $\vec{n}(t)$ given the state q_t, we only need to compute the much smaller $C \times K$ matrix of rates λ_j given q. If we compute the M-step for these conditional rates, we obtain

$$\left(\hat{\lambda}_j|q = m\right)^{(i+1)} = \frac{1}{dt}\frac{\sum_{t=1}^{T} n_i(t)p(q_t = m|\hat{\theta}_i, Y)}{\sum_{t=1}^{T} p(q_t = m|\hat{\theta}^{(i)}, Y)};$$

thus, as usual in the Poisson setting, the update for λ_j is computed in terms of a weighted expectation of n_j, and once again each sample is weighted by $p(q_t = m|\hat{\theta}^{(i)}, Y)$.

Similarly, in the case of the ion channel model with state-dependent noise, we have the updates

$$\hat{\mu}(m)^{(i+1)} = \frac{\sum_{t=1}^{T} y_t p(q_t = m|\hat{\theta}^{(i)}, Y)}{\sum_{t=1}^{T} p(q_t = m|\hat{\theta}^{(i)}, Y)}$$

and

$$\hat{\sigma}^2(m)^{(i+1)} = \frac{\sum_{t=1}^{T} [y_t - \hat{\mu}(m)]^2 p(q_t = m|\hat{\theta}^{(i)}, Y)}{\sum_{t=1}^{T} p(q_t = m|\hat{\theta}^{(i)}, Y)}$$

for the state-dependent conductance mean and variance, respectively; these are weighted versions of the standard sample mean and variance.

We should emphasize here that, while the M-step has a nice unique solution in each of the cases we have encountered so far, there may in fact be multiple maximizers of the full marginal likelihood $p(Y|\theta)$. In fact, a simple symmetry argument shows that local maxima of the likelihood must exist in all but the simplest HMMs, since the likelihood is invariant with respect to relabelings of the state. Thus initial conditions (or alternately, restrictions on the parameter space to restore identifiability of the model) play an important role in optimizing the likelihood of HMMs.

1.6 Example: including Markovian refractory effects in spike-sorting algorithms

In low-SNR extracellular recordings, simple spike-sorting methods can often lead to inferred spike trains with "refractory violations": interspike intervals less than some assumed minimum absolute refractory period (typically taken to be 1-2 ms). We can deal with these violations systematically by introducing a simple Markov model for neural refractoriness, and then estimating the sequence of spike times by inference in a corresponding HMM model, where the observations y_t are given by the observed voltage, mapped into a convenient feature space. (Recall the discussion of the spike sorting problem in the chapter on the EM algorithm.)

For simplicity, assume that just one neuron is present on the electrode; thus we simply have to perform a binary discrimination task, separating spikes from non-spikes in each time bin. Now we introduce a simple Markov model for refractoriness: if the hidden state $q_t = 1$, then the neuron fires with probability λ, but if q_t is in any other state then we assume the neuron can not fire. (Of course this simple model may be generalized, e.g. by including states in which the firing rate is at some intermediate level between zero and λ (Escola and Paninski, 2008), but we will stick to the simpler model for now.) When the neuron spikes, the state moves from $q_t = 1$ to $q_t = 2$, say, and then follows Markovian dynamics to eventually proceed back to state 1. The transition matrix α here determines the shape of the interspike interval distribution:

$$p(\text{spike at time t}|\text{spike at time 0}) = r(t) * geo(\lambda dt),$$

where $r(t)$ denotes the distribution of "first return" times, at which q_t first returns to the firing state $q_t = 1$ given a spike[5], $*$ denotes discrete convolution, and $geo(q)$ denotes the geometric distribution with parameter q. Here the convolution form for the ISI distribution

[5] The first return time distribution can be easily calculated by taking powers of the transition matrix α; we will discuss related computations in much more depth in the continuous-time setting, in section 4.

follows from the fact that the time at which a spike occurs, given that $q_t = 1$, is independent of the time at which q_t returned to state 1.

Finally, we need to define the emissions probabilities: if we are using a mixture-of-Gaussians classifier, as described previously, then the emissions probabilities are simply Gaussian:

$$p(y_t|q_t = 1) = \mathcal{N}(\mu_1, C_1),$$

and

$$p(y_t|q_t \neq 1) = \mathcal{N}(\mu_0, C_0);$$

i.e., μ_1 and C_1 are the mean and covariance of the voltage features y_t given a spike, and μ_0 and C_0 are the mean and covariance given no spike.

The EM algorithm for this model may be derived following the outline described above. The E step remains unchanged; we simply run the forward-backward algorithm to compute the sufficient statistics involving $p(q_t|Y)$. The M step is only slightly modified. Since there is only one spiking state, we update

$$\hat{\mu}_1^{(i+1)} = \frac{\sum_{t=1}^T y_t p(q_t = 1|\hat{\theta}^{(i)}, Y)}{\sum_{t=1}^T p(q_t = 1|\hat{\theta}^{(i)}, Y)}$$

and

$$\hat{C}_1^{(i+1)} = \frac{\sum_{t=1}^T [y_t - \mu_1][y_t - \mu_1]^T p(q_t = 1|\hat{\theta}^{(i)}, Y)}{\sum_{t=1}^T p(q_t = 1|\hat{\theta}^{(i)}, Y)},$$

as before, and

$$\hat{\mu}_0^{(i+1)} = \frac{\sum_{t=1}^T y_t p(q_t \neq 1|\hat{\theta}^{(i)}, Y)}{\sum_{t=1}^T p(q_t \neq 1|\hat{\theta}^{(i)}, Y)}$$

and

$$\hat{C}_0^{(i+1)} = \frac{\sum_{t=1}^T [y_t - \mu_1][y_t - \mu_1]^T p(q_t \neq 1|\hat{\theta}^{(i)}, Y)}{\sum_{t=1}^T p(q_t \neq 1|\hat{\theta}^{(i)}, Y)}.$$

The updates for the transition matrix α remain unchanged.

See (Herbst et al., 2008) for further details; these authors show that a similar HMM approach can help resolve very noisy spike observations and reduce refractory violations. In particular, if the transition matrix α is chosen to have a "ring" structure, in which q_t transitions deterministically from state i to $i + 1$, then the spike train corresponding to the Viterbi path $\arg\max_Q p(Q|Y)$ will by construction contain no interspike intervals shorter than K time steps. In addition, extensions to multiple neurons are possible, but make the method much more computationally difficult, and some approximations become necessary in practice; again, see (Herbst et al., 2008) for details.

2 Multistate generalized linear models for spike trains: extending the switching Poisson model

Clearly the switching Poisson model we introduced above is oversimplified: neither stimulus-dependent nor spike history-dependent terms are included in the model. Thus it is natural to ask how we might generalize the switching Poisson model to make it more useful for the analysis of real neural data. In particular, we would like to combine the strengths of the GLM approach with those of the HMM approach, if possible. For example, neurons might plausibly have not only state-dependent baseline firing rates (as in the switching Poisson model) but also state-dependent receptive fields (Bezdudnaya et al., 2006) and state-dependent spike train properties (for example, tonic and burst modes of cells in the thalamus (Sherman, 2001) and up-and-down states in the cortex (MacLean et al., 2005)). Moreover, it is easy to think of models in which the transition probabilities between states are themselves stimulus (and spike-history) dependent. It turns out to be fairly straightforward to incorporate all of these effects in a single, tractable GLM-HMM model.

To get this to work, we need to make three specific generalizations of the basic homogeneous HMM we introduced above. In particular, we want to allow the emission and transition probabilities to be: 1) time-dependent; 2) dependent on the stimulus $\vec{x}(t)$; 3) autoregressive, in the sense that the emissions and transitions no longer just depend on the value of the state q at time t, but also on past observed values of $y_s, s < t$. We address each of these issues in turn.

2.1 Extension to the case that firing and transition rates are time-dependent

The extension to the time-dependent case is trivial for the emission matrix; as before, we model y_t as a point process with conditional intensity function $\lambda_n(t)$, where again n indexes the state.

The extension for the transition matrix α is slightly more complicated: whereas for the spike count emissions it is perfectly reasonable to allow y_t to be larger than one (if dt is large enough), for the transitions it is inconvenient to allow multiple transitions to occur from state n to state m during a single time-step t. To sidestep this issue, we introduce the following model,

$$\alpha_{nm}(t) = \begin{cases} \dfrac{\lambda'_{nm}(t)dt}{1 + \sum_{l \neq n} \lambda'_{nl}(t)dt} & m \neq n \\ \\ \dfrac{1}{1 + \sum_{l \neq n} \lambda'_{nl}(t)dt} & m = n, \end{cases} \tag{26}$$

where $\lambda'_{nm}(t)$ is the instantaneous "pseudo-rate" of transitioning from state n to state m. This definition of α is convenient because it restricts transitions to at most one per time-step and does not violate Eq. 4; as we will see below, this form of α also simplifies computations in the continuous-time limit $dt \to 0$.

2.2 Including stimulus and spike history dependence

One natural model for the firing rates $\lambda_n(t)$ and the transition rates $\lambda'_{nm}(t)$ is the GLM

$$\lambda'_{nm}(t) = g\Big(\vec{k}'_{nm} \cdot \vec{x}(t) + b'_{nm}\Big) \tag{27}$$

and

$$\lambda_n(t) = f\Big(\vec{k}_n \cdot \vec{x}(t) + b_n\Big) \tag{28}$$

where as usual $\vec{x}(t)$ denotes the spatiotemporal stimulus at time t, $\vec{k}'_{nm}$ and $\vec{k}_n$ are weight vectors that describe the neuron's preferences in stimulus space for transitioning and firing respectively, b represents a constant (scalar) offset term, and $g(.)$ and $f(.)$ are nonlinear rate functions mapping real scalar inputs to non-negative scalar outputs. The $\vec{k}_n$ stimulus filters for firing are the K linear "receptive fields" associated with each of the K states of the neuron. In the degenerate case where $K = 1$, the model reduces to a standard linear-nonlinear-Poisson (LNP) model, and $\vec{k}_1$ becomes the canonical receptive field. The $\vec{k}'_{nm}$ stimulus filters for transitioning are, by analogy, "receptive fields" for transitioning, and since there are $K(K-1)$ of these, there are K^2 total transition and firing stimulus filters describing the full model.

The manner in which spike history dependence enters into the rate equations is, as in the standard GLM setting, mathematically equivalent to that of the stimulus dependence. For convenience, define $\vec{\gamma}(t)$ as the vector of the spike-counts for each of the J time-steps prior to t,

$$\vec{\gamma}(t) \equiv (y_{t-1}, \ldots, y_{t-J})^T \tag{29}$$

Then the transition and firing rate equations are modified by additional linear terms as

$$\lambda'_{nm}(t) = g\Big(b_{nm} + \vec{k}'_{nm} \cdot \vec{x}(t) + \vec{h}'_{nm} \cdot \vec{\gamma}(t)\Big) \tag{30}$$

and

$$\lambda_n(t) = f\Big(b_n + \vec{k}_n \cdot \vec{x}(t) + \vec{h}_n \cdot \vec{\gamma}(t)\Big) \tag{31}$$

where $\vec{h}'_{nm}$ and $\vec{h}_n$ are weight vectors that describe the neuron's preferred spike-history patterns for transitioning and firing, respectively. (As usual, by forming a suitable design matrix X we may treat the stimulus, spike history, and constant offset parameters in a unified way. Therefore we will only treat the case of fitting $\vec{k}$ below; the case of fitting $(\vec{k}, \vec{h}, b)$ simultaneously may be handled in exactly the same way.)

2.3 Example: computing the spike-triggered average in the multistate model

Certain computations remain fairly straightforward in this multistate model. As an example, let's take a look at the spike-triggered average in this model. For simplicity, let's assume that: 1) the stimuli $\vec{x}_t$ are i.i.d. (no temporal stimulus correlations); 2) the stimulus filters are instantaneous ($\vec{k}_n$ and $\vec{k}'_{nm}$ do not implement any temporal filtering); and 3) all spike-history terms are set to zero (e.g., given the hidden state sequence q_t, spikes are generated by an inhomogeneous Poisson process).

Now we want to compute the expectation of the stimulus $\vec{x}_t$ at time t given the event s_u: a spike occurs at time u, with $u > t$. We compute directly:

$$\begin{aligned} E(\vec{x}_t|s_u) &= \frac{1}{p(s_u)} \int p(\vec{x}_t, s_u)\vec{x}_t d\vec{x}_t \\ &= \frac{1}{p(s_u)} \int \sum_{q_t, q_{t+1}} p(\vec{x}_t, s_u, q_t, q_{t+1})\vec{x}_t d\vec{x}_t \\ &= \frac{1}{p(s_u)} \int \sum_{q_t, q_{t+1}} p(q_{t+1}|q_t, \vec{x}_t)p(\vec{x}_t)p(q_t)p(s_u|q_{t+1})\vec{x}_t d\vec{x}_t. \end{aligned} \tag{32}$$

All of the terms in the last line may be computed readily. In particular, we may compute $p(s_u|q_{t+1})$ via the backwards recursion, if we use the fact that the spike sequence s_t and the state sequence q_t form an HMM after marginalizing over the i.i.d. stimulus sequence $\vec{x}_t$. We just need to compute the marginal emission probability

$$p(s_t|q_t) = \int p(s_t, \vec{x}_t|q_t)d\vec{x}_t = \int p(s_t|q_t, \vec{x}_t)p(\vec{x}_t)d\vec{x}_t$$

and the marginal transition probabilities

$$p(q_{t+1}|q_t) = \int p(q_{t+1}, \vec{x}_t|q_t)d\vec{x}_t = \int p(q_{t+1}|q_t, \vec{x}_t)p(\vec{x}_t)d\vec{x}_t,$$

which may be reduced to a series of one-dimensional numerical integrals, in general. Finally, the stationary distributions $p(q_t)$ and $p(s_u)$ can be computed directly from the top eigenvector of the marginal transition matrix $p(q_{t+1}|q_t)$, and the terms $p(\vec{x}_t) = p(\vec{x})$ and $p(q_{t+1}|q_t, \vec{x}_t)$ are both given; the integral and sums in equation (32) can again be reduced to a series of one-dimensional numerical integrals.

Interestingly, we find that in general $E(\vec{x}_t|s_u)$ will be an exponentially-decaying function of the time delay $u - t$ (due to the recursive definition of the backwards density $p(s_u|q_{t+1})$), even though the stimulus filters $\{\vec{k}\}$ do not contribute any temporal filtering themselves.

2.4 Parameter estimation via EM

Again, we may write down the expected complete log-likelihood,

$$\begin{aligned}\langle \log p(Q, Y|\theta)\rangle &= \sum_{t=2}^{T}\sum_{n=1}^{K}\sum_{m=1}^{K} p(q_{t-1}{=}n, q_t{=}m|Y, \hat{\theta}^{(i)}) \log \alpha_{nm} + \sum_{t=1}^{T}\sum_{n=1}^{K} p(q_t{=}n|Y, \hat{\theta}^{(i)}) \log \eta_{ny_t} \\ &+ \sum_{n=1}^{K} p(q_1{=}n|Y, \hat{\theta}^{(i)}) \log \pi_n. \end{aligned} \tag{33}$$

The E-step, as in the vanilla HMM case, corresponds to running the forward-backward algorithm to compute the single and pairwise marginals $p(q_t|Y, \hat{\theta}^{(i)})$ and $p(q_{t-1}, q_t|Y, \hat{\theta}^{(i)})$: the forward-backward algorithm is completely unchanged here once we substitute

$$p\left(y_t|q_t, q_{t-1}, \ldots, q_1, y_{t-1}, \ldots, y_1, \vec{x}(t)\right) = p\left(y_t|q_t, \vec{x}(t), \vec{\gamma}(t)\right)$$

instead of just $p(y_t|q_t)$, and

$$p\left(q_{t+1}|q_t, y_t, \ldots, y_1, \vec{x}(t)\right) = p\left(q_{t+1}|q_t, \vec{x}(t), \vec{\gamma}(t)\right),$$

instead of just $p(q_{t+1}|q_t)$.

For the M-step, we may address each of the three terms in equation (33) individually, since to update α we need only optimize the first term, and to update η we need only optimize the second term, etc. The last term is the easiest: to update π, we simply employ equation (25) again.

The second term is also fairly straightforward:

$$\begin{aligned}\sum_{t=1}^{T}\sum_{n=1}^{K} p(q_t{=}n|Y, \hat{\theta}^{(i)}) \log \eta_{ny_t} &= \sum_{t=1}^{T}\sum_{n=1}^{K} p(q_t{=}n|Y, \hat{\theta}^{(i)}) \log \frac{\left(f\left(\vec{k}_n \cdot \vec{x}(t)\right)dt\right)^{y_t} e^{-f\left(\vec{k}_n \cdot \vec{x}(t)\right)dt}}{y_t!} \\ &= \sum_{t=1}^{T}\sum_{n=1}^{K} p(q_t{=}n|Y, \hat{\theta}^{(i)}) \Big(y_t \log f\left(\vec{k}_n \cdot \vec{x}(t)\right) - f\left(\vec{k}_n \cdot \vec{x}(t)\right)dt\Big) + const.\end{aligned}$$

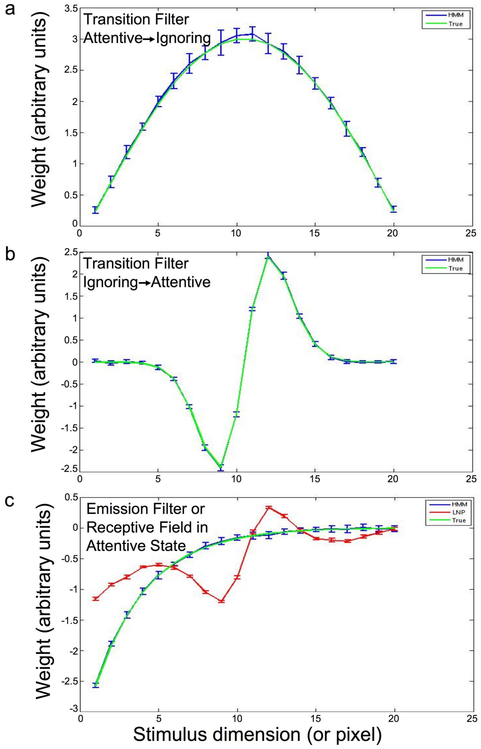

Figure 4: Example fits of the multistate GLM parameters to simulated data. Simulated neuron is a two-state GLM; in one state, the cell was "attentive" (i.e., the spiking probability depended on the stimulus), and in the other state, the filter $\vec{k}$ was zero and therefore the cell ignored the stimulus completely. **A-B**: True and inferred transition filters. **C**: True and inferred emission filter in the attentive state. The estimated filter $\vec{k}$ for a standard (single-state) GLM is shown for comparison; note that this estimated filter is a nonlinear combination of the stimulus and emissions filters, and leads to poor predictive performance overall. 1000 seconds of data (white noise stimuli) were used for the fitting, with firing rates of about $20-30$ Hz; ten experiments were performed to assess the variability of the fits (mean ± s.d. shown).

This is just a weighted version of our standard GLM point process loglikelihood, and may be optimized easily via gradient ascent under the usual convexity and log-concavity conditions on $f(.)$, since the weights $p(q_t = n|Y, \hat{\theta}^{(i)})$ are nonnegative. Conveniently, the optimizations for each $\vec{k}_n$ may be performed in parallel.

The first term is a little more novel: $\sum_{t=2}^{T}\sum_{n=1}^{K}\sum_{m=1}^{K} p(q_{t-1}{=}n,q_t{=}m|Y,\hat{\theta}^{(i)})\log\alpha_{nm}$

$$= \sum_{t=2}^{T}\sum_{n=1}^{K}\left(\begin{array}{l} \sum_{m\neq n} p(q_{t-1}{=}n,q_t{=}m|Y,\hat{\theta}^{(i)})\log\dfrac{g\left(\vec{k}'_{nm}\cdot\vec{x}(t)\right)dt}{1+\sum_{l\neq n} g\left(\vec{k}'_{nl}\cdot\vec{x}(t)\right)dt} \\ +\, p(q_{t-1}{=}n,q_t{=}n|Y,\hat{\theta}^{(i)})\log\dfrac{1}{1+\sum_{l\neq n} g\left(\vec{k}'_{nl}\cdot\vec{x}(t)\right)dt} \end{array}\right)$$

$$\sim \sum_{t=2}^{T}\sum_{n=1}^{K}\left(\begin{array}{l} \sum_{m\neq n} p(q_{t-1}{=}n,q_t{=}m|Y,\hat{\theta}^{(i)})\log g\left(\vec{k}'_{nm}\cdot\vec{x}(t)\right) \\ -\, p(q_{t-1}{=}n|Y,\hat{\theta}^{(i)})\log\left(1+\sum_{l\neq n} g\left(\vec{k}'_{nl}\cdot\vec{x}(t)\right)dt\right) \end{array}\right),$$

since $\sum_m p(q_{t-1}=n, q_t=m|Y,\hat{\theta}^{(i)}) = p(q_{t-1}=n|Y,\hat{\theta}^{(i)})$. Here we need to impose stronger conditions on $g(.)$ to ensure concavity of this objective function with respect to the parameters $\vec{k}'_{nm}$: for example, it is sufficient that $g(.) = \exp(.)$ (Escola and Paninski, 2008).

Just as in the homogeneous, stimulus-independent HMM case, while the M-step has a well-defined global maximum, the likelihood $p(Y|\theta)$ itself may have local maxima. Thus choosing an initialization for θ is more important than in the simpler GLM setting. See Fig. 4 for example fits estimated from simulated data, and (Escola and Paninski, 2008) for further analyses and discussion.

3 Non-Markovian models for identifying "up" and "down" states in spike trains

Fill in later; (Escola and Paninski, 2008); also Kass and Chen et al

4 Continuous-time Markov chains

Above we discussed how to compute the probabilities of various events related to Markov chains in discrete time. However, in certain circumstances it is advantageous or more natural to work in continuous time instead. For example, as we will see, certain computations may be done analytically in continuous time but only numerically in discrete time. In addition, certain formulas take on more intuitive forms in the continuous limit. Thus in this section we will develop some of the continuous-time analogs for the methods we presented in the previous chapter.

Before we dive into the topic of continuous-time hidden Markov chains, it is useful to review the theory of continuous-time, fully-observed Markov chains. As we will see, there are a number of close connections between continuous-time Markov chains and point processes: the Poisson process is in fact a canonical example of a continuous-time Markov chain. In addition, the intuition and tools we develop in the discrete-space setting will be useful in the continuous-space applications (notably the noisy integrate-and-fire model and related Cox process) we'll discuss later.

In discrete time, the transition matrix α_{nm} plays the key role: in particular, we know that to obtain the probability distribution on states at some time t steps in the future, we need only apply the transition matrix t times to the vector describing our current probability distribution on states. In continuous time, it doesn't make as much sense to talk about the probability of being in state m one "time step" after being in state n, but we can certainly talk about rates of change, for example the rate of probability mass flowing from state n to state m at time t.

A physically reasonable Markov chain $X(t)$ will satisfy

$$P(X(t) = m|X(0) = n) = \begin{cases} 1 + A_{nn}t + o(t) & n = m \\ A_{nm}t + o(t) & i \neq m \end{cases}, \quad t \to 0,$$

with $A_{nn} \leq 0$ and $A_{nm} \geq 0, n \neq m$. That is, the probability that system will remain in state n some short time t after being observed in state n will approach 1 as t becomes small. So we will focus on the *rate* A_{nn} at which this probability goes to one, and the corresponding rates A_{nm} at which the probability that the system will be in state m at time t goes to zero.

In particular, we may form derivatives as usual and define

$$A_{nm} = \lim_{t \searrow 0} \frac{P(X(t) = m|X(0) = n)}{t};$$

we call this matrix the "generator" of the process $X(t)$. Here A_{nm} describes the rate at which $X(t)$ jumps from state n to state m. Clearly we must have

$$\sum_m A_{nm} = 0 \quad \forall n,$$

since otherwise mass would be created (i.e., $P(X(t) = m|X(0) = n)$ would not sum to one); thus

$$A_{nn} = -\sum_{m:n \neq m} A_{nm}.$$

In the case that state n is absorbing, for example — i.e., $X(t)$ might jump into state n but then never leaves — $A_{nm} = 0$ for all m.

As a simple example, let's look once again at the homogeneous Poisson process $N(t)$ with rate λ. (Recall that $N(t)$ denotes the counting process: the number of spikes falling before time t.) We have that

$$P(N(t) = m|N(0) = n) = \begin{cases} \exp(\lambda t)(\lambda t)^{m-n}/(m-n)! & n \leq m \\ 0 & n > m \end{cases};$$

we may easily calculate that the generator for this case is

$$A_{nm} = \begin{cases} -\lambda & n = m \\ \lambda & n = m - 1 \\ 0 & \text{otherwise.} \end{cases}$$

(Note that the state space of the Poisson process, the nonnegative integers, is infinite. In general, Markov chains with infinite state spaces do present some mathematical subtleties (Karlin and Taylor, 1981; Norris, 2004), but the examples we will be looking at will pose no major analytical problems.)

Recall that the times between changes of state (jumps) in the Poisson process were exponentially distributed, with rate λ. In the more general case, an identical argument shows that the waiting time for a jump away from state n is exponential with rate $-A_{nn}$, and given that a jump from state n has occurred at time t, the conditional probability of the jump target is just

$$P(X(t^+) = m|X(t^-) = n, jump\ at\ t) = A_{nm}/\sum_{m:n\neq m} A_{nm}; \tag{34}$$

thus when the system decides to jump, it chooses its jump target with probability exactly proportional to A_{nm}. This gives a straightforward recursive algorithm for sampling from the chain: given $X(t) = n$, jump to state m at time $t + s$, where s is an independent $\exp(-A_{nn})$ random variable and m is chosen according to equation (34). Just as in the Poisson case (where the corresponding algorithm is given by the time-rescaling theorem), this algorithm is much more efficient than the obvious discrete-time approximation (in which we sample from a Markov chain in steps of some sufficiently small dt and transition rates $P(m|n) = A_{nm}dt, m \neq n$).

Now we know when we hear "rates" that a differential equation is lurking in the background. Here we have the linear ODE

$$\frac{\partial \vec{P}(t)}{\partial t} = A^T \vec{P}(t),$$

with initial conditions $\vec{P}(0)$, where $\vec{P}(t)$ denotes the vector

$$\vec{P}(t)_n \equiv P(X(t) = n)$$

(we will give a detailed derivation of a more general version of this ODE shortly); as usual, this linear equation has a solution of (matrix) exponential form,

$$\vec{P}(t) = \exp(tA)^T \vec{P}(0).$$

Thus to compute the marginal probability $P(X(t) = n)$ for any arbitrary time in the future we need simply compute a matrix exponential and then perform a matrix-vector multiplication; this is the analog in discrete time of multiplying by the t-th power of the transition matrix α.

4.1 The MLE for the transition rate matrix A generalizes the MLE for the firing rate λ in the Poisson process

How do we estimate A from data? As usual, we start by writing down the likelihood. Since continuous-time Markov chains generalize the Poisson process, it's natural to expect that we can mimic the derivation of the point process likelihood here. Namely, we discretize time in bins of width dt and then let $dt \to 0$. The discretized likelihood of A is

$$L_{discrete}(A) = \prod_{j'}[1 + A_{X(t_{j'})X(t_{j'}+dt)}dt + o(dt)] \prod_j [A_{X(t_j)X(t_j+dt)}dt + o(dt)],$$

where j indexes all time points where a transition is observed in $X(t)$ and j' indexes all other times. As before, when we expand the logarithm and take limits, we have

$$\begin{aligned} L(A) &= \lim_{dt\to 0} \prod_{j'}[1 + A_{X(t_{j'})X(t_{j'}+dt)}dt + o(dt)] \prod_j [A_{X(t_j)X(t_j+dt)}dt + o(dt)] \\ &\propto \prod_j \exp[w_j A_{X(t_j)X(t_j)}] A_{X(t_j)X(t_{j+1})}, \end{aligned}$$

where w_j denotes the j-th waiting time, the length of time between observed transitions j and $j+1$. (These waiting times are exponentially distributed with parameter $-A_{X(t_j)X(t_j)}$, as discussed above.)

Now the MLE $\hat{A}$ can be constructed easily. Using the exponential representation, it's easy to see (using the usual MLE for exponentially-distributed data) that

$$\hat{A}_{nn} = -(\bar{w}_n)^{-1},$$

with $\bar{w}_n$ defined as the mean observed waiting time in state n; this generalizes the MLE for the rate λ in the Poisson process. Now

$$\hat{A}_{nm} = -\hat{A}_{nn} N_{nm} / \sum_{m:n \neq m} N_{nm},$$

where N_{nm} is the observed number of transitions from state n to m. Clearly $\sum_m \hat{A}_{nm} = 0$, as desired.

See e.g. (Karlin and Taylor, 1981; Norris, 2004) for more details on continuous-time Markov chains with discrete state spaces.

4.2 Example: voltage-gated ion channels with time-dependent transition rates

In the case of voltage-gated ion channels, the transition matrix A depends explicitly on the voltage V (where the voltage signal $V(t)$ is assumed fully and noiselessly observed for now), and therefore $A(V(t))$ is now a function of time; thus we modify the above equation to the possibly inhomogeneous counterpart

$$\frac{\partial \vec{P}(t)}{\partial t} = A(V(t))^T \vec{P}(t),$$

with solution

$$\vec{P}(t) = \exp(\int_0^t A(V(s))ds)^T \vec{P}(0).$$

(Note that when $A(t)$ is in fact constant, this reduces to our original time-homogeneous formula.)

Sampling from this model may be performed, once again, by time-rescaling. We recurse as follows: draw $u_j \sim \exp(1)$. Given that X is in state X_j after j transitions, solve the time-rescaling equation

$$u_j = -\int_{t_j}^{t_{j+1}} A_{X_j X_j}(V(t))dt$$

for t_{j+1}; this gives us the time of the next transition t_{j+1}. Choose the next state of X (that is, the state which X jumps to at time t_{j+1} according to the probability mass function

$$\begin{aligned} P(X(t_{j+1}^+) = m | X(t_{j+1}^-) &= n, jump\ at\ t_{j+1}) = A_{nm}(V(t_{j+1})) / \sum_{l:l \neq n} A_{nl}(V(t_{j+1})) \\ &= n, jump\ at\ t_{j+1}) = -A_{nm}(V(t_{j+1}))/A_{nn}(V(t_{j+1})), \end{aligned}$$

and continue recursively.

5 Continuous-time HMMs

We now turn to hidden Markov models whose underlying Markov chains evolve (and whose observations are recorded) in continuous time. It is interesting to note, before we move on, that the direct analog of hidden Markov models on discrete state spaces don't make much sense in continuous time; since the hidden process q_t remains constant for nonzero lengths of time, we would effectively see an infinite number of observations from y_t, and would therefore be able to determine q_t deterministically. (Compare Figs. 2 and 3; the example with Gaussian observations — where the parameters of the Gaussian are independent of the timestep dt — is much more informative than the example in which the observations are given by a Poisson process, where the information per bin scales with dt.) Thus the HMMs we will examine will have a slightly modified structure, in which the observations y_t are not "infinitely informative" for the hidden process q_t.

As in the discrete setting, there are several key problems that we need to solve given observations Y: what is the likelihood $p(Y|\theta)$? How can we efficiently compute the forward probabilities $P(q_t, Y_{0:t})$, or the forward-backward probabilities $P(q_t|Y_{0:T}), t < T$? Many of these problems can be solved by fairly direct adaptations of the corresponding discrete-time algorithms; however, the following examples will illustrate a few cases in which it is more enlightening and computationally efficient to work directly in continuous time.

5.1 Example: the switching Poisson process in continuous time

We have already discussed this model at length in discrete time. Let's look at a special case in which the continuous-time formalism becomes useful: let $\lambda_1 = 0$ and $\lambda_2 = \lambda > 0$. Then y_t becomes a renewal process, and we may profitably apply renewal theory to understand the behavior of the model. For example, the cdf of the interspike intervals here is simply

$$F(t) = \left[\exp(tA)^T \pi\right]_3,$$

with $\pi = (0\ 1\ 0)^t$ and

$$A^T = \begin{pmatrix} -a & b & 0 \\ a & -b-\lambda & 0 \\ 0 & \lambda & 0 \end{pmatrix},$$

where λ is the rate of the Poisson process when X is in state two (recall that the rate in state one is zero), and a, b determine the transition rates for q_t (a is the transition rate from state 1 to state 2, and b is the transition rate in the opposite direction). One way to look at this is as an augmented Markov chain $X(t)$ such that states one and two are as for our original process H, and state three corresponds to having just seen a spike. X begins at state two (since we just observed a spike, we know we must be in the state in which spikes occur with a positive rate), then can either jump into state one or state three. State three is absorbing and signifies that a spike just occurred; since we are only interested in the first spike, we do not allow X to re-enter the spiking state after a spike has been observed. Clearly this augmented-chain approach generalizes to the case that $X(t)$ has more than two states.

From the cdf, we may as usual derive the pdf by differentiating:

$$p(t) = \frac{\partial}{\partial t}\left[\exp(tA)^T \pi\right]_3 = \left[A^T \exp(tA)^T \pi\right]_3,$$

and since this is a renewal process, we may sample from y_t (by drawing i.i.d. from this $p(t)$) without ever sampling from q_t. We may also compute the autocorrelation function and asymptotic mean firing rate using standard renewal theory tools:

$$p(spike\ at\ t|spike\ at\ 0) = \sum_{i>0} p(t)*^i,$$

where $p(t)*^i$ indicates the i-fold convolution of $p(t)$ $(p(t)*^1 = p(t))$ and

$$\lim_{t\to\infty} p(spike\ at\ t) = (\int tp(t)dt)^{-1}.$$

5.2 Example: ion channel models in continuous time

Another important case in which Y is not completely informative about Q is when the emissions probabilities are equal, $p(y_t|q_t = n) = p(y_t|q_t = m)$, for one or more pairs of states (n, m). This case has been well-studied in the context of ion channel models (Colquhoun and Hawkes, 1982; Ball and Sansom, 1989; Ball and Rice, 1992; Hawkes, 2004).

The simplest example of an HMM in this case is as follows. Let the observed current y_t have two possible values: high or low current (we assume this current is observed noiselessly for now, although of course this assumption may be relaxed as well (Chung et al., 1990; Venkataramanan and Sigworth, 2002)). Let the channel have K states, with the first k of these states passing high current and the remaining $K - k$ states passing low current. Then A may be partitioned in the block form

$$A = \begin{pmatrix} A_{hh} & A_{hl} \\ A_{lh} & A_{ll} \end{pmatrix}.$$

Now let's imagine we have observed a set of transition times $\{t_i\}$, from low to high currents or vice versa, perhaps with some gaps in the record (marked by transitions from observed to unobserved).

Let's start with the simplest case: we observe the current fully and see no transitions; the current is high throughout the observation period, $[0, T_1)$. By a simple adaptation of our usual arguments (compute the forward probabilities, then let $dt \to 0$), it's not hard to see that the forward distribution is

$$a_n(t) = P(X(t) = n, Y_{0:t}) = \left[\begin{pmatrix} \exp(tA_{hh})^T & 0 \\ 0 & 0 \end{pmatrix} \pi \right]_n, \ 0 < t < T_1,$$

where π denotes the initial probability vector; the zeros in the above matrix are due to the fact that we know that the channel must be in the high-current state on the observation interval $[0, T_1)$.

Now what if we see a transition from high to low current at time T_1? We form $a(T_1^-)$ by the above method, and then obtain

$$P(X(T_1) = n, Y_{0,T_1}) \propto \left[\begin{pmatrix} 0 & A_{hl} \\ 0 & 0 \end{pmatrix} a(T_1^-) \right]_n = \left[\begin{pmatrix} 0 & A_{hl} \\ 0 & 0 \end{pmatrix} \begin{pmatrix} \exp(T_1 A_{hh})^T & 0 \\ 0 & 0 \end{pmatrix} \pi \right]_n.$$

Similarly, if following this transition at time t_1 we observe T_2 time units of low current, we may compute

$$P(X(t) = n, Y_{0,t}) \propto \left[\begin{pmatrix} 0 & 0 \\ 0 & \exp(tA_{ll})^T \end{pmatrix} \begin{pmatrix} 0 & A_{hl} \\ 0 & 0 \end{pmatrix} \begin{pmatrix} \exp(T_1 A_{hh})^T & 0 \\ 0 & 0 \end{pmatrix} \pi \right]_n, \ T_1 < t < T_1 + T_2.$$

Finally, if we make no observation in the next time interval, $[T_1 + T_2, T_1 + T_2 + T_3]$, we simply use the marginal dynamics $p(q_t|q_{t-1})$ to propagate the forward probabilities (that is, the term $p(y_t|q_t)$ is independent of q_t, and therefore may be neglected):

$$P(X(t) = n, Y_{0,t}) \propto \left[\exp[(t - T_2 - T_1)A]^T \begin{pmatrix} 0 & 0 \\ 0 & \exp(T_2 A_{ll})^T \end{pmatrix} \begin{pmatrix} 0 & A_{hl} \\ 0 & 0 \end{pmatrix} \begin{pmatrix} \exp(T_1 A_{hh})^T & 0 \\ 0 & 0 \end{pmatrix} \pi \right]_n,$$

$T_1 + T_2 < t < T_1 + T_2 + T_3$.

In general, given any sequence of observed transitions from low to high current and vice versa and unobserved intervals, we may simply iterate these simple propagate and transition steps; see (Hawkes, 2004) for details. As usual, we obtain the marginal likelihood by summing over the forward distribution,

$$P(Y_{0:T}) = \sum a_n(T).$$

The key point is that these continuous-time techniques allow us to perform all of these likelihood computations with just a few simple matrix operations; on the other hand, the discrete-time analog, in which we would compute the forward probabilities by recursing equation (16) with some sufficiently small timestep dt, requires $O(T/dt)$ computations, which is clearly inefficient if dt is very small (as is often the case, since the accuracy of the computation increases as we decrease dt).

5.3 Example: multiple ion channels; functions of continuous-time Markov chains

The above discussion assumed that we had access to the current through a single channel. More generally, of course, a given patch of membrane may contain many different channels. The above analysis may be generalized directly to the case of N channels by working with the joint distributions $P(\vec{X}(t))$, but this quickly becomes unwieldy as N increases. An alternate (moment-based) solution may be constructed using basic Markov chain theory, instead of the more sophisticated but computationally intensive HMM technology we have been developing.

Imagine that there are N different channels in our patch of membrane (where N might be quite large, and unknown), and that each channel evolves independently of the others (this approximation makes some sense in a voltage-clamp setting, where the currents passed by one channel are not allowed to perturb the voltage, and therefore the behavior of the other possibly voltage-sensitive channels in the patch). In this context it makes sense to look at bulk quantities, such as the mean and covariance of the current passed as a function of time, since means and covariance functions will add linearly if the channels' behavior is mutually independent. If the total membrane current at time t is given by

$$I(t) = \sum_j a_j I_j(t),$$

where j indexes the different channel types and $a_j \geq 0$ denotes the density of the j-th channel type in the observed membrane patch, then

$$E[I(t)] = \sum_j a_j E[I_j(t)]$$

and

$$Cov\big[I(s), I(t)\big] = \sum_{j,j'} a_j a_{j'} Cov\big[I_j(s), I_{j'}(t)\big].$$

We will assume stationarity here, to keep the notation somewhat manageable: i.e., $E[I(t)] = E[I(0)]$ and $Cov[I(s), I(s+t)] = Cov[I(0), I(t)]$.

Now computing the first two moments $E[I_j(0)]$ and $E[I_j(0)I_j(t)]$ of the current passed by j-th channel type is fairly straightforward, if we recognize these quantities as expectations of a function of a Markov chain. Begin with the mean $E[I_j(0)]$: we have

$$E[I_j(0)] = \sum_n P_j(n) I(n, j),$$

where $I(n, j)$, the current passed by channel j in state n, may be seen as a simple function of the Markov chain $X_j(t)$, and

$$P_j(n) = \lim_{t \to \infty} P(X_j(t) = n),$$

the equilibrium distribution that the channel is in state n. If we assume that the transition matrix A_j is diagonalizable, with a single zero eigenvalue (with corresponding eigenvector $\vec{q}_{0,j}$), then the usual exponential representation of $P(X_j(t) = n)$ shows that

$$P_j(n) = q_{0,j}(n),$$

and therefore we have the explicit solution

$$E[I_j(0)] = \sum_n q_{0,j}(n) I(n, j).$$

For the second moment, write

$$E[I_j(0)I_j(t)] = \sum_n P_j(n) I(n, j) E[I_j(t) | X_j(0) = n],$$

and

$$E[I_j(t) | X_j(0) = n] = \sum_m I(m, j) p(X_j(t) = m | X_j(0) = n);$$

the latter probability may be computed, as usual, by the exponential representation

$$p(X_j(t) = m | X_j(0) = n) = \left[\left(e^{A_j t} \right)^T \delta_n \right]_m,$$

where δ_n denotes the Kronecker delta on n.

Once we have computed these expectations $E[I_j(0)]$ and $E[I_j(0)I_j(t)]$, we may fit a_j using a moment-matching technique: we choose $a_j \geq 0$ such that the predicted moments $\sum_j a_j E(I_j(0))$ and $\sum_j a_j E(I_j(0)I_j(t))$ match the observed moments $E(I(0))$ and $E(I(0)I(t))$ as closely as possible (this may be considered a "method of moments" estimator (Schervish, 1995)). Equivalently, since we have assumed stationarity, we may attempt to fit the empirical power spectrum (i.e., work with the autocovariance of $I(t)$ in the frequency domain instead of the time domain). See e.g. (DeFelice, L., 1981; Colquhoun and Hawkes, 1982; Manwani and Koch, 1999; Steinmetz et al., 2000; Hawkes, 2004) for details, related work, and extensions.

6 Multistate generalized linear models for spike trains in continuous time

We discussed several advantages of the continuous-time formalism above. One primary advantage is computational. If we return to the multistate GLM context and examine the computational requirements of the EM algorithm, for example, we see that we need to calculate the forward and backward probabilities for every time-step t, possibly at a short time scale dt (e.g., ~ 1 ms, if we are interested in short timescale spike history effects). This can quickly lead to high demands on memory and computation time, just to calculate the E-step. It turns out to be possible to reduce these computational requirements by making use of the differential-equation formulation of continuous-time HMMs, as we will discuss in detail below.

We begin by deriving the appropriate rate matrices (the $dt \to 0$ limits of the transition and emission matrices $\alpha(t)$ and $\eta(t)$; we will suppress the dependence of these matrices on dt where possible to reduce notation):

$$\lim_{dt\to 0} \alpha_{nm}(t) = \lim_{dt\to 0} \frac{\lambda'_{nm}(t)dt}{1+\sum_{l\neq n}\lambda'_{nl}(t)dt} = \lambda'_{nm}(t)dt \qquad m \neq n. \tag{35}$$

For the diagonal terms, a Taylor expansion of $1/(1+x)$ yields

$$\lim_{dt\to 0} \alpha_{nn}(t) = \lim_{dt\to 0} \frac{1}{1+\sum_{l\neq n}\lambda'_{nl}(t)dt} = 1 - \sum_{l\neq n}\lambda'_{nl}(t)dt; \tag{36}$$

note that

$$\sum_m \lim_{dt\to 0} \frac{\alpha_{nm}(t)}{dt} = 1,$$

as it must. We therefore define the rate matrix A as

$$A_{nm}(t) = \begin{cases} \lambda'_{nm}(t) & m \neq n \\ -\sum_{l\neq n}\lambda'_{nl}(t) & m = n. \end{cases} \tag{37}$$

Then α can be written as

$$\alpha(t) = I + A(t)dt + o(dt), \tag{38}$$

where I is the identity matrix.

We may handle the $dt \to 0$ limit of η similarly:

$$\lim_{dt\to 0} \eta_{ni}(t) = \lim_{dt\to 0} \frac{(\lambda_n(t)dt)^i \, e^{-\lambda_n(t)dt}}{i!} \qquad i = 0, 1, 2, \ldots \tag{39}$$

Thus,

$$\lim_{dt\to 0} \eta_{n0}(t) = \lim_{dt\to 0} e^{-\lambda_n(t)dt} = 1 - \lambda_n(t)dt, \tag{40}$$

$$\lim_{dt\to 0} \eta_{n1}(t) = \lim_{dt\to 0} (\lambda_n(t)dt)\, e^{-\lambda_n(t)dt} = \lambda_n(t)dt, \tag{41}$$

and

$$\lim_{dt\to 0} \eta_{nk}(t) = \lim_{dt\to 0} (\lambda_n(t)dt)^k \, e^{-\lambda_n(t)dt}/k! = o(dt), \ \forall k > 1.$$

Thus, as $dt \to 0$, there will never be more than one spike per dt, and so the η matrix reduces to a simple two-column matrix (as usual, the Poisson distribution effectively becomes a binary distribution in this limit).

Now we calculate our forward probabilities in continuous time. For time intervals in which no spike was observed, from Eq. 16 we have

$$\begin{aligned} a_n(t) &= \eta_{n0}(t)\left(\sum_{m=1}^{K} \alpha_{mn}(t) a_m(t-dt)\right) \\ &= (1-\lambda_n(t)dt)\left(\sum_{m=1}^{K} \alpha_{mn}(t) a_m(t-dt)\right) \end{aligned} \tag{42}$$

which can be written in matrix form as

$$\begin{aligned} \vec{a}(t) &= \left(I - \mathrm{diag}(\vec{\lambda}(t))dt\right) \alpha^T(t)\vec{a}(t-dt) \\ &= \left(I - \mathrm{diag}(\vec{\lambda}(t))dt\right) (I + A(t)dt)^T \,\vec{a}(t-dt) + o(dt) \\ &= \vec{a}(t-dt) + \left(A^T(t) - \mathrm{diag}(\vec{\lambda}(t))\right) \vec{a}(t-dt)dt + o(dt); \end{aligned}$$

collecting terms of order dt yields a nice linear differential equation,

$$\frac{\partial \vec{a}(t)}{\partial t} = \left(A^T(t) - \mathrm{diag}(\vec{\lambda}(t))\right) \vec{a}(t); \tag{43}$$

this generalizes the ODE for $\vec{P}(t)$ we discussed in the last section. Therefore if t_{i-1} and t_i are consecutive spike times, and we know $\vec{a}(t_{i-1}^+)$, we may determine $\vec{a}(t_i^-)$ by simply evolving this ODE forward to time t_i. Now we just need to compute the update at the spike times t_i:

$$\begin{aligned} \vec{a}(t_i^+) &= \left(\mathrm{diag}(\vec{\lambda}(t_i))dt\right) \alpha^T(t_i)\vec{a}(t_i^-) \\ &= \left(\mathrm{diag}(\vec{\lambda}(t_i))dt\right) \left(I + A^T(t_i)dt\right) \vec{a}(t_i^-) \\ &\propto \mathrm{diag}(\vec{\lambda}(t_i))\vec{a}(t_i^-) + o(dt). \end{aligned}$$

Note that the resulting update rule,

$$\vec{a}(t_i^+) = \mathrm{diag}(\vec{\lambda}(t_i))\vec{a}(t_i^-) \tag{44}$$

is discontinuous at the spike time, whereas it is clear from the ODE representation that $\vec{a}(t)$ is a smooth function of t between the spike times. Finally, $\vec{a}(0)$ is initialized as $\vec{\pi}$.

The backward probabilities may be adapted to the continuous time setting in an analogous manner. Between spikes times we have the update

$$\begin{aligned} b_n(t-dt) &= \sum_{m=1}^{K} \alpha_{nm}(t)\eta_{my_t}(t) b_m(t) \\ &= \sum_{m=1}^{K} \alpha_{nm}(t)(1-\lambda_n(t)dt) b_m(t) \end{aligned} \tag{45}$$

which in matrix form becomes

$$\begin{aligned} \vec{b}(t-dt) &= \alpha(t)\left(I - \mathrm{diag}(\vec{\lambda}(t))dt\right) \vec{b}(t) \\ &= (I + A(t)dt)\left(I - \mathrm{diag}(\vec{\lambda}(t))dt\right) \vec{b}(t) \\ &= \vec{b}(t) - \left(\mathrm{diag}(\vec{\lambda}(t)) - A(t)\right) \vec{b}(t)dt, \end{aligned}$$

yielding the differential equation

$$\frac{d\vec{b}(t)}{dt} = \Big(\text{diag}(\vec{\lambda}(t)) - A(t)\Big)\,\vec{b}(t). \tag{46}$$

The spike-time update follows exactly as before:

$$\vec{b}(t_i^-) = \text{diag}(\vec{\lambda}(t_i))\vec{b}(t_i^+) \tag{47}$$

The initialization of the backward probabilities remains unchanged from the discrete case, $\vec{b}(T) = \vec{1}$.

As in the discrete-time case, the log-likelihood may be calculated as

$$L(\theta|Y) = \log p(Y|\theta) = \log \sum_{n=1}^{K} a_n(T), \tag{48}$$

and the individual marginal distributions of $p(Q|Y,\theta)$ are given by

$$p(q(t){=}n|Y,\theta) = \frac{a_n(t)b_n(t)}{p(Y|\theta)}. \tag{49}$$

While both the forward and backward probabilities jump discontinuously at spike times t_i, the marginals $p(q(t){=}n|Y,\theta)$ are continuous at all times t (assuming that the stimulus $\vec{x}(t)$ is smoothly varying), as can be seen by noting that

$$p(q(t_i^-) = n|Y,\theta) = \frac{a_n(t_i^-)b_n(t_i^-)}{p(Y|\theta)} = \frac{a_n(t_i^-)\lambda_n(t_i)b_n(t_i^+)}{p(Y|\theta)} = \frac{a_n(t_i^+)b_n(t_i^+)}{p(Y|\theta)} = p(q(t_i^+) = n|Y,\theta); \tag{50}$$

clearly the marginals are continuous between spike times, since both the forward and backward probabilities are.

Here it is worth noting the computational advantages of the continuous-time formulation: since the majority of time-steps are associated with the trivial “no spike” emission, it is clearly advantageous to consider numerically efficient methods for computing the forward and backward updates at these times. It is clear that the standard method for performing these updates, as detailed in equations (42) and (45), corresponds to a simple Euler scheme for solving the ODEs (43) and (46), respectively. Utilizing more efficient schemes (e.g., Runge-Kutta with adaptive time steps (Press et al., 1992)) can potentially lead to much more efficient computation.

Finally, to compute the E-step, we need to consider the pairwise marginals $p(q(t), q(s)|Y,\theta)$. Here it is useful to define the conditional transition rates $r_{n\to m}(t)$:

$$\begin{aligned} r_{n\to m}(t) &\equiv \lim_{dt\to 0} \frac{p(q(t){=}m|q(t-dt){=}n, Y, \theta)}{dt} \\ &= \lim_{dt\to 0} \frac{p(q(t){=}m,q(t-dt){=}n|Y,\theta)}{p(q(t-dt){=}n|Y,\theta)dt} \\ &= \lim_{dt\to 0} \frac{a_n(t-dt)\alpha_{nm}(t)\eta_{my_t}(t)b_m(t)/p(Y|\theta)}{a_n(t-dt)b_n(t-dt)dt/p(Y|\theta)} \\ &= \lim_{dt\to 0} \frac{\alpha_{nm}(t)\eta_{my_t}(t)b_m(t)}{b_n(t-dt)dt}. \end{aligned} \tag{51}$$

Between spike times, Eq. 51 becomes

$$\begin{aligned} r_{n\rightarrow m}(t) &= \lim_{dt\rightarrow 0} \frac{\lambda'_{nm}(t)dt(1-\lambda_m(t)dt)b_m(t)}{b_n(t-dt)dt} \\ &= \lambda'_{nm}(t)\cdot\frac{b_m(t)}{b_n(t)} \end{aligned} \tag{52}$$

and at spike times,

$$\begin{aligned} r_{n\rightarrow m}(t_i) &= \lim_{dt\rightarrow 0} \frac{\lambda'_{nm}(t_i)dt\lambda_m(t_i)dt\cdot b_m(t_i)}{b_n(t_i^-)dt} \\ &= \lim_{dt\rightarrow 0} \frac{\lambda'_{nm}(t_i)\cdot\lambda_m(t_i)dt\cdot b_m(t_i)}{\lambda_n(t_i)dt\cdot b_n(t_i)} \\ &= \lambda'_{nm}(t_i)\cdot\frac{\lambda_m(t_i)}{\lambda_n(t_i)}\cdot\frac{b_m(t_i)}{b_n(t_i)}. \end{aligned} \tag{53}$$

Equations (52) and (53) have an intuitive explanation. Between spikes, $r_{n\rightarrow m}(t)$ (the rate of transition from state n to state m, given the stimulus and the observed spike train) is equal to $\lambda'_{nm}(t)$ (the transition rate at time t given the stimulus but no observations of the spike train) scaled by the ratio of the probabilities of the future given that the current state is m versus n. In other words, if the remainder of the spike-train can be better explained by having the neuron in state m than in state n at time t, the rate should be increased beyond $\lambda'_{nm}(t)$; otherwise it should be reduced. At the spike-times, the additional information of knowing that a spike occurred further scales the expected transition rate by the ratio of the firing rates between the two states, which is equal to the ratio of the firing rate in each state. As is obvious from equations (52) and (53), $r_{n\rightarrow m}(t)$ is discontinuous at the spike-times, but continuous between spikes.

Now we have all the ingredients in hand necessary to derive the M-step. We have the same three terms to maximize as in the discrete-time setting. The update for π requires us to maximize $\sum_{n=1}^{K} p(q(0){=}n|Y,\hat{\theta}^{(i)})\log\pi_n$, exactly as before. The update for the emissions parameters required that we optimize a function of the form $\sum_{t=1}^{T}\sum_{n=1}^{K} p(q_t{=}n|Y,\hat{\theta}^{(i)})\log\eta_{ny_t}$ in the discrete-time case; here, this term reduces to

$$\sum_{n=1}^{K}\left(\sum_{i\in spikes} w_n(t)\log\lambda_n(t_i) - \int_0^T w_n(t)\lambda_n(t)dt\right), \tag{54}$$

where we have abbreviate $w_n(t) = p(q(t) = n|Y,\hat{\theta}^{(i)})$; this objective function clearly retains all the usual concavity properties in $\vec{k}$, by the nonnegativity of $w_n(t)$.

Finally, to update the transition matrix α, we optimize the function

$$\begin{aligned}
&\sum_{t=2}^{T}\sum_{n=1}^{K}\sum_{m=1}^{K} p(q(t-1){=}n, q(t){=}m|Y, \hat{\theta}^{(i)}) \log \alpha_{nm}(t) \\
= \;& \sum_{t=2}^{T}\sum_{n=1}^{K} \left(\begin{array}{l} \sum_{m \neq n} p(q(t-1){=}n, q(t){=}m|Y, \hat{\theta}^{(i)}) \log(\lambda'_{nm}(t)dt) \\ + \, p(q(t-1){=}n, q(t){=}n|Y, \hat{\theta}^{(i)}) \log\left(1 - \sum_{l \neq n} \lambda'_{nl}(t)dt\right) \end{array} \right) \\
= \;& \sum_{i=1}^{T}\sum_{n=1}^{K} \left(\begin{array}{l} \sum_{m \neq n} w_n(t) r_{n \to m}(t) dt (\log \lambda'_{nm}(t_i) + \log dt) \\ - \, w_n(t) \sum_{l \neq n} \lambda'_{nl}(t_i) dt \end{array} \right) + o(dt) \\
\sim \;& \sum_{n=1}^{K}\sum_{m \neq n} \int_0^T w_n(t) \left(r_{n \to m}(t) \log \lambda'_{nm}(t) - \lambda'_{nm}(t) \right) dt; \qquad (55)
\end{aligned}$$

once again, the connections to the point-process loglikelihood should be clear. It is interesting to note that the GLM concavity conditions for this α update are more permissive in the continuous than in the discrete case: as equation (55) makes clear, in the continuous setting it is enough that the nonlinearity $g(.)$ in the definition of $\lambda'(t)$ is convex and log-concave (recall that the necessary conditions were stronger in the discrete case).

References

Ball, F. and Rice, J. (1992). Stochastic models for ion channels: introduction and bibliography. *Mathematical Bioscience*, 112:189–206.

Ball, F. and Sansom, M. (1989). Ion-channel gating mechanisms: Model identification and parameter estimation from single channel recordings. *Proceedings of the Royal Society of London. Series B, Biological Sciences*, 236:385–416.

Baum, L., Petrie, T., Soules, G., and Weiss, N. (1970). A maximization technique occuring in the statistical analysis of probabilistic functions of Markov chains. *Annals of Mathematical Statistics*, 41:164–171.

Bellman, R. (1957). *Dynamic Programming.* Princeton University Press.

Bezdudnaya, T., Cano, M., Bereshpolova, Y., Stoelzel, C., J.-M., A., and Swadlow, H. (2006). Thalamic burst mode and inattention in the awake LGNd. *Neuron*, 49:421–432.

Chung, S., Moore, J., Xia, L., Premkumar, L., and Gage, P. (1990). Characterization of single channel currents using digital signal processing techniques based on hidden markov models. *Phil. Trans. Roy. Soc. Lond. B*, 329:265–285.

Colquhoun, D. and Hawkes, A. (1982). On the stochastic properties of bursts of single ion channel openings and of clusters of bursts. *Philosophical Transactions of the Royal Society London B*, 300:1–59.

Cox, D. (1955). Some statistical methods connected with series of events. *Journal of the Royal Statistical Society, Series B*, 17:129–164.

de Gunst, M., Kunsch, M., and Schouten, J. (2001). Statistical analysis of ion channel data using Hidden Markov Models with correlated state-dependent noise and filtering. *JASA*, 96:805–815.

DeFelice, L. (1981). *Introduction to Membrane Noise.* Plenum Press.

Escola, S. and Paninski, L. (2008). Hidden Markov models applied toward the inference of neural states and the improved estimation of linear receptive fields. *Under review, Neural Computation.*

Fredkin, D. R. and Rice, J. A. (1992). Maximum Likelihood Estimation and Identification Directly from Single-Channel Recordings. *Proceedings of the Royal Society of London. Series B: Biological Sciences*, 249:125–132.

Gat, I., Tishby, N., and Abeles, M. (1997). Hidden Markov modeling of simultaneously recorded cells in the associative cortex of behaving monkeys. *Network: Computation in Neural Systems*, 8:297–322.

Hawkes, A. (2004). Stochastic modelling of single ion channels. In Feng, J., editor, *Computational Neuroscience: a comprehensive approach*, pages 131–158. CRC Press.

Herbst, J. A., Gammeter, S., Ferrero, D., and Hahnloser, R. H. (2008). Spike sorting with hidden markov models. *Journal of Neuroscience Methods*, 174(1):126 – 134.

Jones, L. M., Fontanini, A., Sadacca, B. F., Miller, P., and Katz, D. B. (2007). Natural stimuli evoke dynamic sequences of states in sensory cortical ensembles. *Proceedings of the National Academy of Sciences*, 104:18772–18777.

Jordan, M. I., editor (1999). *Learning in graphical models.* MIT Press, Cambridge, MA, USA.

Karlin, S. and Taylor, H. (1981). *A Second Course in Stochastic Processes.* Academic Press, New York.

Kemere, C., Santhanam, G., Yu, B. M., Afshar, A., Ryu, S. I., Meng, T. H., and Shenoy, K. V. (2008). Detecting Neural-State Transitions Using Hidden Markov Models for Motor Cortical Prostheses. *J Neurophysiol*, 100:2441–2452.

MacLean, J., Watson, B., Aaron, G., and Yuste, R. (2005). Internal dynamics determine the cortical response to thalamic stimulation. *Neuron*, 48:811–823.

Manwani, A. and Koch, C. (1999). Detecting and Estimating Signals in Noisy Cable Structures, I: Neuronal Noise Sources. *Neural Computation*, 11(8):1797–1829.

Moeller, J. and Waagepetersen, R. (2004). *Statistical inference and simulation for spatial point processes.* Chapman Hall.

Norris, J. (2004). *Markov Chains.* Cambridge University Press.

Press, W., Teukolsky, S., Vetterling, W., and Flannery, B. (1992). *Numerical recipes in C.* Cambridge University Press.

Rabiner, L. (1989). A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77:257–286.

Rubin, N. (2003). Binocular rivalry and perceptual multi-stability. *Trends in Neuroscience*, 26:289–291.

Sahani, M. (1999). *Latent variable models for neural data analysis.* PhD thesis, California Institute of Technology.

Sakmann, B. and Neher, B., editors (1995). *Single-channel recording.* Springer.

Salakhutdinov, R., Roweis, S. T., and Ghahramani, Z. (2003). Optimization with EM and expectation-conjugate-gradient. *International Conference on Machine Learning*, 20:672–679.

Schervish, M. (1995). *Theory of statistics.* Springer-Verlag, New York.

Sherman, S. (2001). Tonic and burst firing: Dual modes of thalamocortical relay. *Trends in Neuroscience*, 24:122–126.

Snyder, D. and Miller, M. (1991). *Random Point Processes in Time and Space.* Springer-Verlag.

Steinmetz, P., Manwani, A., Koch, C., London, M., and Segev, I. (2000). Subthreshold voltage noise due to channel fluctuations in active neuronal membranes. *Journal of Computational Neuroscience*, 9:133–148.

Venkataramanan, L. and Sigworth, F. (2002). Applying hidden markov models to the analysis of single ion channel activity. *Biophysical Journal*, 82:1930–1942.

Viterbi, A. (1967). Error bounds for convolutional codes and an asymptotically optimal decoding algorithm. *IEEE Trans. Informat. Theory*, IT-13:260–269.