Fast Evaluation of the Likelihood of an HMM: Ion Channel Currents with Filtering and Colored Noise*

Donald R. Fredkin
Department of Physics
University of California, San Diego
La Jolla, CA 92093

John A. Rice
Department of Statistics
University of California, Berkeley
Berkeley, CA 94720

June 18, 1997

Abstract

Hidden Markov models have been used in the study of single-channel recordings of ion channel currents for restoration of idealized signals from noisy recordings and for estimation of kinetic parameters. A key to their effectiveness from a computational point of view is that the number of operations to evaluate the likelihood, posterior probabilities, and the most likely state sequence are proportional to the product of the square of the dimension of the state space and the length of the series. However, when the state space is quite large, computations can become infeasible. This can happen when the record has been low pass filtered and when the noise is colored. In this paper we present an approximate method that can provide very substantial

*Research supported by the National Science Foundation

reductions in computational cost at the expense of only a very small error. We describe the method and illustrate through examples the gains that can be made in evaluating the likelihood.

1 Introduction

Hidden Markov models have recently found application to the analysis of single-channel recordings, both for the construction of an idealized quantal signal from a noisy recording (Chung et al., 1990; Fredkin and Rice, 1992a) and for estimation of kinetic parameters directly from the recording rather than from an idealized reconstruction (Albertson and Hansen, 1994; Fredkin and Rice, 1992b; Venkataramanan et al., 1996; Qin et al., 1994). Hidden Markov models have also been used in a variety of other areas, for example in speech recognition (Rabiner, 1989) and gene finding (Krogh et al., 1994). A key to their computational effectiveness is that the number of operations required to evaluate the likelihood or its gradient or to evaluate posterior probabilities is proportional to the product of the square of the dimension (D) of the state space and the length of the record (T) (Baum et al., 1970).

Filtering and colored noise complicate the application of hidden Markov methodology to ion channel recordings. In principle, the state space can be enlarged to include "metastates" (Fredkin and Rice, 1992a; Venkataramanan et al., 1996) and the standard algorithms can be used. In practice, however, the dimensionality of the new state space can easily become so large that computations are intractible. For example, if the underlying state space has cardinality six and a filter of length five is used, the number of operations required to evaluate the likelihood is of order 6^6T rather than 6^2T—a factor of more than 1000. The problem of large state space dimension also occurs in other extensions of hidden Markov models, for example (Ghahramani and Jordan, 1996).

In this paper we propose and illustrate an approximation strategy that can radically decrease the number of operations required to evaluate the likelihood while entailing little loss in accuracy. The basic idea is to ignore metastates that are either *a priori* or *a posteriori* highly unlikely. In an example to be presented in detail below, the number of operations is reduced by a factor of about 400.

The remainder of this paper is organized as follows. In section 2 we describe the hidden Markov model that relates a kinetic model to an observed noisy digital recording and show how it can be extended to account for filtering and colored noise. We then show how the basic recursions of (Baum et al., 1970) can be accomplished for the extended model and introduce approximations which produce lower bounds on the likelihood. Finally in section 2.5 we describe the way we have implemented evaluation of the likelihood and our approximations. A collection of examples motivated by models that have been proposed for ion channel kinetics are presented in Section 3. Here we examine in some detail the savings that can be accomplished via our approximations and the size of the errors consequently incurred. Section 4 contains a summary, conclusions, and discussion of further directions.

2 Theory

2.1 The Model

We assume that an N_s state Markov process underlies the kinetics. We consider a discrete time process, since in practice the data are samples at times $k\Delta t$. The one step transition probabilities P_{ij} for the transition $i \rightarrow j$ are related to the generator Q_{ij} of a continuous time Markov process by matrix exponentiation: $P = \exp Q\Delta t$.

Current levels $\mathcal{I}_i$ are associated with the states, with the values being, in general, not all distinct. For example, a system with two closed states and one open state would have $\mathcal{I}_1 = \mathcal{I}_2 = 0$, $\mathcal{I}_3 \neq 0$. Denote the temporal sequence of states by $s(t)$. In the absence of filtering and noise the observed current would be $x(t) = \mathcal{I}_{s(t)}$. In practice, because of filtering and noise the observed current is $I(t) = (a * x)(t) + W(t)$, where $a * x$ denotes the convolution $\sum_k a(k)x(t-k)$ and $a(0), a(1), \ldots, a(N_f)$ are filter coefficients; $W(t)$ is additive noise.

In this paper we assume that the noise $W(t)$ is independent of the state $s(t)$. We will usually assume the noise to be independent identically distributed (IID) Gaussian random variables with mean zero and variance σ^2. However, because we are already prepared to consider the effect of a filter, we can easily consider noise that is an autoregressive (AR) random process driven by IID Gaussian noise: $b * W = w$, where $b(0) = 1$ and $w(t)$ is IID Gaussian noise with mean zero and variance σ^2. The FIR filter with coefficients b can be considered a prewhitening filter (Venkataramanan et al., 1996). Applying this filter to the observations $I(t)$, we arrive at

$$y = b * a * x + w = f * x + w, \tag{1}$$

where $y = b * I$ and $f = b * a$. The coefficients $b(k)$ can be determined by some variant of the Levinson algorithm from the autocorrelation sequence of the noise (Venkataramanan et al., 1996). If the maximum lag in the sequence $b(k)$ is N_n, the effective filter $f(k)$ has maximum lag $N_e = N_f + N_n$. From now on we will work with (1), referring to $w(t)$ as the noise and $y(t)$ as the observation at time t. There are T observations at $t = 1 \ldots T$. For most purposes, we do not need the detailed structure of (1); it is sufficient that, conditional on the state sequence $s = s(-N_e + 1) \ldots s(T)$, the observations $y(t)$ are independent and the probability density $p(y(t) \mid s)$ depends only on $s(t) \ldots s(t - N_e)$: $p(y(t) \mid s) = g(y(t) \mid s(t) \ldots s(t - N_e))$.

2.2 Recursive Calculation of the Likelihood

We can include the filter in (1) by extension of the state space (Fredkin and Rice, 1992a) and working with a Markov chain whose states are the $N_s^{N_e+1}$ "metastates" $(s_0 \ldots s_{N_e})$. However, the transition matrix among the metastates is sparse and we find it slightly simpler to work with the original state space and extend the usual recursive procedure (Baum et al., 1970).

Define

$$\alpha_t(s(t) \ldots s(t - N_e)) = P[y(1) \ldots y(t) \,\&\, s(t) \ldots s(t - N_e)], \tag{2}$$

which can be computed recursively: With equilibrium probabilities $\pi(s)$ and transition probabilities $p(s' \mid s)$ we have

$$\alpha_0(s(0) \ldots s(-N_e)) = \prod_{k=0}^{-N_e+1} p(s(k) \mid s(k-1))\, \pi(s(-N_e)),$$

and, for $t = 1 \ldots T$,

$$\begin{aligned}
\alpha_t(s(t) \ldots s(t - N_e)) &= \sum_{s(t-N_e-1)} P[y_1 \ldots y_t \,\&\, s(t) \ldots s(t - N_e - 1)] \\
&= \sum_{s(t-N_e-1)} P[y_1 \ldots y_{t-1} \,\&\, s(t-1) \ldots s(t - N_e - 1))] \\
&\qquad \times P[s(t) \mid y_1 \ldots y_{t-1} \,\&\, s(t-1) \ldots s(t - N_e - 1))]
\end{aligned}$$

$$
\begin{aligned}
& \quad \times P[y_t \mid y_1 \ldots y_{t-1} \,\&\, s(t) \ldots s(t-N_e-1))] \\
= & \sum_{s(t-N_e-1)} \alpha_{t-1}(s(t-1) \ldots s(t-N_e-1))) \\
& \quad \times p(s(t) \mid s(t-1)) g(y_t \mid s(t) \ldots s(t-N_e)) \\
= & \ \tilde{\alpha}_{t-1}(s(t-1) \ldots s(t-N_e)) p(s(t) \mid s(t-1)) \\
& \quad \times g(y_t \mid s(t) \ldots s(t-N_e)),
\end{aligned} \tag{3}
$$

where, in the last line, we defined

$$\tilde{\alpha}_t(s(t) \ldots s(t-N_e+1)) = \sum_{s(t-N_e)} \alpha_t(s(t) \ldots s(t-N_e))$$

The likelihood is

$$L = P[y] = \sum_{s(T) \ldots s(T-N_e)} \alpha_T(s(T) \ldots s(T-N_e)).$$

In practice, we must renormalize the α_t to avoid underflow. We define

$$N_t = \sum_{s(t) \ldots s(t-N_e)} \alpha_t(s(t) \ldots s(t-N_e)),$$

$$\hat{\alpha}_t(s(t) \ldots s(t-N_e)) = \alpha_t(s(t) \ldots s(t-N_e))/N_t,$$

and $\hat{N}_t = N_t/N_{t-1}$. Note that $N_0 = 1$, using the definition of α_0, and N_T is the likelihood. We have

$$
\begin{aligned}
\hat{N}_t \hat{\alpha}_t(s(t) \ldots s(t-N_e)) \quad = & \sum_{s(t-N_e-1)} \hat{\alpha}_{t-1}(s(t-1) \ldots s(t-N_e-1)) \\
& \times p(s(t) \mid s(t-1)) g(y_t \mid s(t) \ldots s(t-N_e))
\end{aligned} \tag{4}
$$

and

$$L = \prod_{t=1}^{T} \hat{N}_t. \tag{5}$$

The $\hat{N}_t$ are determined by the requirement that

$$\sum_{s(t) \ldots s(t-N_e)} \hat{\alpha}_t(s(t) \ldots s(t-N_e)) = 1. \tag{6}$$

2.3 Related Recursive Algorithms

Our focus is on calculation of the likelihood, but we digress briefly to give the form of the EM (Baum et al., 1970) and Viterbi (Viterbi, 1967) algorithms using the formalism of section 2.2. We do not necessarily advocate use of the EM algorithm. Some form of quasi-Newton method (Fletcher, 1987) may be more effective. However, the recursions needed for the EM algorithm can also be regarded as calculations of the posterior probabilities of states given the data, and, as such, can be useful for reconstruction of the ideal signal based on a fictitious hidden Markov model. Similarly, the Viterbi algorithm consists of recursions needed to find the most probable state sequence. All of these recursions are complicated by the large numbers of metastates and our approximations can be applied to all of them.

2.3.1 EM Algorithm

Define

$$\beta_t(s(t)\dots s(t-N_e+1)) = P[y(t+1)\dots y(T) \mid s(t)\dots s(t-N_e+1)], \quad (7)$$

which, like α, can be computed recursively:

$$\beta_T(s(T)\dots s(T-N_e+1)) = 1,$$

and, for $t < T$,

$$\begin{aligned}\beta_t(s(t)\dots s(t-N_e+1)) \quad = \quad & \sum_{s(t+1)} p(s(t+1) \mid s(t)) \\ & \times g(y(t+1) \mid s(t+1)\dots s(t-N_e+1)) \\ & \times \beta_{t+1}(s(t+1), s(t)\dots s(t-N_e+2)).\end{aligned}$$

It is straightforward to show, using a lemma from (Baum et al., 1970), that the EM algorithm leads to the iteration scheme $p^0 \mapsto p$ for the transition probabilities, where

$$p(b|a) = \frac{\mathcal{N}_{ab}}{\mathcal{D}_a}$$

with

$$\begin{aligned}\mathcal{N}_{ab} \quad = \quad & \sum_s \sum_{t=0}^{T} \tilde{\alpha}_t^0(a, s_1 \dots s_{N_e-1}) p^0(b \mid a) g(y(t+1) \mid b, a, s_1 \dots s_{N_e-1}) \\ & \times \beta_{t+1}^0(b, a, s_1 \dots s_{N_e-2}),\end{aligned}$$

$$\mathcal{D}_a = \sum_s \sum_{t=0}^{T-1} \tilde{\alpha}_t^0(a, s_1 \dots s_{N_e-1})\beta_t^0(a, s_1 \dots s_{N_e-1}),$$

and α^0 and β^0 are computed with p^0. (We use the notation $s_1 \dots s_{Ne}$ to emphasize that these dummy variables are not associated with specific times.)

2.3.2 Viterbi Algorithm

The Viterbi algorithm (Viterbi, 1967) is a dynamic programming method for finding the sequence of states, $\{\hat{s}(t)\}$, that is most likely given the observed data. It has been used by (Qin et al., 1994) for finding an idealized record from which the kinetics parameters are estimated by maximizing the likelihood of the resulting sequence of dwell times. It has also been used in the context of speech recognition by (Juang and Rabiner, 1990); in this alternative to standard maximum likelihood estimation, in which the marginal likelihood of the kinetic parameters is maximized, here the joint likelihood of the kinetic parameters and the sequence of unobserved states is maximized. To formulate the Viterbi algorithm in the case of filtering and colored noise we follow the notation of (Fredkin and Rice, 1992a). Let

$$H_t(s(t-N_e), \dots, s(t)) = \{\hat{s}(-N_e), \dots, \hat{s}(t-N_e-1)\} \tag{8}$$

be the most likely state sequence up to and including time $t - N_e - 1$. It maximizes

$$\begin{aligned} &P(s(-N_e), \dots, s(t)) = \\ &\quad P(s(-N_e), \dots, s(0))g(y_0|s(-N_e), \dots, s(0)) \\ &\qquad \times \prod_{k=0}^{t-1} p(s(k+1)|s(k))g(y_{k+1}|s(k-N_e+1), \dots, s(k+1)) \end{aligned}$$

Let

$$\begin{aligned} &L_t(s(t-N_e), \dots, s(t)) = \\ &\quad P(\hat{s}(-N_e), \dots, \hat{s}(t-N_e-1), s(t-N_e), \dots, s(t)) \end{aligned}$$

Then L_t satisfies the recursion

$$\begin{aligned} &L_{t+1}(s(t-N_e+1), \dots, s(t+1)) = \\ &\quad \max_{s(t-N_e)} L_t(s(t-N_e), \dots, s(t)) \\ &\qquad \times p(s(t+1)|s(t))g(y_{t+1}|s(t-N_e+1), \dots, s(t+1)) \end{aligned} \tag{9}$$

Denote the maximizer by $\hat{s}(t-N_e)$. H_t then also satisfies the recursion relation

$$H_{t+1}(s(t-N_e+1),\ldots,s(t+1)) = H_t(s(t-N_e),\ldots,s(t)) \circ \hat{s}(t-N_e)$$

where $\circ$ denotes concatenation.

2.4 Approximations

Consider the computational cost of using (4–6) to compute the likelihood. For each time t we compute $N_s^{N_e+1}$ values of α, one for each of the metastates $s_0 \ldots s_{N_e}$, and each such computation requires order N_s operations.. Similarly, computation of $\hat{N}_t$ requires $N_m - 1$ additions. Calculation of the likelihood thus takes $O(N_s^{N_e+2}T)$ floating point operations. If we compare this with the computational cost when there is neither a filter nor autoregressive noise coloration, we see that the work is multiplied by a factor $N_s^{N_e}$. For a simple scheme involving three states ($N_s = 3$) and maximum lag due to filtering and noise coloration $N_e = 10$, we have a cost amplification of $3^{10} = 59049$. If, to be optimistic, we could compute the likelihood for $N_e = 0$ in 1 μs, we now require a full second to compute the likelihood once. And we will need to compute the likelihood many times to maximize it.

The key to speeding up the calculation of the likelihood is the observation that the exact scheme, whether in the efficient form (4–6) or in the raw form

$$L = P[y] = \sum_s P[y \,\&\, s], \tag{10}$$

where y and s are histories ($y_1 \ldots y_T$ and $s_{-N_e+1} \ldots s_T$), involves a large number of improbable and numerically unimportant sequences of states. For example, in a two state model ("closed" and "open") the transition probabilities (P_{12}, P_{21}) are likely to be extremely small. If they were not we would say that the sampling interval Δt was too large. If $N_e = 4$, say, we expect to encounter metastates containing multiple transitions (like $C \to O \to C \to O \to C$) rarely and their contribution to the sum in (10), or the role of any α for such a metastate, might be negligible.

Our primary approximation is to choose a small tolerance ϵ_1 and neglect any metastate $s_0 \ldots s_{N_e}$ for which the conditional probability

$$P[s_1 \ldots s_{N_e} \mid s_0] < \epsilon_1. \tag{11}$$

We discuss quantitatively the effective reduction in the number of metastates and in the computation time in section 3 for a variety of realistic examples and choices of ϵ_1.

The selection of metastates to be neglected based on (11) is made once at the beginning of the calculation of the likelihood. The selection depends, of course, on the transition probabilties, so the selection must be made repeatedly in the course of maximization of the likelihood, once each time the likelihood is evaluated.

We can make a second approximation of a more dynamical character: whenever, in evaluating (4), we encounter a value

$$\sum_{s(t-N_e-1)} \hat{\alpha}_{t-1}(s(t-1)\dots s(t-N_e-1)) < \epsilon_2 \tag{12}$$

we replace the sum by zero. Note that this sum is the renormalized version of $\tilde{\alpha}_{t-1}(s(t-1)\dots s(t-N_e))$. The elimination of terms using (12) depends on the data, y, while the simplification using (11) depends only on the model and not at all on the data. The utility of this approximation is also discussed in section 3.

Similar approximations can be applied to the EM and Viterbi algorithms. For example in the Viterbi algorithm note that one has to update L_t as in (9) for each of its $N_s^{N_e+1}$ arguments (metastates). An approximation which discards those metastates which have small *a priori* probability can drastically reduce the total number of calculations. Also if $g(y_{t+1}|s(t-N_e+1),\dots,s(t+1))$ is small, an approximation can be made in which $L_{t+1}(s(t-N_e+1),\dots,s(t+1))$ is set equal to zero and then ignored in the step $t+1 \to t+2$.

2.5 Implementation

We use (4–6) to compute the likelihood. In this section we discuss some design decisions we made when implementing the calculation on a computer.

We must store values of $\hat{\alpha}_t(s_0 \dots s_{N_e})$ and update them as t ranges from 0 to T. There are many indices, each with a modest range, and the number of indices depends on the model. This suggests that a multidimensional array, with many nested loops to manipulate the values as t progresses from 0 to T, might not be the best scheme. We prefer to keep track of the various values in a forest of N_s ordered trees. (We use the terminology of (Aho et al., 1974) throughout this section.) Let us use a simple example for ease of exposition:

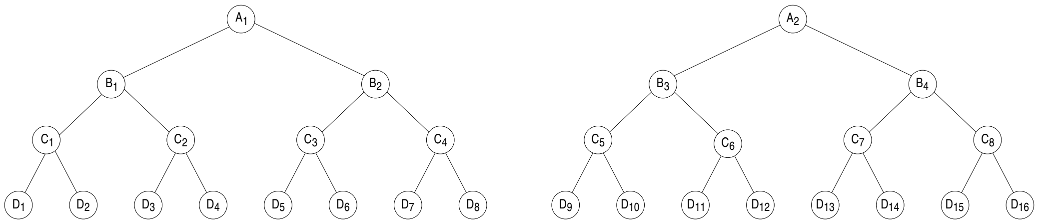

Figure 1: Full forest, before any approximations, for $N_s = 2$ and $N_e = 3$. The labels have no particular significance. See table 1 for information associated with the various nodes.

The model structure is defined by $N_s = 2$ and $N_e = 3$, and the transition matrix is, for illustrative purposes,

$$P = \begin{pmatrix} .99 & .01 \\ .005 & .995 \end{pmatrix}.$$

The general case does not involve anything new, and the discussion would become excessively abstract. The general case is documented in our source code, using the C programming language.

We start by constructing N_s trees (figure 1). Each node represents a partial state history, starting at the roots, corresponding to individual states, and descending to the leaves, which represent metastates, so that the history corresponding to a node of depth d has length $d + 1$ (see the second column of table 1). We store the probability of the partial state history, conditional on the initial state, in each node; these values are built up recursively as the tree is built (see the third column of table 1). In general, all operations that one might think of performing by means of multiple nested loops are, in fact, done by recursive tree traversals.

In practice, we need not build the full tree because we invoke the condition (11) to "prune" the tree as we build it, eliminating any node for which $P < \epsilon_1$ and all of its children. For our example, suppose we choose $\epsilon = 0.001$. Then we actually build the forest in figure 2. It can happen that (11) eliminates all the children of a node without eliminating the node itself; in this case the node is pruned. At the end of the pruning process there are no leaves at levels greater than zero.

After pruning, we multiply the stored probabilities in the leaves by the equilibrium probabilities associated with the roots of the trees to obtain values of $\hat{\alpha}_0$. During the same tree traversal, the means of $y(t)$ conditional

Table 1: Information stored in the nodes of figure 1. "Node" is the label in figure 1. "History" is the sequence of states represented by the node. "P" is the conditional probability of the partial history. The last column indicates whether or not the node is eliminated ("pruned") when $\epsilon_1 = 0.001$. Note that D_5 and D_6 are automatically pruned because C_3 is, and, similarly, D_{11} and D_{12} are eliminated when C_6 is pruned.

Node	History	P	Prune?
A_1	0	1.	no
A_2	1	1.	no
B_1	00	0.99	no
B_2	01	0.01	no
B_3	10	0.005	no
B_4	11	0.995	no
C_1	000	0.9801	no
C_2	001	0.0099	no
C_3	010	0.0005	yes
C_4	011	0.00995	no
C_5	100	0.00495	no
C_6	101	0.00005	yes
C_7	110	0.004975	no
C_8	111	0.990025	no
D_1	0000	0.970299	no
D_2	0001	0.009801	no
D_3	0010	0.0000495	yes
D_4	0011	0.0098505	no
D_5	0100	0.0000495	yes
D_6	0101	0.0000005	yes
D_7	0110	0.00004975	yes
D_8	0111	0.00990025	no
D_9	1000	0.0049005	no
D_{10}	1001	0.0000495	yes
D_{11}	1010	0.00000025	yes
D_{12}	1011	0.00004975	yes
D_{13}	1100	0.00492525	no
D_{14}	1101	0.00004975	yes
D_{15}	1110	0.004950125	no
D_{16}	1111	0.985074875	no

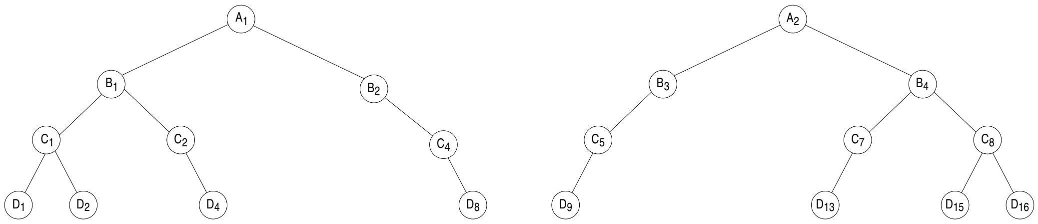

Figure 2: The forest of figure 1 after pruning with $\epsilon = 0.001$.

on the metastate are constructed and stored in the leaves.

It remains to discuss the updating process in which, starting from a forest with $\hat{\alpha}_t$ stored in the leaves, we arrive at a new forest, with the same topology, with $\hat{\alpha}_{t+1}$ in the leaves. Mathematically, we must sum over the oldest state, which is at the roots, to obtain the normalized version of $\tilde{\alpha}_t$, and then we use the last form of (3). All of the index manipulation in (3) will be done automatically by recursive tree traversals. Consider the subtrees rooted at B_1 and B_3. The "sum" of these will become the part of the new tree rooted at A_1 of level greater than zero, and $\tilde{\alpha}_t$ will be stored in its leaves, which are the nodes of level one in the final tree. In general, when "adding" two trees, we add the α's stored in the leaves, except when some leaves are missing because of pruning. Similarly, the part of the new tree rooted at A_2 of level greater than zero is obtained as the sum of the subtrees rooted at B_2 and B_4. It is then straightforward to compute and store the values of α_{t+1} and carry out the normalization process described by (4–6).

3 Examples

We illustrate the computational savings of our method by simulations from three models that have appeared in the ion channel literature. Model I was proposed in (Colquhoun and Hawkes, 1995) for an acetylcholine receptor. When sampled at 10 kHz, the transition matrix of the five state scheme is

$$P_I = \begin{pmatrix} .7373 & .0044 & .0004 & .2325 & .0253 \\ .0001 & .9723 & .0219 & .0053 & .0004 \\ .0002 & .6579 & .1614 & .1588 & .0217 \\ .0012 & .0020 & .0020 & .8142 & .1807 \\ .0000 & .0000 & .0000 & .0009 & .9991 \end{pmatrix} \tag{13}$$

The channel is open in the first two states ($\mathcal{I} = 1$) and closed in the last three ($\mathcal{I} = 0$). We note the fifth is a long lived closed state.

Model II was proposed in (Correa et al., 1992) for a batrachotoxin-modified sodium channel. It too is a five state scheme, which when sampled at 10 kHz yields the transition matrix,

$$P_{II} = \begin{pmatrix} .9903 & .0096 & .0000 & .0000 & .0000 \\ .0577 & .9330 & .0092 & .0001 & .0000 \\ .0017 & .0554 & .9141 & .0274 & .0014 \\ .0001 & .0025 & .0822 & .9152 & .0001 \\ .0000 & .0003 & .0086 & .0001 & .9910 \end{pmatrix} \tag{14}$$

The channel is closed in the first three states and open in the last two. The first closed state and the last open state are particularly long lived, with mean durations of about 100 sampling units.

Model III was used in (Fredkin and Rice, 1992b) and is derived from another model for a batrachotoxin-modified sodium channel (Huang et al., 1984). This model has three states, the first two of which are closed, and when sampled at 10 kHz produces a transition matrix

$$P_{III} = \begin{pmatrix} .9996 & .0004 & .0000 \\ .0090 & .9860 & .0049 \\ .0000 & .0093 & .9907 \end{pmatrix} \tag{15}$$

These models share a feature which makes our approximation schemes effective: many of the entries of the transition matrices are quite small, and the diagonal entries are relatively large, implying that a substantial fraction of metastates have very small probability. Particularly improbable are those with many transitions between different states.

In our simulations we used a digital approximation to an eight pole Bessel filter with a cutoff at 2 kHz, a moving average with coefficients [.0348, .4515, .4556, .0621, -.0064]. Our two noise models were white noise and an autoregressive scheme from (Venkataramanan et al., 1996) with coefficients [1.0, .7152, .4900, .3056, .1427]. The convolution of these two sequences, truncated after eight terms and normalized to sum to one, gave a net composite filter with coefficients [.0131, .1799, .3004, .2341, .1526, .0867, .0305, .0026]. Three different signal to noise ratios were used, the innovation standard deviations being .05, .25, and .75. For each of the three kinetic models, for each of the two noise models, and for each of the three signal to noise level we simulated 100,000 points, or 10 seconds of data. The computations we report were performed on a Sun UltraSparc 2. Our programs were written in C and linked to Matlab[1].

We first discuss the results for the autoregressive noise model with innovation variance $\sigma = .05$. For a composite filter of length eight, the total number of metastates are $5^8 = 390625$ for models I and II and $3^8 = 6561$ for model III. As discussed in the last section, the computational prices to be paid over a model with no filtering and white noise are factors of $5^7 = 78125$ and $3^7 = 2187$. For example, if the likelihood took one second to evaluate with no filtering and white noise (this figure is roughly accurate), it would take approximately 22 hours to evaluate in models I and II allowing for colored noise and filtering.

As explained in the previous section, we can decrease the effective number of metastates, and proportionally the time to evaluate the likelihood, by increasing the parameter ϵ_1. Figure 3a shows the resulting error in approximating the likelihood as a function of the fraction of the number of metastates remaining after pruning. (The actual log likelihoods were of order 10^5 for each of the three models.) For example, if the number of metastates of model I is reduced by a factor of 362, the resulting error in the log likelihood is 3.15 out of 1.57×10^5. For model III reduction of the number of metastates by a factor of 65 resulted in an error of 3.37 out of 1.57×10^5. Although these reductions are large, even with them computational times are quite substantial. For example, after the number of metastates of model I is reduced by factor of 362, 1077 effective metastates still remain. In fact, evaluation of the likelihood allowing for filtering and colored noise, pruning the number of effective metastates to 1077, took 1687 seconds, as compared to 1.3 seconds for evaluation of the likelihood of a model with no filtering and white noise. For model III, the computation of the likelihood took 158 seconds after reduction of the number of metastates by a factor of 65.

Without specification of the use of the approximate log likelihood, it is difficult to determine an acceptable level of error, but we suggest the following heuristic as a guide. Suppose that $\hat{\theta}$ is the maximum likelihood estimate of an m dimensional vector of rate constants. A standard large sample theory result (Cox and Hinkley, 1974) is that an approximate $100(1-\alpha)$ % confidence region for θ is $\{\theta | 2(\ell(\hat{\theta}) - \ell(\theta)) \leq \chi^2_m(\alpha)\}$ where $\ell(\theta)$ is the log likelihood and $\chi^2_m(\alpha)$ is the upper α percentage point of the chi-square

[1]The MathWorks, Inc. Natick, Massachusetts.

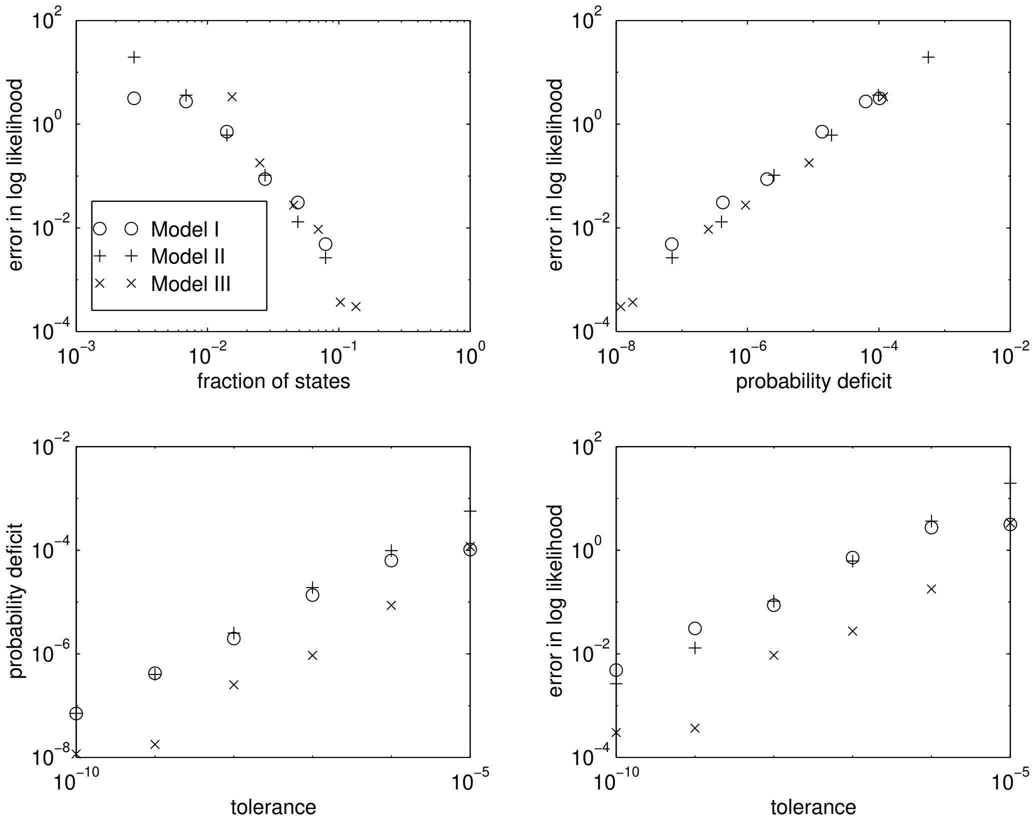

Figure 3: (a) The error in the approximation to the log likelihood as a function of the fraction of meta-states retained. (b) The error in the approximation to the log likelihood as a function of the total equilibrium probability of the metastates pruned from the tree (the probability deficit). (c) The probability deficit as a function of the tolerance, ϵ_2. (d) The error in the approximation to the log likelihood as a function of the tolerance, ϵ_2.

distribution with m degrees of freedom. For example, the underlying kinetic model for model II has six free rate constants which determine the rate matrix Q from which $P_{II} = \exp(Q\Delta t)$ was found. The upper 5% point of the chi-square distribution with six degrees of freedom is 12.59. Thus the effect of an approximation error of order one in the log likelihood is comparable to the variation in the likelihood due to parameter uncertainty. The effect of the approximation error on optimization is discussed in the concluding section.

As described in the previous section, we prune the number of effective metastates by setting the tolerance parameter ϵ_1. Let the sum of the equilibrium probabilities of the metastates which have been discarded be termed the "probability deficit." Figure 3b shows that the error in the log likelihood is proportional to the probability deficit with a constant of proportionality of order 10^4. The probability deficit induced by pruning of the degree discussed in the examples above is roughly of order 10^{-5}, which we believe is negligble when viewed from a broad perspective in which the model itself is a crude approximation to physical reality. Figure 3c shows how the probability deficit is determined by the tolerance. To complete the picture, figure 3d shows how the error in the log likelihood is determined by the tolerance, ϵ_1. From these figures we see that the tolerance, the probability deficit, and the fraction of metastates remaining are all equivalent ways of specifying the amount of pruning. We have found it algorithmically most natural to control the amount of pruning by setting ϵ_1, since the pruning can be accomplished as the forest of metastates is traversed.

Very similar results were found at the lower signal to noise ratios in that the errors induced in estimating the log likelihood by using a small fraction of the total number of metastates were comparable in order of magnitude to those described above for the three models. For example for model I, with $\sigma = .75$, the total log likelihood was -1.13×10^5 and the error when 1077 metastates were used was 1.00.

We next briefly contrast the results discussed above to those obtained when a low pass filter is used, but noise is white rather than colored. The length of the filter is thus five rather than eight, and the relative gains are smaller. On an absolute scale, the computations are less forbidding. For models I and II and a filter of length five, there are $5^5 = 3125$ metastates as compared to $5^8 = 390625$ for a filter of length 8, and gains by factors of about 10 are possible while incurring an error of order one.

Generally, as the length of a filter in increased, the fraction of metastates needed to maintain a given probability deficit decreases rapidly. Figure 4a shows this phenomena for model I and various filter lengths. However, the total number of remaining metastates, and hence the time to evaluate the likelihood, continues to increase, as shown in figure 4b. It thus appears that additional computational strategies, such as distributing the computations over a network of workstations, are still needed for very long filters.

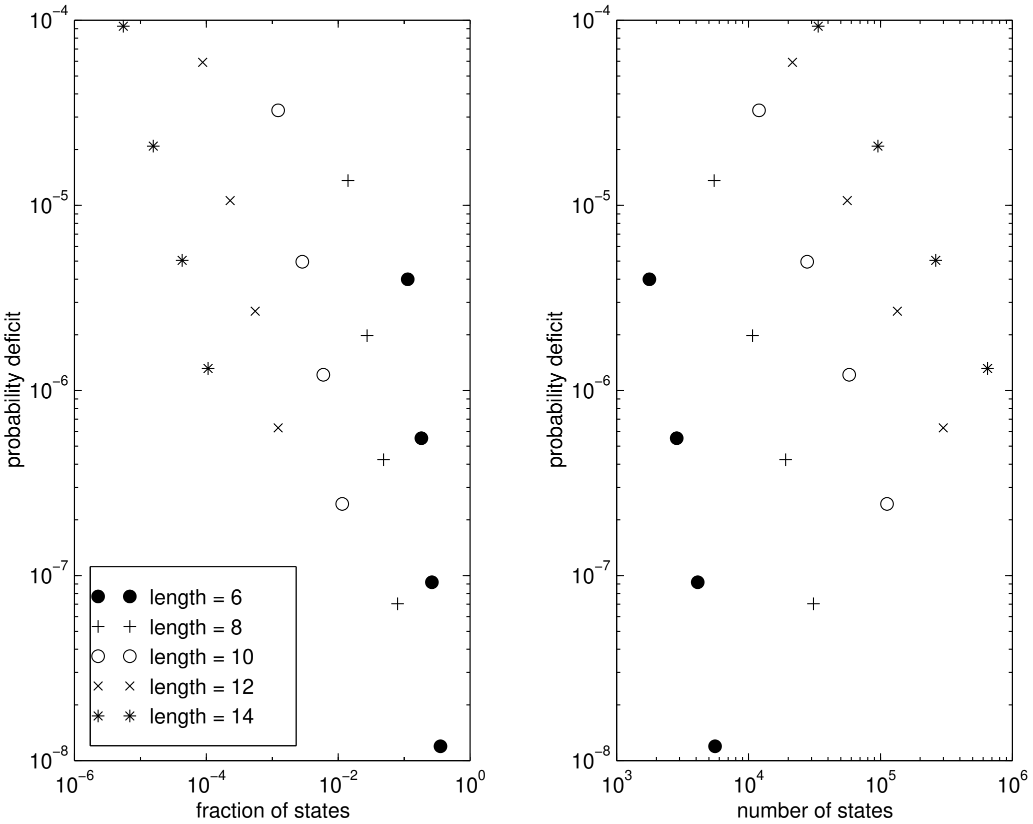

Figure 4: (a) The probability deficit for model I as a function of the fraction of metastates retained for various filter lengths. (b) The probability deficit as a function of the number of metastates retained for filter lengths as in (a).

Finally, we discuss the savings that can be accomplished by imposing the second tolerance, $\epsilon_2 > 0$. In our simulations, we found that with $\epsilon_1 > 0$, decreases in computation time of factors of two to three, with little additional inaccuracy in the approximated log likelihood, could be accomplished by setting ϵ_2 to small values, such as 10^{-9}, when the signal to noise ratio was high. Further increasing ϵ_2 did not result in substantial consequent savings as most metastates that were *a posteriori* unlikely had already been eliminated. At lower signal to noise ratios the effectiveness of ϵ_2 decreased and became insubstantial at $\sigma = .75$. This is to be expected, since using the second tolerance eliminates at each time point metastates which are *a posteriori* unlikely given the observed data, and with a high noise level the data are relatively uninformative.

As an example, for model II with $\sigma = .05$, setting $\epsilon_1 = 10^{-6}$ reduced the number of metastates by a factor of 200—from 390625 to 1946. With $\epsilon_2 = 0$ the error in the log likelihood of 3.63; setting $\epsilon_2 = 10^{-10}$ reduced the computation time by a further factor of 2.1, giving a net reduction by a factor of about 400, while the additional error in the log likelihood was less than 10^{-4}. With this setting of ϵ_2, the average number of metastates discarded per time point was 765 (out of 1946). Examination of the results revealed that when the channel was closed about 750 metastates were typically discarded and when it was open (which was less frequent) about 1150 were discarded.

4 Discussion

We have explained and demonstrated methods which provide dramatic computational gains in the evaluation of the likelihood of a hidden Markov model for single-channel recordings contaminated by filtering and colored noise. These gains are achieved by discarding the contributions to the likelihood from metastates that are either *a priori* or *a posteriori* unlikely. We have found it convenient and effective to organize the computations in a tree structure, but other approaches are possible. With our implementation the greatest gains are made by discarding metastates which are *a priori* unlikely since the pruned branches of the tree are subsequently never traversed during the iterated passes through it. Our methods can be applied to approximate not only the likelihood but also its gradient and posterior probabilities.

In this paper we have concentrated on efficient approximate evaluation of the likelihood but not directly on its maximization. Many additional issues come into play in this latter endeavor, but in any case evaluation of the likelihood function is a key component. Other important components include the choice of starting values and the search strategy. For choice of starting values it may be effective to maximize the likelihood or an approximation to it on a relatively small segment of data. When working with the full data set, one could initially use these maximizers as starting values and relatively large tolerances to find a new maximum. The tolerances could then be decreased and the process continued until there was little change in the maximizers. Since our approximations work by discarding metastates, they produce lower bounds to the likelihood; the success achieved in maximizing such lower bounds rather than the likelihood itself depends in part upon how uniform the bounds are over the relevant parameter space. We have not yet investigated this question, but the observed proportionality of the error in the log likelihood to the probability deficit provides some reason for optimism that maintaining a fairly constant probability deficit as the parameters change would produce nearly uniform lower bounds. Given the time that it takes to evaluate the likelihood function, it is clearly important to use a search strategy that entails a minimum number of function evaluations.

Although we have developed and illustrated the methods in the context of single-channel recordings, we believe that they may have relevance to other phenomena modeled by hidden Markov in which the dimensionality of the state space makes exact computation of the likelihood prohibitive or impractical. Within the context of the statistical analysis of patch clamp recordings, we believe that our methods will be especially effective in evaluating the likelihood of superpositions of independent channels. Such superpositions produce a very high dimensional state space which has hindered the successful application of otherwise promising hidden Markov model techniques (Albertson and Hansen, 1994).

Our code is written in C to be driven by Matlab, and we will be pleased to share it with anyone who is interested.

References

Aho, A. V., Hopcroft, J. E., and Ullman, J. D. (1974). *The Design and Analysis of Computer Algorithms.* Addison-Wesley Publishing Company.

Albertson, A. and Hansen, U.-P. (1994). Estimation of kinetic rate constants from multi-channel recordings by a direct fit of the time series. *Biophysical Journal*, 67:1393–1403.

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

Chung, S. H., Moore, J. B., Xia, L., Premkumar, L. S., and Gage, P. W. (1990). Characterization of single channel currents using digital signal processing techniques based on hidden markov models. *Philosophical Transactions of the Royal Society of London, Series B*, 329:265–285.

Colquhoun, D. and Hawkes, A. G. (1995). The principles of the stochastic interpretation of ion-channel mechanisms. In Sakmann, B. and Neher, E., editors, *Single-Channel Recording*, chapter 18. Plenum Press.

Correa, A. M., Benzanilla, F., and Latorre, R. (1992). Gating kinetics of batrachotoxin-modified Na^{+} channels in the squid giant axon. *Biophysical Journal*, 61:1332–1352.

Cox, D. R. and Hinkley, D. V. (1974). *Theoretical Statistics*. Chapman and Hall.

Fletcher, R. (1987). *Practical Methods of Optimization*. John Wiley and Sons, second edition.

Fredkin, D. R. and Rice, J. A. (1992a). Bayesian restoration of single channel patch clamp recordings. *Biometrics*, 48:427–448.

Fredkin, D. R. and Rice, J. A. (1992b). Maximum likelihood estimation and identification directly from single-channel recordings. *Proceedings of the Royal Society of London, Series B*, 249:125–132.

Ghahramani, Z. and Jordan, M. (1996). Factorial hidden markov models. In Touretzky, D., Mozer, M., and Hasselmo, M., editors, *Advances in Neural Information Processing Systems*, volume 8. MIT Press.

Huang, L., Moran, N., and Ehrenstein, G. (1984). Gating kinetics of batrachotoxin-modified sodium channels in neuroblastoma cells determined from single-channel measurements. *Biophysical Journal*, 45:313–324.

Juang, B. and Rabiner, L. (1990). The segmental k-means algorithm for estimating parameters of hidden markov models. *Institute of Electrical and Electronic Engineers Transactions on Acoustic, Speech, and Signal Processing*, 38:1639–1641.

Krogh, A., Mian, S., and Haussler, D. (1994). A hidden markov model that finds genes in e. coli dna. *Nucleic Acids Research*, 22:4769–4778.

Qin, F., Chen, A., Auerbach, A., and Sachs, F. (1994). Extracting channel kinetic parameters using hidden markov techniques. *Biophysical Journal*, 66:392.

Rabiner, L. (1989). A tutorial on hidden markov models and selected applications in speed processing. *Proceedings of the Institute of Electrical and Electronic Engineers*, 77:257–285.

Venkataramanan, L., Walsh, J. L., Kuc, R., and Sigworth, F. J. (1996). Identification of hidden markov models for ion channel currents containing colored background noise. Manuscript.

Viterbi, J. (1967). Error bounds for convolution codes an an asymptotically optimal decoding algorithm. *Institute of Electrical and Electronic Engineers Transactions on Information Theory*, 13:260–269.