Statistical analysis of neural data:
Continuous-space models (First 2/3)

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

April 27, 2009

Contents

A "state-space model" is essentially a hidden Markov model in which the hidden component q_t has a continuous state space. As we will see in this chapter, state-space methods provide a coherent framework for modeling stochastic dynamical systems that are measured or observed with error; applications to neuroscience are quite numerous. The famous "Kalman filter" model is the simplest (and most analytically-tractable) case of the state-space model, but this framework can be generalized in a number of useful directions.

1 Autoregressive models and Kalman filter models are Gaussian Markov and hidden Markov models, respectively

As usual in an HMM, to specify the model we must identify two components: 1) the dynamics of the hidden state $p(q_{t+1}|q_t)$ and 2) the observation (emission) probabilities $p(y_t|q_t)$. In the simplest state-space model, the Kalman model, both of these components are additive and Gaussian:

$$q_{t+1}|q_t \sim \mathcal{N}(Aq_t + a, C_q),$$

and

$$y_t|q_t \sim \mathcal{N}(Bq_t + b, C_y),$$

where the system parameters A, B, C_q, C_y are fixed matrices (or scalars in the simplest case that q_t and y_t are one-dimensional) and a and b are vectors which set the mean of q_t and y_t. The matrix A sets the dynamics of q_t, while B controls the dependence of the observations y_t on q_t; C_q and C_y set the covariance of q_t and y_t given q_{t-1} and q_t, i.e., the noisiness of the dynamics and the noisiness of the observations, respectively. As we will see, this Kalman model serves as a base from which we can generalize in a number of directions; thus we will lay out the theory of this model in some detail.

The simplicity of this model is due in large part to its close relationship with the classical autoregressive (AR) process from the statistical theory of time-series analysis. In particular, q_t here may be written as a standard "AR(1)" process[1]:

$$q_{t+1} = Aq_t + \epsilon_t, \tag{1}$$

where ϵ_t are i.i.d. Gaussian random vectors with mean zero and covariance C_q. (We have neglected the mean a here for simplicity; there is no loss of generality, as may be seen by a simple shift of the coordinate system.) Thus, before we dive into the details of inference and fitting with the Kalman model, it is worth discussing the basics of this simpler AR model; in particular, the problem of fitting the parameters in the (partially observed) Kalman model may be considered a generalization of the problem of fitting the (fully observed) AR model.

First we write out some basic facts. Much of the theory of AR processes relies heavily on the superposition principle (linearity): q_t may be written as a sum of terms, with each term corresponding to the input to the system at past times t:

$$q_{t+1} = Aq_t + \epsilon_t = A(Aq_{t-1} + \epsilon_{t-1}) + \epsilon_t = \sum_{i=0}^{\infty} A^i \epsilon_{t-i}.$$

[0]Thanks to Yashar Ahmadian, Baktash Babadi, Emery Brown, Ana Calabrese, Quentin Huys, Shinsuke Koyama, Jayant Kulkarni, Kamiar Rahnama Rad, Michael Vidne, Joshua Vogelstein, and Wei Wu for help preparing these notes.

[1]The 1 in "AR(1)" refers to the number of lags in the equation; we will discuss examples of the multiple-lag case below.

This is the moving average (MA) representation of the AR process; we interpret the sequence $(I, A, A^2, \ldots)$ as a linear filter which is applied to the sequence (ϵ_t) in order to obtain q_t. Note that this MA representation only makes sense if the sum exists, i.e., $A^i \to 0$; if A is diagonalizable, an equivalent condition is that all of the eigenvalues of A are less than one. Also note that each input ϵ_t continues to influence q_s for s arbitrarily far into the future: thus in the engineering literature AR processes are often referred to as infinite impulse response (IIR) systems.

It is straightforward to compute the conditional moments and correlation function in this model if we use the representation

$$q_t|q_0 = A^t q_0 + \sum_{i=0}^{t-1} A^{t-i-1}\epsilon_i,$$

where q_0 is a known initial state. For example, we see that

$$E(q_t|q_0) = A^t q_0 + E\left(\sum_{i=0}^{t-1} A^{t-i-1}\epsilon_i\right) = A^t q_0,$$

since ϵ_t has zero mean, and

$$Cov(q_t|q_0) = Cov\left(\sum_{i=0}^{t-1} A^{t-i-1}\epsilon_i\right) = \sum_{i=0}^{t-1} A^{t-i-1} Cov(\epsilon_i)(A^{t-i-1})^T = \sum_{i=0}^{t-1} A^{t-i-1} C_q (A^{t-i-1})^T,$$

since ϵ_t are i.i.d. The auto- and cross-covariance functions follow similarly; both of these functions decay exponentially, as powers of A.

Thus we see that the first and second moments of q_t are very easy to calculate in this model. In fact, the full distribution of Q (not just the moments) is easy to obtain if we note that Q is in fact a Gaussian vector, since as we have seen Q may be written as a linear function of a sequence of Gaussian random variables; in other words, we may write

$$Q = \mathbf{A}\epsilon,$$

where ϵ is the vector of Gaussian variables ϵ_t and $\mathbf{A}$ is a suitable matrix (this is a block-lower-triangular convolution matrix in which the i-th block from the diagonal is given by A^i). Since a Gaussian distribution is uniquely specified by its first and second moments, the calculations above uniquely identify this Gaussian distribution; in the matrix form above, we have that

$$Q \sim \mathcal{N}(0, \mathbf{A}Cov(\epsilon)\mathbf{A}^T),$$

where $Cov(\epsilon)$ is block-diagonal with blocks C_q.

How can we fit the model parameters? We start, as always, by writing down the loglikelihood,

$$\log p(Q|A, C_q) = \log p(q_1) - \frac{1}{2}\sum_{t=2}^{T}\left(\log[(2\pi)^p|C_q|] + (q_t - Aq_{t-1})^T C_q^{-1}(q_t - Aq_{t-1})\right).$$

The first term (which corresponds to our initial density π in the discrete-space HMM setting) can often be ignored, since the sum will clearly dominate if T is sufficiently large. At least in

the one-dimensional case it is clear that this is a least-squares problem we have encountered before: the MLE for A is obtained by solving the regression implicit in equation (1), and the MLE for C_q corresponds to the residual variance of this regression. Now we see the reason for the term "autoregressive": to estimate the parameters we form a regression equation in which the past state q_{t-1} is regressed onto the current state q_t. The case of multidimensional q_t is solved in a similar manner (we will describe the detailed solution of a more general case shortly).

In sum, it is easy to compute conditional means and variances and to fit the parameters of this autoregressive model if q_t is fully observed.

1.1 Example: voltage smoothing and interpolation; inferring biophysical parameters

Imagine we are observing a neuron's somatic voltage $V(t)$ as a function of time. In many cases, we might not be able to observe this voltage completely: there may be observation noise, or we may only be able to make observations intermittently (this is particularly true in the context of voltage-sensitive imaging (Nuriya et al., 2006), in which case there are limits to how quickly the CCD camera or photomultiplier can sample the image data). It is natural to ask how we might optimally recover the true voltage given a noisy, intermittently sampled observed voltage. As we will see, in some settings we can attack this question directly in the Kalman framework; however, once again, before we dive into the more challenging partially-observed case, it makes sense to build up some intuition in the fully-observed case. We start by discussing a number of AR-type models for the voltage process $V(t)$.

The simplest model is of AR(1) form:

$$V(t+1) = V(t) + dt\,(-gV(t) + b + kI(t)) + \epsilon_t, \quad \epsilon_t \sim \mathcal{N}(0, \sigma^2 dt). \tag{2}$$

This is a one-dimensional model: all of a neuron's highly complex, nonlinear biophysical processing is reduced to a single stochastic equation (Paninski et al., 2004b; Jolivet et al., 2004). Here $I(t)$ is the current input to the neuron; g is the membrane conductivity ("leakiness," or "forgetfulness"), which is constrained to be nonnegative (since the resistance can not be negative, on physical grounds); the noise term ϵ_t is white Gaussian noise, and σ represents the noisiness of this integration process. (The dt term attached to σ^2 is there to ensure a well-defined continuous limit as $dt \to 0$; we will discuss this in more depth in the next chapter.) In general we think of all of these terms — $I(t)$, g, and ϵ_t — as corresponding to some kind of lumped current input, or membrane conductance, or synaptic and membrane noise — in reality, of course, all of these parameters vary both temporally and as a function of spatial location on the neuronal membrane. Thus, as usual, this model is a caricature, albeit a useful caricature, and there are many ways to inject more biophysical realism into the model (as we will see below). We should also note the more standard reversal potential form of this model,

$$V(t+1) = V(t) + dt\big(g(V_L - V(t)) + kI(t)\big) + \epsilon_t,$$

where the leak potential $V_L = b/g$; of course these two model forms are equivalent, but the form in (2) is slightly more convenient from a statistical point of view, as we will see shortly.

We may fit this model exactly as described above. We rewrite the model in standard regression form:

$$V(t+1) - V(t) = \theta_1 V(t) + \theta_2 + \theta_3 I(t) + \epsilon_t,$$

or in vector form,

$$Y = X\theta + \epsilon,$$

with the design matrix

$$X_t = \begin{pmatrix} V(t) & 1 & I(t) \end{pmatrix}$$

and the observed variable $Y_t = V(t+1) - V(t)$. Now we may choose

$$\hat{\theta}_{ML} = \arg \min_{\theta:\theta_1 \leq 0} ||Y - X\theta||_2^2;$$

this is a standard regression problem with linear inequality constraints, and may be solved with the usual quadratic programming techniques. Finally, we identify

$$(\hat{g}, \hat{b}, \hat{k}) = \frac{1}{dt}(-\hat{\theta}_1, \hat{\theta}_2, \hat{\theta}_3),$$

and $\hat{\sigma}^2 dt$ as the residual variance of the regression.

Now, as usual, we may generalize the model in a number of different ways, simply by modifying the design matrix X:

- The AR(p) (p-lag) model is:

$$V(t+1) = V(t) + dt \left(\sum_{i=0}^{p-1} a_i V(t-i) + b + kI(t) \right) + \epsilon_t,$$

  with corresponding design matrix

$$X_t = \begin{pmatrix} V(t) & V(t-1) & \dots & V(t-p+1) & 1 & I(t) \end{pmatrix}.$$

  This allows us to capture resonant effects in the voltage dynamics (Izhikevich, 2001; Badel et al., 2008a), instead of just the "leaky" exponentially-decaying dynamics observed in the AR(1) model.

- The AR(1) model with filtered inputs:

$$V(t+1) = V(t) + dt \left(-gV(t) + b + \sum_{i=1}^{p} k_i I_i(t) \right) + \epsilon_t,$$

$$X_t = \begin{pmatrix} V(t) & 1 & I_1(t) & I_2(t) & \dots & I_p(t) \end{pmatrix}.$$

  Here we model the input current as a weighted sum of elementary currents,

$$I(t) = \sum_{i=1}^{p} k_i I_i(t) :$$

  for example, if $I_i(t)$ is chosen to be the time-delayed current $I_i(t) = I(t-i+1)$, then the weights k_i implement a linear temporal filter. More generally $I_i(t)$ could implement stimulus-dependent and/or spike history terms (Paninski et al., 2004b; Jolivet et al., 2004; Paninski et al., 2007). See Fig. 1 for an application to intracellular data from a cortical neuron recorded *in vitro*.

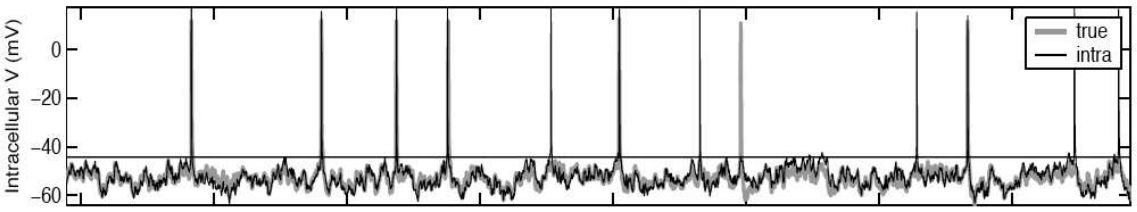

Figure 1: Comparison of true vs. predicted voltage $V(t)$ in a cortical neuron recorded *in vitro*, stimulated with Gaussian white noise injected current $I(t)$ (not shown). Predicted voltage generated by an AR(1) model with time-filtered input $\sum_{i=1}^{p} k_i I(t-i)$, with spikes generated on each threshold crossing (threshold indicated by solid horizontal line); predicted voltage (gray trace) matches true voltage (black) fairly well. See (Paninski et al., 2004b) for details and (Jolivet et al., 2004) for a similar application.

- Another important generalization of the AR(1) model is to allow the conductance g to vary as a function of time (Stevens and Zador, 1998; Jolivet et al., 2004; Badel et al., 2008c; Badel et al., 2008b); for example, we know on biophysical grounds that the membrane conductance following a spike or presynaptic release should increase transiently (Koch, 1999). We may implement this easily as

$$V(t+1) = V(t) + dt\left(-V(t)\sum_{i=1}^{p} a_i g_i(t) + b + kI(t)\right) + \epsilon_t,$$

$$X_t = \big(g_1(t)V(t) \;\; g_2(t)V(t) \;\; \ldots \;\; g_p(t)V(t) \;\; 1 \;\; I(t)\big).$$

  The only twist here is that instead of finding the optimal parameters θ under the single linear constraint $g \geq 0$, now we need to satisfy multiple linear constraints, namely

$$g(t) \geq 0 \;\; \forall t.$$

  (Each of these constraints is linear in the parameters $(a_1, a_2, \ldots a_p, b, k)$, since

$$g(t) = \sum_i a_i g_i(t) \geq 0.$$

  This is still a quadratic optimization problem with linear constraints in θ and may therefore be solved via standard quadratic programming techniques.

- Finally, we may consider simple AR-type models in which $V(t)$ evolves nonlinearly:

$$V(t+1) = V(t) + dt\left(\sum_{i=1}^{p} a_i f_i[V(t)] + kI(t)\right) + \epsilon_t,$$

$$X_t = \big(f_1[V(t)] \;\; f_2[V(t)] \;\; \ldots \;\; f_p[V(t)] \;\; I(t)\big).$$

  This allows us to fit the dynamical nonlinearity

$$V(t+1) = V(t) + dt\big(f[V(t)] + kI(t)\big) + \epsilon_t$$

where we represent the nonlinearity $f(V)$ in some basis set

$$f(V) = \sum_i a_i f_i(V);$$

note that choosing $f_1(V) = 1, a_1 = b, f_2(V) = V$, and $a_2 = -g$ recovers the linear model. See section 3.2 below for a nonparametric approach to this model.

Of course, the various components described above — multiple lags, time-varying currents, conductances, and nonlinear dynamics — may be incorporated in a single model, with a suitably defined design matrix X; we have presented these models separately just for the sake of clarity.

One last important application of these ideas is discussed in (Huys et al., 2006): imagine we have observations not just at a single location in the neuron, but instead at multiple contiguous locations (i.e., multiple neighboring "compartments" on the dendritic tree). A simple AR-type model for the evolution of the voltage $V_i(t)$ at these multiple locations i is

$$V_i(t+1) = V_i(t) + dt \left(-g_i V_i(t) + b_i + k_i I_i(t) + \sum_{j \in N(i)} a_{ij}[V_j(t) - V_i(t)] \right) + \epsilon_t, \qquad (3)$$

where $N(i)$ represents the "neighbor set" of a given compartment i (i.e., those compartments j which physically abut i), and a_{ij} represent the intercompartmental conductance between i and j. There are typically two neighbors for every compartment, since dendrites are locally linear, but occasionally a compartment may have one neighbor (if the compartment is at the terminus of the dendritic branch) or more than two (if the compartment is at a branch point).

To fit this model we may define

$$X_i(t) = dt\big(-V_i(t) \;\; 1 \;\; I_i(t) \;\; \{[V_j(t) - V_i(t)]\}_{j \in N(i)} \big),$$

$$Y_i(t) = V_i(t+1) - V_i(t),$$

and

$$\theta_i = \big(g_i \;\; b_i \;\; k_i \;\; \{a_{ij}\}_{j \in N(i)} \big)^T;$$

now we want to minimize the quadratic function

$$\sum_i ||X_i \theta_i - Y_i||_2^2 \qquad (4)$$

of $\theta = \{\theta_i\}$ under the set of linear constraints

$$g_i \geq 0$$

$$a_{ij} \geq 0$$

$$a_{ij} = a_{ji},$$

where the last symmetry constraint is due to the fact that the conductance from one compartment i to another compartment j must be the same as the conductance in the opposite direction.

This is, once again, a quadratic programming problem, albeit one with many parameters. We can reduce the number of parameters significantly if we assume that each parameter

is relatively constant as a function of location, i.e., $k_i = k, b_i = b$, and so on (this is a reasonable assumption if the compartments are of roughly equal size, since it is known that the compartmental resistance varies (inversely) with the size of the compartment (Koch, 1999; Dayan and Abbott, 2001)). This assumption may be imposed strictly (by redefining θ in this lower dimensional space of single global parameters instead of multiple local parameters, and rewriting the quadratic form in equation (4) in terms of this reduced θ), or in a "softer" manner by retaining the original parameterization but adding penalty terms, e.g.

$$\sum_i ||X_i\theta_i - Y_i||_2^2 + \sum_{i,j \in N(i)} \lambda_{ij} ||\theta_i - \theta_j||_2^2,$$

where $\lambda_{ij} > 0$ sets the size of these penalty terms. See (Huys et al., 2006) for full details.

1.2 We may perform inference in the Kalman model either via the forward-backwards recursion or by direct optimization methods

Now we will turn to the more challenging Kalman setting, in which y_t is fully observed but q_t is not. As in any HMM, in the Kalman model we are interested in computing the posterior expectations $E(q_t|Y)$ and related quantities. We will discuss two ways to approach this computation here.

We will begin by describing the forward-backward method, which is most closely related to the methods we discussed in the previous chapter. The forward step for the Kalman model is a straightforward adaptation of the forward step for the standard HMM: we have the recursion

$$p(q_t, Y_{1:t}) = \left(\int p(q_{t-1}, Y_{1:t-1}) p(q_t|q_{t-1}) dq_{t-1} \right) p(y_t|q_t),$$

which is the continuous version of the "transition forward, then incorporate the observation" update we saw in the discrete setting. The special feature in the Kalman setting is that, if the initial density $p(q_0)$ is a (weighted) Gaussian, then all the densities in sight remain (weighted) Gaussian:

$$\begin{aligned}
p(q_t, Y_{1:t}) &= \left(\int p(q_{t-1}, Y_{1:t-1}) p(q_t|q_{t-1}) dq_{t-1} \right) p(y_t|q_t) \\
&= \left(\int w_{t-1}^f G_{\mu_{t-1}^f, C_{t-1}^f}(q_{t-1}) G_{Aq_{t-1}, C_q}(q_t) dq_{t-1} \right) G_{Bq_t, C_y}(y_t) \\
&= w_{t-1}^f G_{E(q_t|Y_{1:t-1}), C(q_t|Y_{1:t-1})}(q_t) G_{Bq_t, C_y}(y_t) \qquad (5) \\
&= w_t^f G_{\mu_t^f, C_t^f}(q_t). \qquad (6)
\end{aligned}$$

Here the weight w_t^f represents $p(Y_{1:t})$ and the Gaussian density $G_{\mu_t^f, C_t^f}(q_t)$ represents the forward conditional distribution $p(q_t|Y_{1:t})$. This density has covariance

$$C_t^f = \left[C(q_t|Y_{1:t-1})^{-1} + B^T C_y^{-1} B \right]^{-1}$$

(interestingly, this covariance does not depend on the observed data and may therefore be precomputed before we make any observations) and mean

$$\mu_t^f = C_t^f \left[C(q_t|Y_{1:t-1})^{-1} E(q_t|Y_{1:t-1}) + B^T C_y^{-1} (y_t - b) \right];$$

the forward weight w_t^f is updated according to

$$\begin{aligned}\log w_t^f &= \log w_{t-1}^f + \frac{1}{2}\Big(\log|C_t^f| - \log|C(q_t|Y_{1:t-1})| - \log|C_y| \\ &\quad + \mu_t^{fT}(C_t^f)^{-1}\mu_t^f - E(q_t|Y_{1:t-1})^T C(q_t|Y_{1:t-1})^{-1} E(q_t|Y_{1:t-1}) - (y_t-b)^T C_y^{-1}(y_t-b)\Big),\end{aligned}$$

with the "one-step" mean

$$E(q_t|Y_{1:t-1}) = A\mu_{t-1}^f + a$$

and covariance

$$C(q_t|Y_{1:t-1}) = AC_{t-1}^f A^T + C_q,$$

and we have made the abbreviations

$$\mu_t^f = E(q_t|Y_{1:t})$$

and

$$C_t^f = C(q_t|Y_{1:t})$$

for the mean and covariance of the forward distribution $p(q_t|Y_{1:t})$. These formulas for w_t^f, μ_t^f, and C_t^f may be derived by taking logs of the Gaussian multiplicands in equation (5), gathering the quadratic terms, and completing the square. Note that, as usual, we may obtain the full likelihood easily from the output of this forward recursion: $p(Y_{1:T}) = w_T^f$.

It is possible to derive the backward step along exactly the same lines discussed above; we will see a few applications of this approach below. However, the "coupled" variant of the forward-backward method we introduced in the last chapter turns out to be more flexible here, because when the forward dynamics are stable (i.e., the matrix norm $||A||$ is less than one), the backward dynamics are unstable, in the sense that the backward mean $E(q_t|Y_{t+1:T})$ or covariance $C(q_t|Y_{t+1:T})$ may be unbounded (or fail to exist) as T becomes large.

Thus, we use the following backwards recursion, given the forward probabilities (we described a similar method for sampling from discrete HMMs in the last chapter):

$$\begin{aligned}p(q_t, q_{t+1}|Y) &= p(q_{t+1}|Y)p(q_t|q_{t+1}, Y) \\ &= p(q_{t+1}|Y)p(q_t|q_{t+1}, Y_{1:t}) \\ &= p(q_{t+1}|Y)\frac{p(q_t, q_{t+1}, Y_{1:t})}{p(q_{t+1}, Y_{1:t})} \\ &= \frac{p(q_{t+1}|Y)p(q_{t+1}|q_t)p(q_t, Y_{1:t})}{p(q_{t+1}, Y_{1:t})} \\ &= \frac{G_{\mu_{t+1}^s, C_{t+1}^s}(q_{t+1})G_{Aq_t, C_q}(q_{t+1})G_{\mu_t^f, C_t^f}(q_t)}{G_{E(q_{t+1}|Y_{1:t}), C(q_{t+1}|Y_{1:t})}(q_{t+1})} \\ &= G_{E(q_t, q_{t+1}|Y), C(q_t, q_{t+1}|Y)}(q_t, q_{t+1}).\end{aligned} \tag{7}$$

The pairwise means and covariances on the last line can be computed using the same complete-the-squares method we used to obtain the forward moments; see e.g. (Shumway and Stoffer, 2006) or (Minka, 1999) for a couple different derivations. We will skip the (somewhat lengthy)

details here and simply list the results[2]: we obtain the "smoothed" mean

$$\mu_t^s = E(q_t|Y) = \mu_t^f + J_t(\mu_{t+1}^s - A\mu_t^f)$$

and covariance

$$C_t^s = C(q_t|Y) = C_t^f + J_t\left[C_{t+1}^s - C(q_{t+1}|Y_{1:t})\right]J_t^T,$$

where we have made the abbreviation

$$J_t = C_t^f A^T\left[C(q_{t+1}|Y_{1:t})\right]^{-1}.$$

The pairwise covariance has the simple form

$$C(q_t, q_{t+1}|Y) = \begin{pmatrix} C_t^s & C_{t+1}^s J_t^T \\ J_t C_{t+1}^s & C_{t+1}^s \end{pmatrix}. \tag{8}$$

(Once again, note that these covariances do not depend on the observations Y, and may therefore be precomputed; the means, on the other hand, clearly depend on Y.)

Thus, as in the forward-backward sampler, we may run the forward algorithm to obtain μ_t^f, C_t^f for $1 \le t \le T$, then initialize the recursion

$$\mu_T^s = \mu_T^f$$
$$C_T^s = C_T^f$$

and propagate backwards to obtain all of the pairwise probabilities $p(q_t, q_{t+1}|Y)$. As in the discrete setting, the smoother only depends on the observations Y through the forward probabilities $p(q_t|Y_{1:t})$; that is, once we have obtained the forward means and covariances μ_t^f and C_t^f, we may compute the smoothed quantities μ_t^s and C_t^s with no further knowledge of Y. Also note that $Cov(q_t) \ge Cov(q_t|Y_{1:t}) \ge Cov(q_t|Y_{1:T})$: the more data we observe, the smaller our posterior uncertainty.

An alternate approach for computing $E(Q|Y)$ is based on matrix methods. Since (Q, Y) may be written as a linear function of a sequence of Gaussian random variables, (Q, Y) itself form a Gaussian random vector. This implies that the posterior mean of Q given Y, $E(Q|Y)$, is in fact equal to the MAP solution:

$$\begin{aligned}
E(Q|Y) &= \arg\max_Q p(Q|Y) \\
&= \arg\max_Q \log p(Q, Y) \\
&= \arg\max_Q \left(\log p(q_1) + \sum_{t=2}^T \log p(q_t|q_{t-1}) + \sum_{t=1}^T \log p(y_t|q_t)\right) \\
&= \arg\max_Q \Bigg[-\frac{1}{2}\Bigg((q_1 - E(q_1))^T C(q_1)^{-1}(q_1 - E(q_1)) + \sum_{t=2}^T (q_t - Aq_{t-1})^T C_q^{-1}(q_t - Aq_{t-1}) \\
&\qquad + \sum_{t=1}^T (y_t - Bq_t)^T C_y^{-1}(y_t - Bq_t)\Bigg)\Bigg],
\end{aligned} \tag{9}$$

[2]We should note that many equivalent formulas for these quantities are available. We have chosen the most compact of these formulas for our presentation, but it is worth emphasizing that this is not the most numerically stable representation (for example, in some degenerate situations the covariance computed via this formula may become non-positive semidefinite, due to numerical error). More stable formulas may be constructed by devising similar recursions for the Cholesky square root of the covariance, i.e., the upper triangular matrix W such that $C = W^T W$; this ensures that C remains positive definite. See, e.g. (Howard and Jebara, 2005) for details.

where the first equality is due to the fact that the posterior distribution $p(Q|Y)$ is Gaussian (since (Q, Y) are jointly Gaussian) and the mean and mode of a Gaussian distribution agree. It is worth noting that the last term is an unconstrained quadratic program in Q, and may therefore be solved by forming

$$\hat{Q} = \arg\max_Q \frac{1}{2} Q^T \mathbf{A} Q + \mathbf{b}^T Q = \mathbf{A}^{-1}\mathbf{b}, \tag{10}$$

where

$$\mathbf{A} = -\begin{pmatrix} C(q_1)^{-1} + A^T C_q^{-1} A & -A^T C_q^{-1} & \mathbf{0} & \dots & \\ -C_q^{-1} A & A^T C_q^{-1} A + C_q^{-1} & -A^T C_q^{-1} & \mathbf{0} & \dots \\ \mathbf{0} & -C_q^{-1} A & A^T C_q^{-1} A + C_q^{-1} & -A^T C_q^{-1} & \mathbf{0} \\ & \vdots & & \ddots & \\ & \dots & \mathbf{0} & -C_q^{-1} A & C_q^{-1} \end{pmatrix} - I_T \otimes B^T C_y^{-1} B,$$

where $\otimes$ denotes the Kronecker product and I_T the T-dimensional identity matrix, and

$$\mathbf{b} = \begin{pmatrix} B^T C_y^{-1} y_1 + C(q_1)^{-1} E(q_1) \\ B^T C_y^{-1} y_2 \\ \vdots \\ B^T C_y^{-1} y_T \end{pmatrix}.$$

This block-tridiagonal form of $\mathbf{A}$ implies that the linear equation $\mathbf{A}\hat{Q} = \nabla$ may be solved in $O(\dim(q)^3 T)$ time (e.g., by block-Gaussian elimination (Press et al., 1992); note that we never need to compute $\mathbf{A}^{-1}$ explicitly). Thus this matrix formulation of the Kalman smoother is equivalent both mathematically and in terms of computational complexity to the forward-backward method. In fact, the matrix formulation is often easier to implement; for example, if $\mathbf{A}$ is sparse and banded, the standard Matlab command $Q = \mathbf{A} \backslash \nabla$ calls the $O(T)$ algorithm automatically — Kalman smoothing in just one line of code.

We should also note that a second key application of the Kalman filter is to compute the posterior state covariance $Cov(q_t|Y)$ and also the nearest-neighbor second moments $E(q_t q_{t+1}^T|Y)$; the posterior covariance is required for computing confidence intervals around the smoothed estimates $E(q_t|Y)$, while the second moments $E(q_t q_{t+1}^T|Y)$ are necessary to compute the sufficient statistics in the expectation-maximization (EM) algorithm for estimating the Kalman model parameters, as we will see below. These quantities may easily be computed in $O(T)$ time in the matrix formulation. For example, since the matrix $\mathbf{A}$ represents the inverse prior covariance matrix of our Gaussian vector Q, $Cov(q_t|Y)$ is given by the (t, t)-th block of $\mathbf{A}^{-1}$, and it is well-known that the diagonal and off-diagonal blocks of the inverse of a block-tridiagonal matrix can be computed in $O(T)$ time; again, the full inverse $\mathbf{A}^{-1}$ (which requires $O(T^2)$ time in the block-tridiagonal case) is not required (Rybicki and Hummer, 1991; Rybicki and Press, 1995; Asif and Moura, 2005).

So from equation (10) it is clear that the Kalman smoother is just a linear filter applied to the observations Y, since $\mathbf{b}$ depends linearly on Y and the quadratic Hessian $\mathbf{A}$ is independent of Y. This linear filter is not strictly time-invariant (convolutional), due to boundary effects at the times $t = 0$ and $t = T$; this can clearly be seen from either the forward or the backward recursions developed above, since the covariances C_t^f and C_t^s depend on time. However, it turns out that both of these covariances display limiting behavior:

$$C_t^f \to C^f$$

as t becomes large and

$$C_t^s \to C^s$$

for t sufficiently less than T, for suitable limiting matrices C^f and C^s of Riccati form[3]. Now for times t sufficiently far from the boundary values (i.e., $0 \ll t \ll T$), the Kalman filter does behave like a linear time-invariant system, and therefore the filter is completely specified by the impulse response.

This impulse response may be computed easily. First, let's take another look at the forward filter. In the time-invariant case $C_t^f = C^f$, we have

$$\mu_t^f = C^f \left[(AC_f A^T + C_q)^{-1} A\mu_{t-1}^f + B^T C_y^{-1} y_t \right]$$

(we have dropped the offset b term for simplicity); this may be rewritten in the form

$$\mu_t^f = R_f \mu_{t-1}^f + S_f y_t \tag{11}$$

for suitable matrices R_f and S_f. Thus in this time-invariant setting the forward filter is just a simple infinite-impulse-response filter driven by the input $S_f y_t$, and we obtain the impulse response

$$\{S_f, R_f S_f, R_f^2 S_f, \ldots\}.$$

We may similarly write

$$\mu_t^s = R_s \mu_{t+1}^s + S_f \mu_t^f,$$

for suitable R_s, S_s; thus the full smoothing filter is obtained by running the output of the forward filter backwards through a second filter whose impulse response is

$$\{S_s, R_s S_s, R_s^2 S_s, \ldots\}.$$

For example, in the one-dimensional case, the impulse response corresponding to the full forward-backward smoother is a double exponential whose parameters may be computed explicitly.

Finally, it is also useful to think of the optimization problem (9) as a penalized maximization problem, in which we try to choose Q to maximize the likelihood of the data Y (the last term on the right-hand side of (9)) under a smoothness and initialization penalty (the first two terms on the right-hand side of (9)). As in any optimal filter, the properties of the filter depend on both the observation noise and the smoothness of the underlying signal: the larger the observation noise C_y, or the smaller the dynamics noise C_q (i.e., the smoother the signal), the larger the penalization and the smoother the resulting path Q.

Thus we have two algorithms — forward-backward and direct optimization — for computing $E(Q|Y)$. These methods are very nearly equivalent (in terms of computational accuracy, stability, and efficiency) in the linear-Gaussian setting, but we should emphasize that these

[3]These matrices may be obtained as solutions to the "Riccati equations"

$$C^f = \left[(AC^f A^T + C_q)^{-1} + B^T C_y^{-1} B \right]^{-1}$$

and

$$C^s = C^f + J \left[C^s - (AC^f A^T + C_q) \right] J^T,$$

with J a suitable limit of J_t, but in many cases it is easier to just run the forward and backward recursions until C_t^f and C_t^s converge to obtain the limiting matrices C^f and C^s.

algorithms typically do not generally agree if either the observation or evolution dynamics are non-Gaussian. As we will see, each approach has its advantages and disadvantages in this more general setting: the forward-backward recursion is best for online applications (where estimates must be updated in real time as new information is observed) and extends nicely to cases in which the log-conditional density $\log p(Q|Y)$ is strongly non-concave (see section **??** below), while the direct optimization approach is typically preferred when $\log p(Q|Y)$ is concave and in the offline setting, where we are interested in inferring Q based on all the available past and future information Y (see section **??**).

The M-step in the Kalman model splits into two regression problems

As in the discrete HMM case, we need only compute pairwise expectations:

$$\begin{aligned} E_{p(Q|Y,\hat{\theta}^{(i)})} \log p(Y,Q|\theta) &= \int p(q_1|Y,\hat{\theta}^{(i)}) \log p(q_1|\theta) dq_1 + \sum_{t=1}^{T} \int p(q_t|Y,\hat{\theta}^{(i)}) \log p(y_t|q_t,\theta) dq_t \\ &+ \sum_{t=2}^{T} \int\int p(q_t,q_{t+1}|Y,\hat{\theta}^{(i)}) \log p(q_{t+1}|q_t,\theta) dq_{t-1} dq_t; \end{aligned}$$

once again, the M-step breaks up into three independent pieces, one for each term above. The update for the initial distribution π is, as always, the simplest:

$$\hat{\pi}^{(i+1)} = G_{\mu_1^s, C_1^s}(q_1).$$

The update for the observation matrix B requires that we maximize

$$\sum_{t=1}^{T} E_{p(q_t|Y,\hat{\theta}^{(i)})} \left(-\frac{1}{2}(y_t - Bq_t)^T C_y^{-1} (y_t - Bq_t) \right);$$

taking the gradient with respect to B leads to an update of regression form,

$$\hat{B}^{(i+1)} = \left(\sum_{t=1}^{T} E_{p(q_t|Y,\hat{\theta}^{(i)})}(q_t q_t^T) \right)^{-1} \left(\sum_{t=1}^{T} E_{p(q_t|Y,\hat{\theta}^{(i)})}(q_t y_t^T) \right),$$

while the update for C_y is an average residual

$$\hat{C}_y^{(i+1)} = \frac{1}{T} \sum_{t=1}^{T} E_{p(q_t|Y,\hat{\theta}^{(i)})} \left((y_t - \hat{B}^{(i+1)} q_t)(y_t - \hat{B}^{(i+1)} q_t)^T \right).$$

Similarly, for the dynamics, we update

$$\hat{A}^{(i+1)} = \left(\sum_{t=1}^{T-1} E_{p(q_t,q_{t+1}|Y,\hat{\theta}^{(i)})}(q_t q_t^T) \right)^{-1} \left(\sum_{t=1}^{T-1} E_{p(q_t|Y,\hat{\theta}^{(i)})}(q_t q_{t+1}^T) \right),$$

and

$$\hat{C}_q^{(i+1)} = \frac{1}{T-1} \sum_{t=1}^{T-1} E_{p(q_t|Y,\hat{\theta}^{(i)})} \left((q_{t+1} - \hat{A}^{(i+1)} q_t)(q_{t+1} - \hat{A}^{(i+1)} q_t)^T \right).$$

Note that only the first and second moments $E(q_t q_t^T|Y), E(q_t q_{t+1}^T|Y), E(q_t|Y), E(q_t y_t^T|Y)$, and $y_t y_t^T$ (or more precisely, the sums of these expectations over time t) are required to compute these updates. See, e.g., (Shumway and Stoffer, 2006) for further discussion.

1.3 The Kalman model is only identifiable up to linear transformations of the state variable

Before we move on to some examples, it is important to point out one last important fact about the Kalman model: if we only observe Y, then not all of the parameters (A, B, C_y, C_q) are identifiable (uniquely constrained by the data). For example, a scaling of $C_q \to aC_q$ may be countered by an inverse scaling of $B \to (1/a)B$ without any changes in the properties of the observed data Y: thus, clearly, C_q may not be inferred uniquely given Y if B is unconstrained.

The classical restriction which restores identiability here is to constrain $C_q = I$ (Roweis and Ghahramani, 1999) and B to be a lower triangular matrix with nonnegative diagonal entries. To see that these restrictions entail no loss of generality, just note that any Kalman model (Q, Y) with $C_q \neq I$ and unrestricted B always has an equivalent model (Q', Y) with $C_{q'} = I$ and restricted B' (note the observed Y is unchanged), with Q' related to Q via a linear (whitening) transformation: to construct such a Q', let W^T be a Cholesky square root of C_q,

$$C_q = WW^T$$

(such a decomposition exists for any symmetric positive semidefinite matrix, i.e., any covariance matrix), let O be an arbitrary orthogonal (rotation) matrix, and set

$$q'_t = OW^{-1}q_t,$$

make the similarity transformation

$$A \to A' = OW^{-1}A(OW^{-1})^T,$$

and undo the effect of OW^{-1} on Y via the substitution

$$B \to B' = BWO^T.$$

Since O is arbitrary here, we may choose a rotation O (by a Gram-Schmidt orthogonalization procedure) such that B'^T is upper triagonal, with a nonnegative diagonal. Thus we have constructed the desired Q'.

While these restrictions on C_q and B are fairly standard, in some cases alternate parameterizations (e.g., fixing the observation matrix B and allowing C_q to vary freely instead) are more convenient or more physiologically interpretable; we will point out some examples below.

1.4 Examples: intermittent, noisy, filtered, nonlinearly transformed voltage observations

As a first example application of these Kalman filtering ideas, let's examine the simplest case of the AR-type models for voltage fluctuations discussed above:

$$V(t+1) = V(t) + dt\big(-gV(t) + b + kI(t)\big) + \epsilon_t$$

$$\epsilon_t \sim \mathcal{N}(0, \sigma_V^2 dt)$$

$$y_t \sim \mathcal{N}(V(t), \sigma_y^2),$$

where we consider the hidden variable q_t to be the true voltage $V(t)$. Clearly, the dynamics matrix A here may be taken to be $A = (1 - gdt)$. An example of the Kalman smoother applied

to this type of model data is shown in Fig. 2: we make noisy, intermittent observations of the voltage, then apply the forward filter and full forward-backward smoother to estimate the true underlying voltage $V(t)$. (Recall that the forward step in the case that no observation is made at time t reduces to the marginal forward step: $\mu_t^f = A\mu_{t-1}^f$ and $C_t^f = AC_{t-1}^f A^T + C_q$, since in this case the observation probabilities $p(y_t|q_t)$ may be taken to be constant in q_t.)

Note that we have added a new term to the basic Kalman model:

$$b(t) = b + kI(t).$$

It is necessary to modify the E- and M-steps slightly to include this new term, and to fit the parameters b and k. The basic trick for the E-step (Roweis and Ghahramani, 1999) is to augment the state space: we add an element to the state vector $q_t \to (q_t\ q_t')$, and fix this element $q_t' = 1$. We also augment A:

$$A \to \begin{pmatrix} A & b(t) \\ 0 & 1 \end{pmatrix},$$

and then proceed using exactly the same formulae for the forward and smoothed means and covariances as before. In this case, A and $q(t)$ are scalars, but clearly the same idea works when q_t and $b(t)$ are vectors and A is a matrix. (It is worth noting that a similar state-augmentation trick can be used to allow the observations y_t to take on a mean different from Bq_t, though we will not make use of this generalization here.)

The generalization of the M-step is also straightforward. The updates for σ_Y^2 and $p(V_1)$ remain the same (we have assumed $B = 1$ here in order to maintain the identifiability of the model, since we are assuming $C_q = \sigma_V^2 dt$ to be an unknown free parameter). We need only modify the updates for the dynamics terms governing $p(q_{t+1}|q_t)$; this involves maximizing the term

$$\sum_{t=2}^{T} E_{p(q_t,q_{t+1}|Y,\hat{\theta}^{(i)})} \left(-\frac{1}{2}(q_{t+1} - Aq_t - b(t))^T C_q^{-1}(q_{t+1} - Aq_t - b(t)) \right),$$

or in this case

$$\sum_{t=2}^{T} E_{p(V(t),V(t+1)|Y,\hat{\theta}^{(i)})} \left(-\frac{1}{2\sigma_V^2}[V(t+1) - V(t) - dt(-gV(t) + b + kI(t))]^2 \right).$$

This expression is jointly quadratic in the parameters (g, b, k); as usual, we need to enforce the linear constraint $g \geq 0$, and the problem may be solved via standard quadratic programming methods, once we have computed the pairwise sufficient statistics $E(V(t)V(t+1)|Y, \hat{\theta}^{(i)})$, etc., in the E-step. Similarly, we have the usual residual formula

$$(\hat{\sigma}_V^2)^{(i+1)} = \frac{1}{dt}\frac{1}{T-1}\sum_{t=2}^{T} E_{p(V(t),V(t+1)|Y,\hat{\theta}^{(i)})} \left(V(t+1) - V(t) - dt[-\hat{g}^{(i+1)}V(t) + \hat{b}^{(i+1)} + \hat{k}^{(i+1)}I(t)] \right)^2.$$

The multicompartmental case follows similarly. Here the dynamics matrix A includes both leak terms g and intercompartmental coupling terms a_{ij}: in the simplest case of a linear

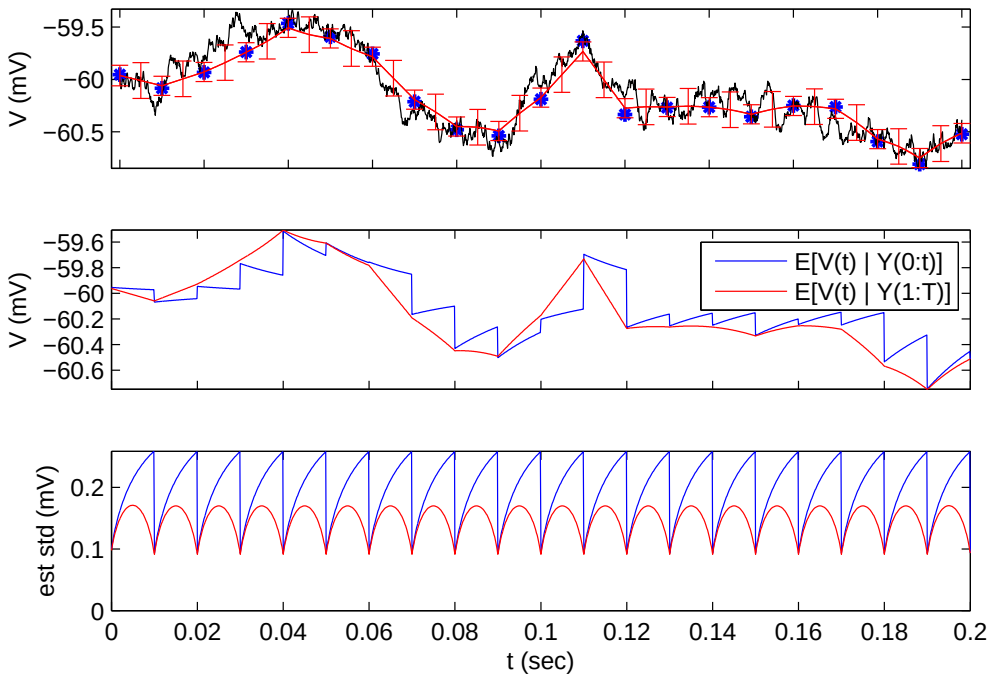

Figure 2: Illustration of Kalman smoother applied to simulated temporal voltage data $V(t)$. Top: Black trace: original data, generated via equation (2); input current $I(t)$ set to zero, for simplicity. Blue dots: observed data Y (intermittent and noisy samples of $V(t)$). Red trace: Kalman smoother estimate $E(V(t)|Y) \pm \sigma(V(t)|Y)$; note that smoother does a fairly good job of interpolating and noise filtering here. Middle: Comparison of forward Kalman filter $E(V(t)|Y_{1:t})$ (blue trace) and full Kalman smoother $E(V(t)|Y)$ (red); note that the forward estimate is discontinuous, jumping with each new data point, while the full smoother estimate is continuous. Bottom: Comparison of forward standard deviation $\sigma(V(t)|Y_{1:t})$ (blue) and full smoother standard deviation $\sigma(V(t)|Y)$ (red); note that the full smoother, which makes use of more data, has less uncertainty about the estimate than does the forward smoother. Note also that the forward variance is discontinuous in time, jumping downwards at the time of each new observation.

dendrite segment with N compartments, for example, A is given by the tridiagonal matrix

$$A = I + dt \begin{pmatrix} -(g_1 + a_{12}) & a_{12} & 0 & \dots & & 0 \\ a_{12} & -(g_2 + a_{12} + a_{23}) & a_{23} & 0 & & \vdots \\ 0 & a_{23} & -(g_3 + a_{23} + a_{34}) & a_{34} & \ddots & \\ \vdots & \ddots & & \ddots & \ddots & 0 \\ & & & & -(g_{N-1} + a_{N-1,N} + a_{N-2,N-1}) & a_{N-1,N} \\ 0 & \dots & & 0 & a_{N-1,N} & -(g_N + a_{N-1,N}) \end{pmatrix}.$$

In the simplest case of constant leak conductance $g_i = g$ and coupling parameters $a_{i,i+1} = a$, we have

$$A = (1 - gdt)I + adtD^2,$$

with D^2 denoting the second-difference operator (with so-called Neumann boundary condi-

tions — i.e., differences are only taken towards the interior of the segment)

$$D^2 = \begin{pmatrix} -1 & 1 & 0 & 0 & \dots & 0 \\ 1 & -2 & 1 & 0 & & \\ 0 & 1 & -2 & 1 & 0 & \\ & & & \ddots & & \\ & 0 & 1 & -2 & 1 & 0 \\ & & 0 & 1 & -2 & 1 \\ 0 & \dots & & 0 & 1 & -1 \end{pmatrix};$$

this second-difference form reduces in the limit of small compartment length to a second-derivative operator, and we obtain the familiar cable equation (Koch, 1999). Once again, the M-step reduces to a simple quadratic program. See Fig. 3 for an application to a model dendrite whose voltage is sampled via a raster scan[4].

If we divide the dendritic tree into N spatial compartments, then the standard forward-backward Kalman filter requires $O(N^3T)$ time, which may be prohibitively slow for finely-discretized neurons. However, in many cases, as illustrated in Fig. 3, the conditional covariance $Cov(V(x,t)|Y)$ of the voltage in compartment x at time t behaves in a stereotyped manner (remember, this variance is non-random, since the covariance in the linear Kalman setting does not depend on the observations). For example, in Fig. 3 the sampling schedule is periodic in time and we see that the covariance matrix $Cov(V(x,t)|Y)$ quickly settles into time-periodic behavior. Thus we may iterate the standard $O(N^3)$ Kalman recursion forward until $Cov(V(x,t)|Y)$ has sufficiently converged to its limiting behavior, and then simply use the recursion for μ_t^f in equation (11) to compute the remaining values of t. Since the matrices in (11) may be considered known for large enough times t, we can compute this recursion in $O(N^2)$ time, which leads to considerable savings. A similar method may be employed to efficiently compute the full forward-backward smoother.

We may further speed up the computations in the case that the voltage is observed in just one compartment at a time (e.g., in the case of a scanning two-photon laser experiment). Here the time-varying observation matrix B_t is of rank one; applying the Woodbury lemma, it is easy to see that updating the forward covariance now just requires $O(N^2)$ time and the forward mean recursion just $O(N)$ time (due to the sparseness of the dynamics and noise covariance matrices here). Similarly, it may be shown that the backwards smoother requires just $O(N^2)$ time for the covariance and $O(N)$ time for the mean, although to see this it is necessary to rewrite the backwards recursion slightly (e.g., we may use the "fast Kalman" formulation described in (Koopman, 1993; Durbin and Koopman, 2001)).

Above we assumed that the voltage in the full spatial extent of the dendritic tree is observed (albeit noisily and intermittently in time). Of course, this is typically not the case: for example, as in the example discussed in Fig. 3, it is common to use one-dimensional raster scans along the dendritic shaft to increase temporal resolution, but this comes at the cost of not being able to image along branched structures. (However, it is reasonable to assume that the shape of the dendritic tree is known, even if we are not able to image the voltage in the full tree, since post-hoc semi-automatic anatomical reconstruction of dendritic trees is now routine in many labs.) Thus we might frequently confront the problem of branches (and therefore unobserved intercompartmental currents) in our analysis. We can incorporate these

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

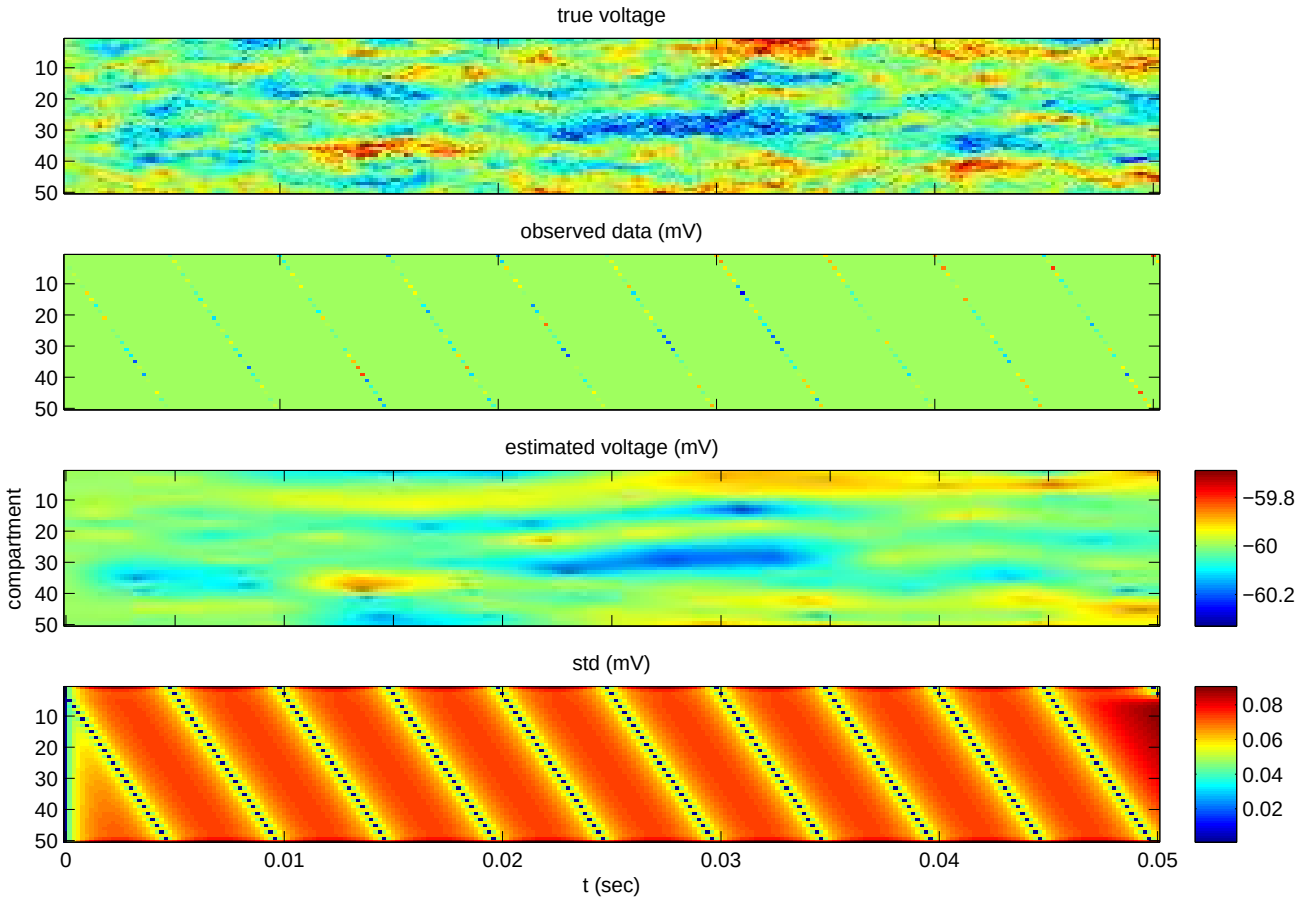

Figure 3: Illustration of Kalman smoother applied to simulated spatiotemporal voltage data $V_i(t)$. Top: Original data, generated via equation (3); input current $I(t)$ set to zero, for simplicity. Colorbar (mV) in lower right applies to all panels; red corresponds to high voltages, blue to low. Second panel: Observed data Y, sampled via raster scan over the compartments i (this is handled in our model simply by using a different observation matrix B at each time step); diagonal "stripes" in image indicate the trajectory of the raster scanner (i.e., compartments i are observed sequentially in time t). Third panel: Kalman smoother estimate $E[V_i(t)|Y]$. In both this and the previous figure, the true parameters are assumed known (i.e., we are illustrating the E-step, not the M-step). Bottom: posterior standard deviation $\sigma(V(t)|Y)$. As in Fig. 2, this uncertainty diminishes ahead of observations, due to the fact that the smoother takes future data into account.

missing components into our model in the obvious way: we simply append new variables to the hidden state vector q_t (where each new state element represents the voltage in one of the unobserved compartments) and then estimate the corresponding dynamics parameters (i.e., elements of the A matrix) which describe how the voltages in these unobserved compartments are coupled to the observed compartmental voltages.

Another important extension is to incorporate a dynamical model of the imaging process.

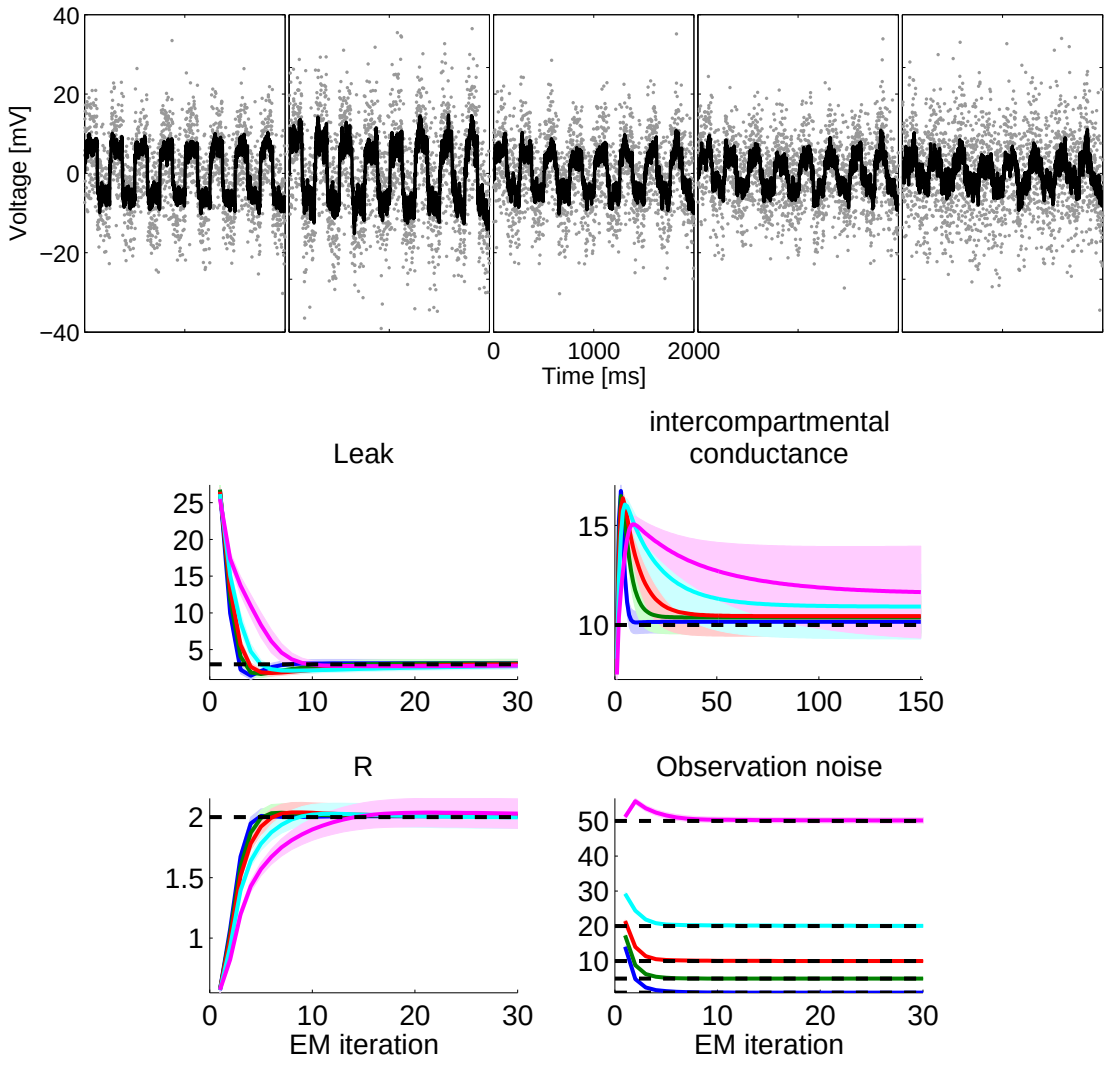

Figure 4: Inferring biophysical parameters from noisy measurements in a simple passive five-compartmental model cell, driven by periodic current input (Huys and Paninski, 2009). **Top**: True voltage (black) and noisy data (gray dots) from the five compartments of the cell with noise level $\sigma = 10$ mV. **Bottom**: Parameter inference with Kalman EM. Each panel shows the average inference time course $\pm$ one s.d. of one of the cellular parameters (A: Leak conductance; B: intercompartmental conductance; C: input resistivity; D: observation noise variance); gray dotted line shows true values. The colored lines show the inference for varying levels of noise, as indicated in panel D. Note that accurate estimation is possible even when the noise is five times as large as that shown in the top panels. Inference of the intercompartmental conductance suffers most from the added noise because the small intercompartmental currents have to be distinguished from the apparent currents arising from noise fluctuations in the observations from neighbouring compartments.

Particularly in genetically-encoded voltage indicators, the fluorescence signal will be some filtered version of $V(t)$ (this filering is an even bigger issue in the context of calcium-sensitive

dyes, where calcium buffering dynamics play a key role; we will address this topic in more depth below). It is often a decent approximation to assume that the filtering that takes $V(t)$ to the fluorescence signal is linear. If this linear filter can be captured with one or a few exponentials, it is straightforward to incorporate this feature into the Kalman model, simply by attaching additional dynamical variables at each imaged compartment, then coupling these variables to the voltage variable $\vec{V}(t)$ with linear weights (this corresponds to simply augmenting our dynamics matrix A) and setting our observation matrix B to sample these filtered-voltage variables instead of the unfiltered voltage V.

Finally, one of the major obvious drawbacks of the Kalman model is that the observations y_t are assumed to be linearly related to the underlying dynamical variables q_t. We will discuss relaxations of this assumption in much more depth below, but for now it is worth noting that we can easily incorporate observations of the form

$$y'_t = g(y_t),$$

$$y_t = Bq_t + \eta_t,\ \eta \sim \mathcal{N}(0, C_y)$$

for any invertible observtation nonlinearity $g(.)$ (since we may simply apply the inverse nonlinearity $g^{-1}(.)$ to y'_t to return to our original model); this is useful in the context of voltage-sensitive imaging, where the relationship between voltage and fluourescence may not be linear over the full observable dynamic range.

1.5 Example: spike sorting given nonstationary data using a mixture-of-Kalman-filters model

We have discussed the "spike sorting" problem from a few different points of view in previous chapters. Recall the mixture-of-Gaussian (MoG) model framework: the idea is that we observe voltage data V^i, and we model each voltage snippet as Gaussian,

$$V^i \sim \mathcal{N}_{\mu_{z_i}, C_{z_i}},$$

where z_i denotes the cluster identity of the i-th snippet, α_j denotes the probability of choosing the j-th cluster, and μ_j and C_j denote the j-th cluster's mean and covariance matrix, respectively.

Now consider the (quite common) situation that the mean voltage waveforms μ_j are nonstationary. In real data, this is most often due to drifts in the position of the recording electrode relative to the cell body, but nonstationarities in the spike shape can also be due to changes in the health of the cell, for example. (Note that we will not address the important case of temporary changes in spike shape due to partial inactivation of sodium channels during the relative refractory period (Lewicki, 1998); instead, we have longer-lasting nonstationarities in mind here.)

It turns out to be relatively straightforward to adapt the MoG model to handle these nonstationarities. The idea is that, for each cluster j, instead of letting μ_j be constant in time, we let it drift randomly[5]:

$$\mu_j^{i+1} = \mu_j^i + \epsilon_j^i, \quad \epsilon_j^i \sim \mathcal{N}(0, C_j^\mu). \tag{12}$$

[5]To be precise, it would be better to let the drift depend on experimental time instead of spike number; i.e., the variance of ϵ^i should depend linearly on the length of time between spikes i and $i-1$. This generalization is straightforward, so for clarity we stick to the simpler case here.

Meanwhile, we retain our Gaussian model for the observations V^i given z_i and μ_j^i:

$$V^i = \mu_{z_i}^i + \eta_{z_i}^i, \qquad \eta_{z_i}^i \sim \mathcal{N}(0, C_{z_i}^V).$$

Thus, given the sequence of cluster identities z_i, we just have a Kalman filter, as can be seen if we identify the vector of time-varying means $\vec{\mu}^i$ as our hidden state q_t and the voltage data V^i as our observations y_t. Of course, in practice, z_i are unobserved (these are exactly the variables we're trying to infer), and so we need to marginalize over these latent variables, and we are left with a mixture-of-Kalman-filters model, instead of our usual mixture-of-fixed-Gaussians model[6].

To perform inference in this model, we need to combine the familiar EM method from the MoG model with the Kalman filter. Let's begin by more explicitly casting this inference problem in terms of the EM framework. The parameters we want to infer are $\theta = \{(\mu_j^i, C_j^\mu, C_j^V, \alpha_j)_{0 \le j \le J, 1 \le i \le N}\}$, given the N observed voltage waveforms V^i, where i represents the experiment time index and J denotes the number of clusters. We have a Gaussian prior $p(\mu)$ on the vector of means μ_j^i, from equation (12); note that this prior is improper, since we have not constrained the initial value of μ_j. For now, assume that our prior on the remaining elements of θ is flat, though of course this may be generalized. Now we want to optimize the marginal log-posterior

$$\log p(\theta|V) = \log p(\theta) + \log p(V|\theta) = \log p(\mu) + \log \sum_z p(V, z|\theta).$$

First, we need to write out the complete log-posterior:

$$\begin{aligned}
\log p(V, z|\theta) + \log(\mu) &= \log p(V|z, \theta) + \log p(z|\theta) + \log(\mu) \\
&= \sum_{i=1}^N \log p(z_i|\theta) + \sum_{i=1}^N \log p(V^i|z_i, \theta) + \sum_{j=0}^J \sum_{i=2}^N \log p(\mu_j^i|\mu_j^{i-1}) \\
&= \sum_{i=1}^N \log \alpha_{z_i} + \sum_{i=1}^N \log \mathcal{N}_{\mu_{z_i}^i, C_{z_i}^V}(V_i) + \sum_j \sum_{i=2}^N \log \mathcal{N}_{\mu_j^{i-1}, C_j^\mu}(\mu_j^i) \\
&= \sum_{i=1}^N \left(\log \alpha_{z_i} - \frac{1}{2} \left(\log |C_{z_i}^V| + (V^i - \mu_{z_i}^i)^T \left(C_{z_i}^V\right)^{-1} (V^i - \mu_{z_i}^i) \right) \right) \\
&\quad -\frac{1}{2} \sum_j \sum_{i=2}^N \left(\mu_j^i - \mu_j^{i-1}\right)^T \left(C_j^\mu\right)^{-1} \left(\mu_j^i - \mu_j^{i-1}\right) + const.
\end{aligned}$$

[6]This model may be seen as a special case of the "switching Kalman filter" model (Wu et al., 2004), which models an observed time series whose dynamics and observation processes change randomly according to a Markov process; here we have $J + 1$ processes with fixed dynamics, and our observation is switching between these in an i.i.d. manner (which is a special case of a Markov process). It turns out that inference in the general switching Kalman model is significantly more difficult than the mixture-of-Kalmans model discussed here.

Now the expected log-posterior is

$$\begin{aligned} E_{p(z|V,\hat{\theta})} \log p(V,z|\theta) + \log(\mu) &= E_{p(z|V,\theta)} \left(\sum_{i=1}^{N} \log p(z_i|\theta) + \sum_{i=1}^{N} \log p(V^i|z_i,\theta) \right) + \sum_{j=0}^{J} \sum_{i=2}^{N} \log p(\mu_j^i|\mu_j^{i-1}) \\ &= \sum_{j=0}^{J} \sum_{i=1}^{N} p(z_i = j|V^i,\hat{\theta}) \left(\log \alpha_j - \frac{1}{2} \left(\log |C_j^V| + (V^i - \mu_j^i)^T \left(C_j^V\right)^{-1} (V^i - \mu_j^i) \right) \right) \\ &\quad -\frac{1}{2} \sum_{j=0}^{J} \sum_{i=2}^{N} \left(\mu_j^i - \mu_j^{i-1}\right)^T \left(C_j^\mu\right)^{-1} \left(\mu_j^i - \mu_j^{i-1}\right) + const. \end{aligned}$$

with

$$p(z_i = j|V^i,\hat{\theta}) = \frac{e^{\log \hat{\alpha}_j - \frac{1}{2}(\log|\hat{C}_j| + (V^i - \hat{\mu}_j^i)^T \left(\hat{C}_j^V\right)^{-1} (V^i - \hat{\mu}_j^i))}}{\sum_{j'=0}^{J} e^{\log \hat{\alpha}_{j'} - \frac{1}{2}(\log|\hat{C}_{j'}^V| + (V^i - \hat{\mu}_{j'}^i)^T \left(\hat{C}_j^V\right)^{-1} (V^i - \hat{\mu}_{j'}^i))}}. \tag{13}$$

Thus we see that the E-step here is exactly as in the MoG model: we compute the probability that the j-th cluster was responsible for the i-th observation.

Now for the M-step. As usual, the optimization over θ breaks up into a few independent terms: one involving α, $J+1$ involving C_j^V, and $J+1$ involving μ_j. It is not hard to see that the updates for α and C_j^V are exactly as in the MoG setting; thus, we focus on the μ_j updates here. If we collect terms involving μ_j, we see that we want to minimize

$$\sum_{j=0}^{J} \left(\sum_{i=1}^{N} p(z_i = j|V^i,\hat{\theta})(V^i - \mu_j^i)^T \left(C_j^V\right)^{-1} (V^i - \mu_j^i) + \sum_{i=2}^{N} \left(\mu_j^i - \mu_j^{i-1}\right)^T \left(C_j^\mu\right)^{-1} \left(\mu_j^i - \mu_j^{i-1}\right) \right).$$

Clearly, to optimize this sum over j we just need to optimize each individual summand

$$\sum_{i=1}^{N} p(z_i = j|V^i,\hat{\theta})(V^i - \mu_j^i)^T \left(C_j^V\right)^{-1} (V^i - \mu_j^i) + \sum_{i=2}^{N} \left(\mu_j^i - \mu_j^{i-1}\right)^T \left(C_j^\mu\right)^{-1} \left(\mu_j^i - \mu_j^{i-1}\right),$$

and here, finally, we recognize a block-tridiagonal quadratic function of the vector μ_j that strongly resembles the block-tridiagonal quadratic form optimized by the Kalman filter; the only difference is that here, as usual in the EM setting, the observations are weighted by the term $p(z_i = j|V^i,\hat{\theta})$.

We can solve this weighted Kalman problem directly using the matrix approach of equation (10), or by the forward-backward method. It turns out the forward-backward method is a bit more flexible here, because it allows us to update our estimates of the cluster means μ_j^i in an online manner, as follows. As each new spike is observed at time T, we compute the weight $p(z_T = j|V^T,\hat{\theta})$ (this entails a straightforward, very slight modification of equation (13)), run the filter backwards from time T (incorporating this new piece of information V^T) to time $T-s$, where s is the number of time steps required for the updated values of μ^i to effectively settle back down to the values obtained before the data at time T were observed[7], then update the weights $p(z_i = j|V^i,\hat{\theta})$ for $T-s \le i \le T$ and repeat these partial EM updates until convergence. In lengthy experiments this partial-sweep method can save a significant

[7]This lag s will in general depend on the model parameters, especially the dynamics noise C^μ: the noisier the dynamics of μ, the more quickly we can forget the recent history of the observations.

amount of computational time, since there is no need to reprocess all the past data every time a new spike is observed. In principle, these online sweeps could be performed in the context of real-time experiments.

To derive the forward step here, note that our initial conditions are "diffuse": we have a flat prior on the initial mean μ_0. Thus we can initialize by setting $E(\mu_j^1|V^1) = V^1$ and $Cov(\mu_j^1|V^1) = C_j^V$. Now we iterate forward by setting

$$Cov(\mu_j^{i+1}|V^{1:i+1}) = \left(\left(Cov(\mu_j^i|V^{1:i}) + C_j^\mu \right)^{-1} + p(z_i = j|V^i, \hat{\theta}) \left(C_j^V \right)^{-1} \right)^{-1}$$

and

$$E(\mu_j^{i+1}|V^{1:i+1}) = Cov(\mu_j^{i+1}|V^{1:i+1}) \left(\left(Cov(\mu_j^i|V^{1:i}) + C_j^\mu \right)^{-1} E(\mu_j^i|V^{1:i}) + p(z_i = j|V^i, \hat{\theta}) \left(C_j^V \right)^{-1} V^i \right);$$

these recursions can be derived by the usual complete-the-squares argument, just as in our computation of the terms in equation (6). The backwards recursion is exactly as in the standard Kalman setting, since the backwards step does not depend on the observed data except through the sufficient forward statistics $E(q_t|Y_{1:t})$ and $Cov(q_t|Y_{1:t})$. Finally, the update for the dynamics noise C_j^μ is standard once the pairwise conditional moments of μ_j have been computed.

2 An approximate point process filter may be constructed via our usual Gaussian approximations

Now that we have a good handle on the basic Kalman filter, in which both the hidden dynamics and observations are linear and Gaussian, we can start generalizing. We begin with the case that the observations $p(y_t|q_t)$ are not Gaussian, but instead are merely log-concave. (We have in mind the case that y_t are spike train observations from a GLM of the form

$$\lambda_t = f(I_t + Bq_t),$$

with $f(.)$, as usual, convex and log-concave; we will see several examples below.) This log-concavity allows us to exploit our standard Gaussian approximations, in this case approximating the forward probabilities $p(q_t, Y_{1:t})$ — which are now non-Gaussian, in general, due to the non-Gaussianity of the observations $p(y_t|q_t)$ — as Gaussian. This allows us to adapt many of the standard Kalman filter tools to this more general setting[8]. Similar ideas have been exploited quite fruitfully in the statistics literature (Fahrmeir and Tutz, 1994; West and Harrison, 1997).

As usual, we will be interested in computing quantities such as the smoothed mean $E(q_t|Y)$, covariance $C(q_t|Y)$, etc. As we emphasized in the preceding section, the coupled

[8]In the engineering literature the "extended Kalman filter" often refers to a linearization of nonlinear dynamics $p(q_t|q_{t-1})$, which leads to an approximation which is known to be somewhat unstable in general (Julier and Uhlmann, 1997); while we will address such nonlinear dynamics later, for now we restrict our attention to examples in which the hidden state q_t may reasonably be modeled with linear dynamics (and moreover to settings in which our usual log-concavity properties hold, making the expansions discussed in this section fairly robust).

forward-backward algorithm for these quantities requires that we compute the forwards probabilities $p(q_t, Y_{1:t})$ — which incorporate the observed data $Y_{1:t}$ directly — and then the backwards probabilities are computed using only the forwards probabilities and the dynamics $p(q_t|q_{t-1})$ (i.e., no further knowledge of the data Y is necessary once we have obtained the forwards probabilities). If we have a Gaussian approximation for these forward probabilities (with approximate forwards means and covariances $\mu_t^f \approx E(q_t|Y_{1:t})$ and $C_t^f \approx C(q_t|Y_{1:t})$), then the backwards step remains exactly the same as in the Kalman setting, with the same recursions of the (now approximate) smoothed means and covariances $\mu_t^s \approx E(q_t|Y_{1:T})$ and $C_t^s \approx C(q_t|Y_{1:T})$.

Thus we may focus on constructing our approximations for the forward probabilities. We emulate the derivation of the corresponding equation (6) in the Kalman setting:

$$\begin{aligned} p(q_t, Y_{1:t}) &= \left(\int p(q_{t-1}, Y_{1:t-1}) p(q_t|q_{t-1}) dq_{t-1} \right) p(y_t|q_t) \\ &\approx \left(\int w_{t-1}^f G_{\mu_{t-1}^f, C_{t-1}^f}(q_{t-1}) G_{Aq_{t-1}, C_q}(q_t) dq_{t-1} \right) p(y_t|q_t) \\ &= w_{t-1}^f G_{A\mu_{t-1}^f, AC_{t-1}^f A^T + C_q}(q_t) p(y_t|q_t) \\ &\approx w_t^f G_{\mu_t^f, C_t^f}(q_t), \end{aligned} \tag{14}$$

where the mean μ_t^f and covariance C_t^f of this Gaussian approximation are computed recursively, using one of the methods that we have discussed previously (Laplace approximation, Expectation Propagation, direct numerical integration, or Monte Carlo): for example, the Laplace approximation takes the form

$$\begin{aligned} \mu_t^f &= \arg\max_q \left[G_{A\mu_{t-1}^f, AC_{t-1}^f A^T + C_q}(q) p(y_t|q) \right] \\ &= \arg\max_q \left[-\frac{1}{2}(q - A\mu_{t-1}^f)^T (AC_{t-1}^f A^T + C_q)^{-1} (q - A\mu_{t-1}^f) + \log p(y_t|q) \right], \end{aligned} \tag{15}$$

and

$$\begin{aligned} C_t^f &= -\left(\frac{\partial^2}{\partial q^2} \left[-\frac{1}{2}(q - A\mu_{t-1}^f)^T (AC_{t-1}^f A^T + C_q)^{-1} (q - A\mu_{t-1}^f) + \log p(y_t|q) \right]_{q=\mu_t^f} \right)^{-1} \\ &= \left((AC_{t-1}^f A^T + C_q)^{-1} + J \right)^{-1}, \end{aligned}$$

where we have abbreviated the one-sample Fisher information matrix

$$J = -\frac{\partial^2}{\partial q^2} \log p(y_t|q)_{q=\mu_t^f}.$$

Note that, importantly, the covariance here does depend on the observations y_t (since J clearly depends on y_t in general; the Gaussian case, in which J is independent of the observed data, is somewhat exceptional in this sense); this is one major departure from the fully Gaussian setting.

To define the recursion for the weight w_t^f, note that we are approximating

$$w_{t-1}^f G_{A\mu_{t-1}^f, AC_{t-1}^f A^T + C_q}(q_t) p(y_t|q_t) \approx w_t^f G_{\mu_t^f, C_t^f}(q_t).$$

We may choose to make this approximation precise at a single point — $q = \mu_t^f$ is a reasonable choice, since this is the point we are expanding around in this second-order approximation — i.e.,

$$w_{t-1}^f G_{A\mu_{t-1}^f, AC_{t-1}^f A^T + C_q}(\mu_t^f) p(y_t|\mu_t^f) = w_t^f G_{\mu_t^f, C_t^f}(\mu_t^f),$$

which implies the update

$$\begin{aligned} w_t^f &= w_{t-1}^f \frac{G_{A\mu_{t-1}^f, AC_{t-1}^f A^T + C_q}(\mu_t^f) p(y_t|\mu_t^f)}{G_{\mu_t^f, C_t^f}(\mu_t^f)} \\ &= w_{t-1}^f \left(\frac{|C_t^f|}{|AC_{t-1}^f A^T + C_q|} \right)^{1/2} \exp\left(-\frac{1}{2} (A\mu_{t-1}^f - \mu_t^f)^T (AC_{t-1}^f A^T + C_q)^{-1} (A\mu_{t-1}^f - \mu_t^f) \right) p(y_t|\mu_t^f). \end{aligned}$$

Note that this Gaussian approximation requires that we compute a maximization on each time step t, which (while feasible due to the concavity of the objective function) may be time-consuming. In the limit of small uncertainty ($C_t^f \to 0$) or a weakly nonquadratic log-likelihood term $\log p(y_t|q_t)$, where the quadratic term due to the Gaussian dominates in equation (15), we may compute a simpler approximate update, corresponding to a single Newton step starting from μ_{t-1}^f: we obtain the covariance update

$$C_t^f = \left((AC_{t-1}^f A^T + C_q)^{-1} + J_0 \right)^{-1},$$

and the mean update

$$\begin{aligned} \mu_t^f &= \arg\max_q \Big[-\frac{1}{2} (q - A\mu_{t-1}^f)^T (AC_{t-1}^f A^T + C_q)^{-1} (q - A\mu_{t-1}^f) \\ &\quad + \log p(y_t|\mu_{t-1}^f) + g^T (q - \mu_{t-1}^f) - \frac{1}{2} (q - \mu_{t-1}^f)^T J_0 (q - \mu_{t-1}^f) \Big] \\ &= C_t^f \left((AC_{t-1}^f A^T + C_q)^{-1} (A\mu_{t-1}^f) + J_0 \mu_{t-1}^f + g \right), \end{aligned}$$

with the gradient

$$g = \nabla_q \log p(y_t|q)_{q=\mu_{t-1}^f}$$

and Hessian

$$J_0 = -\frac{\partial^2}{\partial q^2} \log p(y_t|q)_{q=\mu_{t-1}^f}$$

evaluated at μ_{t-1}^f now instead of μ_t^f.

2.1 Example: decoding hand position and neural prosthetic design via fixed-lag smoothing

One exciting recent application of statistical methods in neuroscience is in the design and implementation of neural prosthetic devices (Donoghue, 2002; Nicolelis et al., 2003; Truccolo et al., 2005; Wu et al., 2006; Santhanam et al., 2006; Velliste et al., 2008): the goal is to build a prosthetic device for use in paralyzed patients that can be implanted in the brain (e.g. in primary motor or parietal cortex) to decode these neural signals and provide a control signal

that the patient could use to drive a robot arm, or more generally aid in communication with the outside world.

From a statistical point of view, this is the decoding problem again — we are trying to decode some signal $\vec{x}(t)$ given spike train data D (though the signal $\vec{x}(t)$ here is not interpreted as a "stimulus" anymore, since the temporal causality of $\vec{x}$ and D are reversed) — but here processing speed and robustness are especially critical, since the decoding must be done in real time and with (in principle) no manual intervention. Hence recursive algorithms are preferred here.

One straightforward framework for recursive decoding in this context was employed by (Truccolo et al., 2005). The goal was to decode the two-dimensional position of the hand as a primate performed a simple visuomotor tracking task; the observed data here include the spike trains of multiple simultaneously recorded neurons from the contralateral primary motor cortex. More recently, similar (albeit simpler) techniques have been applied in human clinical studies (Hochberg et al., 2006), in which case the true hand position is constant (due to paralysis) but the intended hand position may be experimentally monitored and decoded.

We begin by defining the system dynamics: we let the hidden state q_t include the two-dimensional hand position and velocity, i.e.,

$$q_t = \begin{pmatrix} s_x(t) & v_x(t) & s_y(t) & v_y(t) \end{pmatrix}^T,$$

where $s(t)$ denotes the hand position at time t, $v(t)$ velocity, and x and y horizontal and vertical, respectively (it is of course possible to include more dynamical variables in q_t, e.g. acceleration, joint angle, etc., but this four-dimensional model makes a nice illustrative example). If the horizontal and vertical positions evolve independently "on average," then a reasonable dynamics matrix A is of block diagonal form

$$A = \begin{pmatrix} 1 & dt & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & dt \\ 0 & 0 & 0 & 1 \end{pmatrix},$$

with

$$C_q = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & \sigma_x^2 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \sigma_y^2 \end{pmatrix},$$

i.e., the position at time t is deterministic given the position and velocity at $t-1$, and the variance in the horizontal and vertical velocity is set by the possibly different constants σ_x^2 and σ_y^2. Thus in this case $s_x(t)$ and $s_y(t)$ are given by two independent AR(2) processes; we may fit these processes (or more elaborate AR processes, with interaction terms or higher-order lags) directly to the hand kinematic data, which is assumed to be fully observed in this case.

Now we need to define the emissions probabilities $p(y_t|q_t)$. Perhaps the simplest effective approach is to model the observations y_t as linear-Gaussian:

$$y_t \sim \mathcal{N}(Bq_t, C_y).$$

To decode Q now, we simply employ our standard linear Kalman filtering tools, as discussed in section 1.2 above. This approach was successfully pursued by (Wu et al., 2006); in particular,

this Kalman approach turns out to lead to more accurate decoding in practice than does the direct approach of multiple linear regression of D onto $\vec{x}(t)$ (Humphrey et al., 1970; Nicolelis et al., 2003; Hochberg et al., 2006). This may be somewhat surprising, since of course the Kalman filter may itself be interpreted as a linear filter applied to the observed data; thus it would appear that the optimal linear estimate constructed in the regression setting should outperform the Kalman filter (since the regression solution is by construction "optimal"). However, it is important to remember that the regression solution is optimal on the *training* data, not the test data; since the Kalman method fits many fewer parameters to the data than do typical direct linear regression approaches, the Kalman solution is much less prone to overfitting, and therefore its generalization (test) error is often superior.

An alternate approach is to use a GLM to define the emissions $p(y_t|q_t)$ (Paninski et al., 2004a; Truccolo et al., 2005):

$$\lambda_i(t) = f(b_i + B_i^T q_t + \sum_j h_{ij} n_{t-j}),$$

where b_i is a constant offset term, the i-th row B_i of the observation matrix B encapsulates the i-th neuron's preferences for the hand velocity and position (for example, if the i-th neuron fires more rapidly when the hand velocity is rightward, then the second element of B_i should be positive), and as usual h_{ij} captures the i-th neuron's spike history effects (we assume no interneuronal interaction terms here to keep the notation manageable, but it is straightforward to add these terms (Truccolo et al., 2005)). The GLM is, as we have argued previously, perhaps a better model for neural responses than the linear-Gaussian model, and involves very little additional computational expense: the parameters (b_i, B, h_{ij}) may be fit via standard concave maximum likelihood; i.e., no EM is required here since we assume that q_t and y_t are fully observable during the parameter-fitting stage.

Now we have all the necessary components in place to define our decoding algorithm. Imagine we have some maximal acceptable response lag, τ: i.e., in the neural prosthetic context, we may need to provide the robot arm with a command signal (an estimate of where the hand should be, for example) no more than τ milliseconds after the current time. (Delays longer than a few hundred milliseconds might cause instabilities and oscillations due to overcompensatory behavior.) This means that we can potentially take advantage of up to τ extra milliseconds of data to better estimate the hand position and velocity at time t; that is, we can compute the "fixed-lag smoother" estimate $E(q_t|Y_{1:t+\tau})$ instead of the forward estimate $E(q_t|Y_{1:t})$ (clearly the fixed-interval smoother $E(q_t|Y_{1:T})$ is inappropriate in this online decoding context). Computing this fixed-lag smoother is now easy: we simply use the forward algorithm to compute the approximate moments μ_s^f and C_s^f for all times s up to $s = t+\tau$ (this step can obviously be done recursively), then propagate the backwards smoother τ steps back to obtain our approximation for $E(q_t|Y_{1:t+\tau})$. The conceptual simplicity of this algorithm illustrates the power of the general state-space framework we have developed so far; see Fig. 5 for an example of this recursive decoding method applied to primate cortical data.

2.2 Example: decoding rat position given hippocampal place field activity

Uri/Emery will fill this in... one mathematical point: the loglikelihoods can be very non-concave here, which means that local optima are possible, though this does not seem to be an issue in practice, if sufficient information about the initial location q_0 is provided.

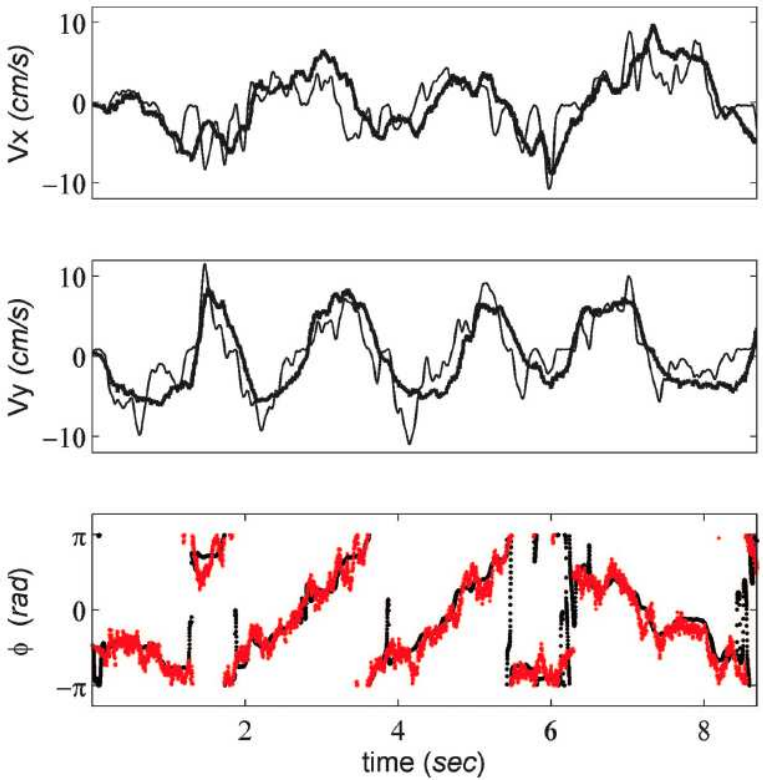

Figure 5: An example of neural decoding of (x, y) hand velocities and movement direction by the point process filter $E(q_t|Y_{1:t})$ computed via the Gaussian approximation technique (figure adapted from Fig. 10 of (Truccolo et al., 2005).). Estimated velocities (thick curve) and direction (red dots), together with true velocities and direction, are shown for a single decoded test trial. Time was discretized at a 1-ms resolution. 20 simultaneously-recorded neurons from the primary motor cortex were used in the decoding.

2.3 Example: decoding position under endpoint constraints

Uri/Emery will fill this in, based on (Srinivasan et al., 2006); describe recursive backwards propagation of information in the Kalman setting.

basic idea: we are given some endpoint information y_T before the experiment begins, and we want to incorporate this information in our online decoder. This can be accomplished quite efficiently via our backwards recursion.

We want to compute

$$\begin{aligned}
p(q_t|q_0, y_T, Y_{0:t}) &= \frac{1}{Z} p(q_t, y_T, Y_{0:t}|q_0) \\
&= \frac{1}{Z} \int \dots \int p(q_t, q_{t+1}, q_{t+2}, \dots, q_T, y_T, Y_{0:t}|q_0) dq_{t+1} dq_{t+2} \dots dq_T \\
&= \frac{1}{Z} \int \dots \int p(q_t, Y_{0:t}|q_0) p(q_{t+1}|q_t) p(q_{t+2}|q_{t+1}) \dots p(q_T|q_{T-1}) p(y_T|q_T) dq_{t+1} dq_{t+2} \dots dq_T \\
&= \frac{1}{Z} p(q_t, Y_{0:t}|q_0) \int p(q_{t+1}|q_t) dq_{t+1} \int p(q_{t+2}|q_{t+1}) dq_{t+2} \dots \int p(q_T|q_{T-1}) p(y_T|q_T) dq_T
\end{aligned}$$

This may be done efficiently by recursing backwards from T; each of the above integrals may be computed exactly using standard Gaussian formulas.

2.4 Example: using the point process filter to approximately calculate mutual information

We have previously discussed the problem of estimating the mutual information between a stimulus $\vec{x}$ and spike train data D. The decoding-based approach we discussed before required that we perform an optimization and determinant computation over a $\dim(\vec{x})$-dimensional space; these computations are tractable for modest $\dim(\vec{x})$, but can quickly become more challenging as $\dim(\vec{x})$ becomes very large.

It is thus natural to ask if we can use state-space methods to perform this computation in a recursive manner. Define the information rate (Cover and Thomas, 1991) as the large-T limit

$$\lim_{T\to\infty} \frac{1}{T} I(Q_{1:T}; Y_{1:T}) = \lim_{T\to\infty} \frac{1}{T} \left(H(Q_{1:T}) - H(Q_{1:T}|Y_{1:T}) \right);$$

for an autoregressive model, we have

$$\begin{aligned}
\lim_{T\to\infty} \frac{1}{T} H(Q_{1:T}) &= \lim_{T\to\infty} \frac{1}{T} H\left(p(q_1) \prod_{t=2}^{T} p(q_t|q_{t-1}) \right) \\
&= \lim_{T\to\infty} \frac{1}{T} \left(H(q_1) + \sum_{t=2}^{T} H(q_i|q_{t-1}) \right) \\
&= H(q_i|q_{t-1}) = \frac{1}{2} \log |C_q| + const.,
\end{aligned}$$

since $(q_i|q_{t-1})$ is Gaussian with covariance C_q, independent of q_{t-1}.

We may deal with the conditional entropy $H(Q|Y)$ if we recall that a hidden Markov model $(Q|Y)$ conditioned on the observables Y is still a Markov chain (albeit with time-inhomogeneous parameters). Now, starting with the standard Kalman setting for simplicity, we have

$$\begin{aligned}
\lim_{T\to\infty} \frac{1}{T} H(Q_{1:T}|Y_{1:T}) &= \lim_{T\to\infty} \frac{1}{T} E_{Y_{1:T}} H\left(p(q_1|Y_{1:T}) \prod_{t=2}^{T} p(q_t|q_{t-1}, Y_{1:T}) \right) \\
&= \lim_{T\to\infty} \frac{1}{T} \left(E_{Y_{1:T}} H(q_1|Y_{1:T}) + \sum_{t=2}^{T} E_{Y_{1:T}} H(q_t|q_{t-1}, Y_{1:T}) \right) \\
&= \lim_{T\to\infty} \frac{1}{T} \sum_{t=2}^{T} E_{Y_{1:T}} H(q_t|q_{t-1}, Y_{1:T}) \\
&= \lim_{T\to\infty} \frac{1}{T} \sum_{t=2}^{T} \frac{1}{2} \log |C(q_t|q_{t-1}, Y_{1:T})| + const. \\
&= \frac{1}{2} \lim_{T\to\infty} \frac{1}{T} \sum_{t=2}^{T} \log |C_t^s - J_{t-1} C_t^s J_{t-1}^T| + const.,
\end{aligned}$$

where we have used the fact that the covariance in the standard Kalman model is independent of the observed Y, and in the last line we use the forward-backward pairwise covariance (8)

and the standard formula for computing the conditional covariance of a Gaussian: if (x, y) are jointly Gaussian with covariance

$$\begin{pmatrix} C_x & C_{xy} \\ C_{xy}^T & C_y, \end{pmatrix}$$

then $y|x$ is Gaussian with mean

$$\mu_{y|x} = \mu_y + C_{xy}^T C_x^{-1}(x - \mu_x)$$

and covariance

$$C_{y|x} = C_y - C_{xy}^T C_x^{-1} C_{xy}$$

(note that we don't need the mean here, just the covariance).

Now, finally, in the standard Kalman case we have

$$\lim_{T\to\infty} \frac{1}{T} \sum_{t=2}^{T} \log |C_t^s - J_{t-1} C_t^s J_{t-1}^T| = \log |C_*^s - J_* C_*^s J_*^T| + const.,$$

where we have abbreviated

$$C_*^s = \lim_{T\to\infty} C_{T/2}^s$$

and

$$J_*^s = \lim_{T\to\infty} J_{T/2},$$

the values of these matrices in the limit as infinitely many measurements Y are made in the past and future (note that generally $C_{T/2}^s < C_T^s$, since $C_{T/2}^s$ incorporates more measurements than C_T^s: $C_{T/2}^s$ incorporates both the past and future measurements, while C_T^s incorporates only the past); thus the information rate may be evaluated quite explicitly in this case,

$$\lim_{T\to\infty} \frac{1}{T} I(Q_{1:T}; Y_{1:T}) = \frac{1}{2} \log \frac{|C_q|}{|C_*^s - J_* C_*^s J_*^T|}.$$

If we mimic this derivation in the case of non-Gaussian observations, substituting approximate covariances for exact covariances where necessary, we find

$$\lim_{T\to\infty} \frac{1}{T} I(Q_{1:T}; Y_{1:T}) = \frac{1}{2} \left(\log |C_q| - E_Y \log |C_{t+1}^s - J_t C_{t+1}^s J_t^T| \right),$$

where the expectation over Y may be approximated numerically by

$$E_Y \log |C_{t+1}^s - J_t C_{t+1}^s J_t^T| \approx \frac{1}{T} \sum_{t=2}^{T} \log |C_t^s - J_{t-1} C_t^s J_{t-1}^T|;$$

this gives us a highly tractable method for approximately computing the information rate in the state-space setting. See (Barbieri et al., 2004) for an application of a somewhat simpler version of this idea to rat hippocampal data.

2.5 Example: modeling learning in behavioral experiments; combining observations over multiple experiments

Uri/Emery will fill this in...

Our next example involves behavioral data. A basic question in systems neuroscience is: how does neural activity change as an animal learns? For example, consider an experiment in which the animal forms a novel association between one of S stimuli and a reward. It would be informative to plot a neuron's firing rate as a function of the animal's certainty about this association: for example, it is known that neurons in some brain areas fire more rapidly when the animal is making a new association (Wirth et al., 2003), while other neurons might fire more vigorously once the association is completely consolidated; studying these neurons should provide us with insight into how memories are formed and consolidated in the intact brain.

Unfortunately, the animal's certainty about the association is a hidden variable: we cannot observe this quantity directly in the laboratory. Thus we need to infer it from data: this may be done within the state-space framework.

A reasonable model of this experiment is as follows: we define q_t as the animal's certainty about the association. At the beginning of the experiment, the animal knows nothing about the association (e.g., which stimulus is associated with the reward), and so we may assume that the certainty variable starts deterministically at some baseline "uncertain" value q_0. Then with each trial the animal might learn a little bit about which stimulus is associated with the reward; some trials are more informative than others, so the change in the certainty value might be stochastic,

$$q(t+1) = q(t) + b + \epsilon_t;$$

the scalar b controls the speed of learning, while ϵ_t reflects the variability in the "informativeness" of each trial.

Finally, we need to define the observation $p(y_t|q_t)$, where $y_t = 1$ if the animal responded correctly on trial t, and 0 otherwise. A simple model is

$$p(y_t = 1|q_t) = g(Bq_t + m),$$

where $0 < g(.) < 1$ is an increasing function and B and m are parameters we would like to infer from the data.

To fit the model parameters (B, m, b, C_q) we would like to combine data over many such experiments (where each experiment consists of many trials over which the anmial learns a novel association). This is a standard generalization of our usual EM approach: instead of maximizing a weighted expectation of the complete loglikelihood (where the weighted expectation is taken over time bins t within a single experiment), we maximize a summed weighted expectation, where the sum is taken over all experiments. A slightly simpler case of this model (in which the mean drift rate b was assumed to be zero) has been considered in (Smith et al., 2004); see (Smith and Brown, 2003; Smith et al., 2004) for full details, and (Smith et al., 2005) for some further extensions to this methodology.

It should be noted that our model for the observations $p(y_t|q_t)$ is somewhat weak here: we have modeled the responses as a Bernoulli process given the learning state q_t, but this is somewhat unrealistic, since trials may vary systematically in difficulty (i.e., $p(y_t|q_t)$ could also depend on further trial information). Also, because our model fitting procedure only makes use of the binary variable $y_t \in \{0, 1\}$ (correct vs. incorrect trials), we throw out some

potentially useful information: some mistakes are worse than others, and therefore the type of mistake can tell us something about q_t, i.e., how well the subject is actually learning the task.

2.6 Example: tracking nonstationary parameters in spike train models; adaptive online estimation of model parameters

In many (perhaps all) neurophysiological settings, we must deal with issues of nonstationarity: biological preparations can never be considered completely static due to slow degradations and changes in the experimental setup, and in the neurophysiology context we must contend with slow ongoing modulations in behavioral or attentive state, as well as plasticity or learning-induced changes which might be the primary focus of the experiment. In general we may attempt to manually select epochs of the experiment in which the responses "look stable," but of course it would be more elegant and efficient if we could instead make more complete use of the data, including any "nonstationary" epochs — which in many important cases account for the majority of the data set.

We can accomplish this goal by making use of this extended Kalman technology. The idea is to let the hidden state q_t be the unknown parameter θ_t at time t. The only item we need to change in our framework is the emissions probability:

$$p(y_t|q_t) = p(y_t|x_t, \theta_t).$$

If $p(y_t|x_t, \theta_t)$ is a log-concave function of θ_t (as, of course, in the GLM setting), then we may employ the extended Kalman technique without further modification: we simply write the state-space model

$$\begin{aligned} \theta_{t+1} &= A\theta_t + \epsilon_t \\ y_t &\sim p(y_t|x_t, \theta_t). \end{aligned}$$

In this setting the dynamics matrix A is often set to the identity matrix (i.e., there is no "mean drift" in the parameters, only stochastic fluctuations), but of course we may infer A directly via EM is sufficient data are available.

It is worth considering the limiting case that $Cov(\epsilon_t) = 0$ (or the more general mixed case, where some but not all elements of this covariance matrix are zero). If ϵ_t is known to be zero at each time t, then the extended Kalman technique reduces to an online updating scheme for computing the MAP estimates $\arg\max_\theta p(\theta|X, Y)$, in which each time a new data point (x_t, y_t) is observed, we update our estimate $\mu_t^f \approx E(\theta|Y_{1:t}, X_{1:t})$ and our posterior covariance matrix $C_t^f \approx Cov(\theta|Y_{1:t}, X_{1:t})$, and since no additional uncertainty is introduced on each time step (since $\epsilon = 0$), our posterior uncertainty C_t^f converges to zero. It is possible to show, under some technical conditions, that these online updates are asymptotically efficient: the online estimate μ_t^f converges to the underlying "correct" value of θ just as quickly (as a function of t) as the full nonrecursive estimate $\arg\max_\theta p(\theta|X_{1:t}, Y_{1:t})$, which is more computationally expensive but does not make any Gaussian approximations. (More explicitly, both the recursive and nonrecursive estimates converge to θ at a $\sqrt{N}$ rate, with the asymptotic covariance given by the inverse of the expected Fisher information evaluated at the true θ; i.e., no information is lost asymptotically by computing the estimate recursively.) See (Sharia, 2008) for details.

The simplest example in which this nonstationary model is useful is as follows. Imagine we are recording from a neuron whose excitability is changing slowly with time (perhaps due

to degradation of the preparation, or fluctuations in anesthesia levels). If we are fitting a GLM to the cell's responses,

$$\lambda(t) = f(\vec{k}^T \vec{x}_t),$$

say, we might simply add a time-varying hidden variable q_t,

$$\lambda(t) = f(q_t + \vec{k}^T \vec{x}_t),$$

and then use the extended Kalman technology to infer the timecourse of q_t directly from the data, either so that we may subtract q_t to obtain better estimates of the parameter $\vec{k}$, or so that we might study the dependence of q_t on other observable variables. (In this case it makes sense to fix the emissions matrix $B = 1$ and infer C_q from data, just as in the voltage smoothing setting.)

We will discuss further examples below. (In fact, we will see that if online estimates are not required, then alternate methods for inferring q_t and $\vec{k}$, and more generally θ_t, are preferred.) See also (Lewi et al., 2009) for a discussion of how to incorporate this nonstationarity tracking idea in the context of optimal online stimulus design.

2.7 Second-order Laplace approximations lead to more accurate filters in the high-information regime

Kass will fill in something here...

3 The maximum a posteriori path and Laplace approximation may often be computed directly via efficient tridiagonal-matrix methods

We saw in the last section that forward-backward methods are particularly useful in the context of online decoding analyses (Brown et al., 1998; Brockwell et al., 2004; Truccolo et al., 2005; Shoham et al., 2005; Wu et al., 2006; Srinivasan et al., 2006; Kulkarni and Paninski, 2007b) and in the analysis of plasticity and nonstationary tuning properties (Brown et al., 2001; Frank et al., 2002; Eden et al., 2004; Smith et al., 2004; Czanner et al., 2008; Lewi et al., 2009; Rahnama Rad and Paninski, 2009), where we need to track a dynamic "moving target" in real time given noisy, indirect spike train observations.

However, in offline applications the forward-backward approach is not always preferred. For example, in many cases the dynamics $p(q_t|q_{t-1})$ or observation density $p(y_t|q_t)$ may be non-smooth (e.g., the state variable q may be constrained to be nonnegative, leading to a discontinuity in $\log p(q_t|q_{t-1})$ at $q_t = 0$). In these cases the forward distribution $p(q_t|Y_{1:t})$ may be highly non-Gaussian, and the basic forward-backward Gaussian approximation methods described above break down.

More precisely, the forward-backward point-process smoother discussed above approximates the MAP path for Q given Y using iterative "local" approximations at each time point t. These approximations are valid in two limiting situations (Ahmadian et al., 2009; Koyama and Paninski, 2009): the "low-information" limit in the case of linear-Gaussian prior dynamics, where the signal-to-noise of the spiking response is poor and the non-Gaussian observation terms $p(y_t|q_t)$ vary weakly as a function of q_t, making the Gaussian prior $p(q_t)$ dominant; and

the "high-information" limit, where the posterior $p(q_t|Y)$ becomes very sharply peaked around the mode and a local Laplace approximation is valid.

A more general and direct approach for computing the exact MAP path (instead of approximating this path via the forward-backward method) is well-known in the state space literature (Fahrmeir and Kaufmann, 1991; Fahrmeir and Tutz, 1994; Bell, 1994; Davis and Rodriguez-Yam, 2005; Jungbacker and Koopman, 2007; Koyama and Paninski, 2009; Paninski et al., 2009). The method is a direct generalization of the tridiagonal matrix methods for the linear Kalman model discussed in section 1.2. The key is to re-examine the derivation of equation (9):

$$\begin{aligned}\arg\max_Q p(Q|Y) &= \arg\max_Q \left(\log p(q_1) + \sum_{t=2}^{T} \log p(q_t|q_{t-1}) + \sum_{t=1}^{T} \log p(y_t|q_t) \right) \\ &= \arg\max_Q \left(\frac{1}{2} Q^T H Q + \nabla^T Q \right) \\ &= -H^{-1}\nabla,\end{aligned}$$

where we have abbreviated the Hessian and gradient of $\log p(Q|Y)$:

$$\nabla = \nabla_Q \log p(Q|Y)\big|_{Q=0}$$

$$H = \nabla\nabla_Q \log p(Q|Y)\big|_{Q=0}.$$

Recall the key point: the Hessian matrix H is block-tridiagonal, since $\log p(Q|Y)$ is a sum of simple one-point potentials ($\log p(q_t)$ and $\log p(y_t|q_t)$) and nearest-neighbor two-point potentials ($\log p(q_t, q_{t-1})$). More explicitly, we may write

$$H = \begin{pmatrix} D_1 & R_{1,2}^T & 0 & \cdots & & 0 \\ R_{1,2} & D_2 & R_{2,3}^T & 0 & & \vdots \\ 0 & R_{2,3} & D_3 & R_{3,4} & \ddots & \\ \vdots & \ddots & & \ddots & \ddots & 0 \\ & & & & D_{N-1} & R_{N-1,N}^T \\ 0 & \cdots & & 0 & R_{N-1,N} & D_N \end{pmatrix} \tag{16}$$

where

$$D_i = \frac{\partial^2}{\partial q_i^2} \log p(y_i|q_i) + \frac{\partial^2}{\partial q_i^2} \log p(q_i|q_{i-1}) + \frac{\partial^2}{\partial q_i^2} \log p(q_{i+1}|q_i), \tag{17}$$

and

$$R_{i,i+1} = \frac{\partial^2}{\partial q_i \partial q_{i+1}} \log p(q_{i+1}|q_i) \tag{18}$$

for $1 < i < N$.

From here it is straightforward to extend this approach to directly compute $\hat{Q}_{MAP}$ in non-Gaussian models of interest in neuroscience. In this section we will focus on the case that $\log p(q_{t+1}|q_t)$ is a concave function of Q; in addition, we will assume that the initial density $\log p(q_0)$ is concave and also that the observation density $\log p(y_t|q_t)$ is concave in q_t. Then it is easy to see that the log-posterior

$$\log p(Q|Y) = \log p(q_0) + \sum_t \log p(y_t|q_t) + \sum_t \log p(q_{t+1}|q_t) + const.$$

is log-concave in Q, and therefore computing the MAP path $\hat{Q}$ is a concave problem. Further, if $\log p(q_0)$, $\log p(y_t|q_t)$, and $\log p(q_{t+1}|q_t)$ are all smooth functions of Q, then we may apply standard approaches such as Newton's algorithm to solve this concave optimization. (We will discuss extensions to the non-smooth case in section 3.2 below.)

To apply Newton's method here, we simply iteratively solve the linear equation[9]

$$\hat{Q}^{(i+1)} = \hat{Q}^{(i)} - H^{-1}\nabla,$$

where we have again abbreviated the Hessian and gradient of the objective function $\log p(Q|Y)$:

$$\nabla = \nabla_Q \log p(Q|Y)\big|_{Q=\hat{Q}^{(i)}}$$

$$H = \nabla\nabla_Q \log p(Q|Y)\big|_{Q=\hat{Q}^{(i)}}.$$

Clearly, the only difference between the general non-Gaussian case here and the special Kalman case described above is that the Hessian H and gradient ∇ must be recomputed at each iteration $\hat{Q}^{(i)}$; in the Kalman case, again, $\log p(Q|Y)$ is a quadratic function, and therefore the Hessian H is constant, and one iteration of Newton's method suffices to compute the optimizer $\hat{Q}$.

In practice, this Newton algorithm converges within a few iterations for all of the applications discussed in this section. Thus we may compute the MAP path *exactly* using this direct method, in time comparable to that required to obtain the approximate MAP path computed by the recursive approximate smoothing algorithm discussed in the previous section. This close connection between the Kalman filter and the Newton-based computation of the MAP path in more general state-space models is well-known in the statistics and applied math literature; see (Fahrmeir and Kaufmann, 1991; Fahrmeir and Tutz, 1994; Bell, 1994; Davis and Rodriguez-Yam, 2005; Jungbacker and Koopman, 2007; Koyama and Paninski, 2009) for further discussion.

3.1 Example: inferring voltage given calcium observations on a dendritic tree

As emphasized above, we may apply these direct optimization methods to any of the examples discussed in the previous sections. Let's examine a new application here. In section 1.4 we discussed the problem of inferring the spatiotemporal dendritic voltage from noisy, intermittent voltage observations. Voltage-sensing techniques are currently hampered by low SNR (although optimal filtering methods can at least partially alleviate this problem, as we have demonstrated in Fig. 3 above). On the other hand, spatiotemporal calcium-sensing fluorescence signals may be currently recorded at relatively high SNR (Gobel and Helmchen, 2007; Larkum et al., 2008), but calcium signals, by their nature, provide only highly-thresholded information about the underlying voltage, since calcium channels are inactive at hyperpolarized voltages. The obvious question is: can we exploit these high-SNR superthreshold calcium observations to reconstruct (at least partially) the subthreshold voltage signal? More generally, we could combine both calcium and voltage measurements (where voltage measurements may be available via imaging techniques, or whole-cell patch recordings at the soma

[9] In practice, the simple Newton iteration does not always increase the objective $\log p(Q|Y)$; the standard remedy for this instability is to perform a simple backtracking linesearch along the Newton direction $\hat{Q}^{(i)} - \delta^{(i)} H^{-1}\nabla$ to determine a suitable stepsize $\delta^{(i)} \leq 1$.

or apical dendrite, or even through dense multielectrode recordings (Petrusca et al., 2007)) in an optimal way to obtain good estimates of the subthreshold voltage, but for now we stick with the case that only calcium observations are available.

As usual, we begin by writing down a model for the dynamics $p(q_t|q_{t-1})$ and observations $p(y_t|q_t)$. For clarity, we stick with the simple linear dynamics for the spatiotemporal voltage discussed in Fig. 3. Now in general we need to write down a model for voltage-dependent calcium influx, as well as for calcium extraction/buffering and for spatial diffusion of calcium within the dendrite. While we may reasonably approximate the latter terms with linear calcium dynamics, the voltage-dependent influx term will necessarily be nonlinear.

We use a first-order model for the calcium dynamics:

$$dC_x(t)/dt = -C_x(t)/\tau_C + k\left[C_{x+dx}(t) - 2C_x(t) + C_{x-dx}(t)\right] + f[V_x(t)] + \epsilon_{xt},$$

where $C_x(t)$ denotes the calcium concentration in compartment x at time t, k is a diffusion constant, $k\left[C_{x+dx}(t) - 2C_x(t) + C_{x-dx}(t)\right]$ represents diffusion within the dendrite (this assumes that x corresponds to an interior compartment of a linear segment of the dendrite, and may be easily modified in the case of a boundary or branching compartment), and $f(V)$ is a nonlinear function corresponding to voltage-dependent calcium influx. (Note that we have approximated the calcium channel dynamics as instantaneous here; this is another assumption that can be relaxed.)

Now the important point is that the linear k and $1/\tau_C$ terms here are relatively small; in the subthreshold regime the calcium concentration changes much more slowly than the voltage. We can take advantage of this fact: if we sample sufficiently rapidly along the dendritic tree, then we may obtain $C_x(t)$ (up to some observation noise) and then numerically subtract the estimated $dC_x(t)/dt$ from the linear terms on the right hand side of our dynamics equation to obtain, finally, our observation y_t, which will correspond to a nonlinear measurement $f[V_x(t)]$ of the voltage. Of course, this will be a noisy measurement; the variance of this noise can be computed given the variance of the dynamics noise ϵ and the calcium-sensitive observation fluorescence noise.

Now we can apply our direct optimization methods to infer the spatiotemporal voltage V given the observations Y; we optimize

$$\log p(V|Y) = \log p(Y|V) + \log p(V),$$

where the autoregressive log-prior $\log p(V)$ is a quadratic function of V, and the log-likelihood

$$\begin{aligned}\log p(Y|V) &= \sum_t \log p(y_t|V_{x(t)}(t)) \\ &= \sum_t -\frac{1}{2\sigma_t^2}\left(y_t - f[V_{x(t)}(t)]\right)^2 + const.\end{aligned}$$

is in general non-concave, due to the nonlinear nature of the observations; here $x(t)$ denotes the compartment x that was imaged at time t, and σ_t^2 is the effective variance of the Gaussian observation noise.

The non-concave nature of the log-posterior here makes it essential to find a good initialization for the optimizer. One workable strategy is to set initialize $V_{x(t)}(t) = f^{-1}(y_t)$, and then to run a Kalman smoother over this data with $V_{x(t)}(t)$ held fixed (i.e., artificially set the observation noise to zero at these points) to obtain a full initial V; then from here we obtain

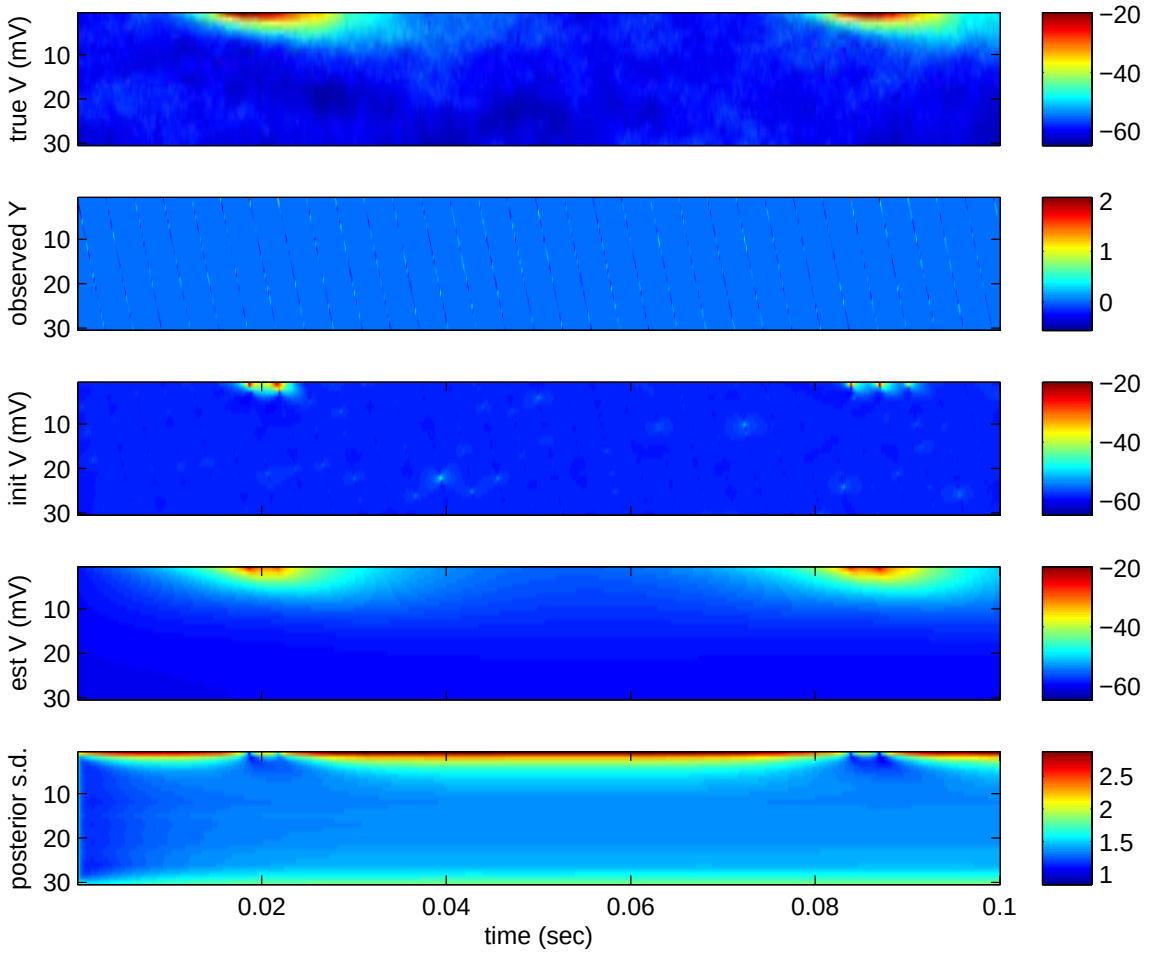

Figure 6: Inferring spatiotemporal voltage from noisy, subsampled simulated calcium-sensing recordings. Top: true spatiotemporal voltage on a simulated linear dendritic branch. Compartment 1 of this model neuron recieved a periodic current input; color indicates voltage response. Panel 2: observed data. We used the subtraction method described in the text to reduce the noisy calcium measurement to a noisy, nonlinear measurement of the voltage, $y_t = f[V_{x(t)}(t)] + \eta_t$, where η_t in this case is independent Gaussian noise. Here the voltage-dependent calcium current $f(V)$ had an activation potential at -20 mV (i.e., the calcium current is effectively zero at voltages significantly below -20 mV; at voltages > 10 mV the current is ohmic, i.e., linearly related to the voltage). We observe a spatially-rastered subsampled version of this y_t signal here. Note, as in Fig. 3, the low effective SNR. Panel 3: initialized estimate computed via modified Kalman filter described in text. Panel 4: inferred voltage signal given noisy, subsampled observations y_t shown in panel 2. Bottom panel: posterior standard deviation of the voltage estimate shown in panel 4. Note that only a few y_t observations — those corresponding to superthreshold calcium currents $f[V(t)]$ — provide most of the information here; when no superthreshold observations are available (e.g., between times 0.04 and 0.06 sec), the estimated voltage misses the small fluctuations in the true voltage response, although the superthreshold spatiotemporal voltage transients are reconstructed fairly accurately. Despite the high dimensionality ($> 10^4$) of the inferred spatiotemporal voltage here, the optimal filter requires just a couple seconds to run on a laptop computer.

a local maximum of the full log-posterior $\log p(Y|V)$. This simple strategy works well when the noise is small and the likelihood term is strong (and the objective $\log p(Y|V)$ is sharply peaked near the initialized V), but fails when the noise becomes larger and the observations become less trustworthy. In the latter case, the prior dominates and the optimizer selects $V \equiv V_{rest}$ as the solution (since this point maximizes $p(V)$), and this simple MAP approach fails; in these cases more intricate inference methods (e.g., expectation propagation, or Monte Carlo approaches) are necessary.

Fig. 6 illustrates a simulated example. (See also (Huys and Paninski, 2009) for an example application to a single-compartment model with more detailed (nonlinear) calcium channel and voltage dynamics.) We see that in this case superthreshold voltages are recovered fairly well, but as expected, details of the subthreshold voltage are lost, due to the thresholding inherent in the calcium observation.

3.2 Constrained optimization problems may be handled easily via the log-barrier method

So far we have assumed that the MAP path may be computed via an unconstrained smooth optimization. In many examples of interest we have to deal with constrained optimization problems instead. In particular, nonnegativity constraints arise frequently on physical grounds; forward-backward methods based on Gaussian approximations for the forward distribution $p(q_t|Y_{0:t})$ typically do not accurately incorporate these constraints. To handle these constrained problems while exploiting the fast tridiagonal techniques discussed above, we can employ standard interior-point (aka "barrier") methods (Boyd and Vandenberghe, 2004; Cunningham et al., 2008; Koyama and Paninski, 2009). The idea is to replace the constrained concave problem

$$\hat{Q}_{MAP} = \arg \max_{Q:q_t \geq 0} \log p(Q|Y)$$

with a sequence of unconstrained concave problems

$$\hat{Q}_\epsilon = \arg \max_Q \log p(Q|Y) + \epsilon \sum_t \log q_t;$$

clearly, $\hat{Q}_\epsilon$ satisfies the nonnegativity constraint, since $\log u \to -\infty$ as $u \to 0$. (We have specialized to the nonnegative case for concreteness, but the idea may be generalized easily to any convex constraint set; see (Boyd and Vandenberghe, 2004) for details.) Furthermore, it is easy to show that if $\hat{Q}_{MAP}$ is unique, then $\hat{Q}_\epsilon$ converges to $\hat{Q}_{MAP}$ as $\epsilon \to 0$.

Now the key point is that the Hessian of the objective function $\log p(Q|Y) + \epsilon \sum_t \log q_t$ retains the block-tridiagonal properties of the original objective $\log p(Q|Y)$, since the barrier term contributes a simple diagonal term to the Hessian H. Therefore we may use the $O(T)$ Newton iteration to obtain $\hat{Q}_\epsilon$, for any ϵ, and then sequentially decrease ϵ (in an outer loop) to obtain $\hat{Q}$. We give a few applications of this barrier approach below.

3.2.1 Example: the integrate-and-fire model with hard threshold

As discussed above, the integrate-and-fire model may be written in state space form (Paninski, 2006; Paninski et al., 2008; Koyama and Paninski, 2009): the simplest model that incorporates noise, leakiness, and external input is

$$V_{t+dt} = V_t + (-gV_t + I_t)\, dt + \sqrt{dt}\sigma\epsilon_t, \quad \epsilon_t \sim \mathcal{N}(0,1),$$

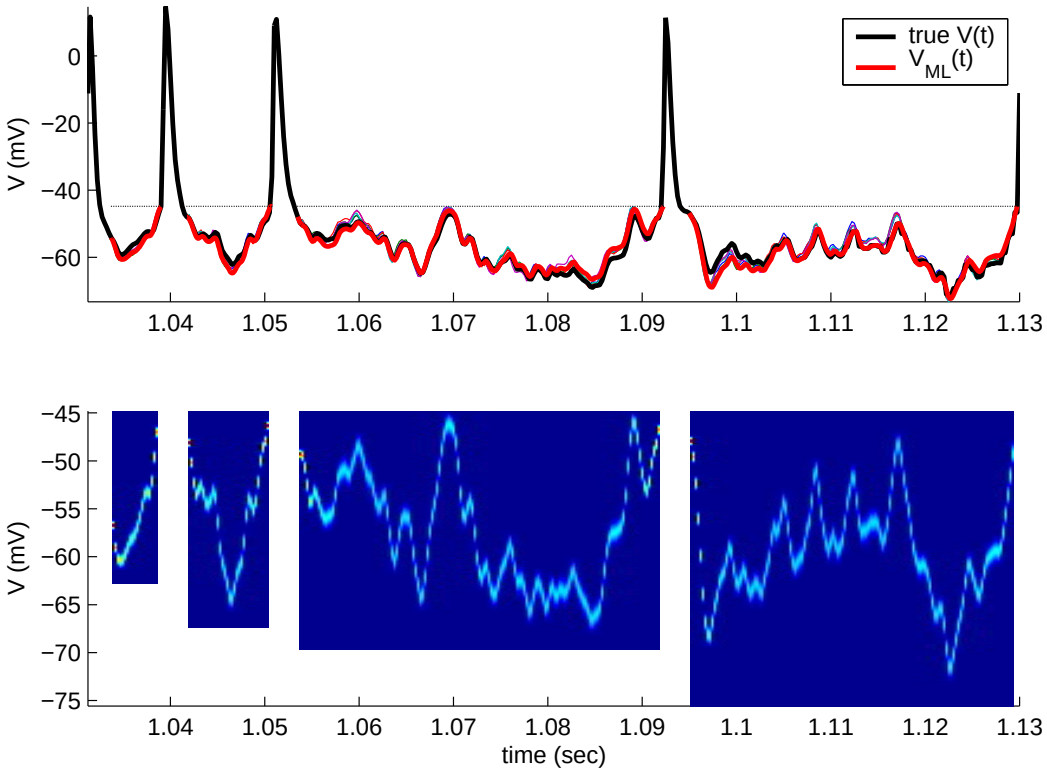

Figure 7: Computing the MAP voltage path given *in vitro* physiological data; a single cortical cell was driven by a white noise current input I_t and the voltage V_t was recorded. **Top**: Comparison of true voltage path (bold black trace) with computed $V_{MAP}(t)$ (bold gray trace) and samples from conditional distribution $p(V_t|Y, \{I_t\}_{0\leq t\leq T}, \theta)$ (thin traces, partially obscured; computed by forward-backward sampling method). Trace shown is a randomly chosen segment of a 25-second long white noise experiment; dashed trace indicates voltage threshold estimated via the maximum likelihood method described in (Paninski et al., 2004b). $V_{MAP}(t)$ is computed via tridiagonal quadratic programming under the hard-threshold model $dV = (-gV_t + I_t)dt + \sigma dB_t$; the model parameters θ were estimated by fitting an AR(1) model to the observed intracellular V_t given I_t, using an independent segment of training data (data not shown). Note that $V_{MAP}(t)$ matches the true voltage V_t quite well; see (Paninski et al., 2004b; Paninski, 2006) for further details. **Bottom**: Conditional distributions $p(V_t|Y, \{I_t\}_{0\leq t\leq T}, \theta)$, computed via forward-backward method. White space indicates gaps in time where voltage was superthreshold, and thus not modeled. Note small scale of estimated intracellular current noise σ (Mainen and Sejnowski, 1995).

with the appropriate boundary conditions: $V_t \leq V_{th}$, and $V_{t^+} = V_{reset}$ if $V_{t^-} = V_{th}$. The observation y_t in this model has a very simple form: $y_t = 1$ if $V_t = V_{th}$, and $y_t = 0$ otherwise.

Now, computing the MAP voltage path in this model given an observed sequence of spikes via the barrier method is straightforward: we simply need to maximize the log-posterior

$$\log p(V|Y) = -\frac{1}{2\sigma^2 dt}\sum_t \left[V_{t+dt} - (V_t - gV_t + I_t)\right]^2 + const.,$$

which is defined on the domain of voltage paths satisfying the conditional boundary conditions given the observed spike train Y: $V_t \leq V_{th}$, and $V_t = V_{th}$ at spike times. The Hessian of the Gaussian $\log p(V)$ term is tridiagonal, as usual. The Hessian of the barrier term enforcing the subthreshold constraint $V_t \leq V_{th}$ is diagonal, and therefore our full objective function

$$-\frac{1}{2\sigma^2 dt}\sum_t \left[V_{t+dt} - (V_t - gV_t + I_t)\right]^2 + \epsilon \sum_t \log(V_{th} - V_t)$$

(with V_t simply defined as V_{th} at all spike times), may be optimized via Newton's method in $O(T)$ time. See Fig. 7 for an example application to *in vitro* data. This method may also be easily applied to the common-noise model discussed in (Iyengar, 1985; Paninski, 2006; de la Rocha et al., 2007), in which a common noise source drives a network of IF cells, and to the multidimensional linear IF neuron considered by (Badel et al., 2008a) and others; in each of these cases, the loglikelihood may be written as a block tridiagonal quadratic form (where in the one-dimensional case described above, the block size was just one).

3.2.2 Example: point process smoothing under Lipschitz or monotonicity constraints on the intensity function

A standard problem in neural data analysis is to smooth point process observations; that is, to estimate the underlying firing rate $\lambda(t)$ given single or multiple observations of a spike train (Kass et al., 2003). One simple approach to this problem is to model the firing rate as $\lambda(t) = f(q_t)$, where $f(.)$ is a convex, log-concave, monotonically increasing nonlinearity (Paninski, 2004) and q_t is an unobserved function of time we would like to estimate from data. Of course, if q_t is an arbitrary function, we need to contend with overfitting effects; the "maximum likelihood" $\hat{Q}$ here would simply set $f(q_t)$ to zero when no spikes are observed (by making q_t very large and negative) and $f(q_t)$ to be very large when spikes are observed (by making q_t very large and positive).

A simple way to counteract this overfitting effect is to include a penalizing prior; for example, if we model q_t as a linear-Gaussian autoregressive process

$$q_{t+1} = q_t + \epsilon_t,\ \epsilon_t \sim \mathcal{N}(0, \sigma^2),$$

then computing $\hat{Q}_{MAP}$ leads to a tridiagonal optimization, as discussed above. (The resulting model, again, is mathematically equivalent to those applied in (Smith and Brown, 2003; Truccolo et al., 2005; Kulkarni and Paninski, 2007a; Czanner et al., 2008; Vidne et al., 2009).) Here $1/\sigma^2$ acts as a regularization parameter: if σ^2 is small, the inferred $\hat{Q}_{MAP}$ will be very smooth (since large fluctuations are penalized by the Gaussian autoregressive prior), whereas if σ^2 is large, then the prior term will be weak and $\hat{Q}_{MAP}$ will fit the observed data more closely.

A different method for regularizing Q was introduced by (Coleman and Sarma, 2007). The idea is to impose hard Lipschitz constraints on Q, instead of the soft quadratic penalties imposed in the Gaussian state-space setting: we assume

$$|q_t - q_s| < K|t - s|$$

for all (s, t), for some finite constant K. (If q_t is a differentiable function of t, this is equivalent to the assumption that the maximum absolute value of the derivative of Q is bounded by

K.) The space of all such Lipschitz Q is convex, and so optimizing the concave loglikelihood function under this convex constraint remains tractable. (Coleman and Sarma, 2007) presented a powerful method for solving this optimization problem (their solution involved a dual formulation of the problem and an application of specialized fast min-cut optimization methods). In this one-dimensional temporal smoothing case, we may solve this problem in a somewhat simpler way, without any loss of efficiency, using the tridiagonal log-barrier methods described above. We just need to rewrite the constrained problem

$$\max_Q \log p(Q|Y) \ s.t. \ |q_t - q_s| < K|t - s| \ \ \forall s, t$$

as the unconstrained problem

$$\max_Q \log p(Q|Y) + \sum_t b\left(\frac{q_t - q_{t+dt}}{dt}\right),$$

with dt some arbitrarily small constant and the hard barrier function $b(.)$ defined as

$$b(u) = \begin{cases} 0 & |u| < K \\ -\infty & otherwise. \end{cases}$$

The resulting concave objective function is non-smooth, but may be optimized stably, again, via the log-barrier method, with efficient tridiagonal Newton updates. (In this case, the Hessian of the first term $\log p(Q|Y)$ with respect to Q is diagonal and the Hessian of the penalty term involving the barrier function is tridiagonal, since $b(.)$ contributes a two-point potential here.) We recover the standard state-space approach if we replace the hard-threshold penalty function $b(.)$ with a quadratic function; conversely, we may obtain sharper estimates of sudden changes in q_t if we use a concave penalty $b(.)$ which grows less steeply than a quadratic function (so as to not penalize large changes in q_t as strongly), as discussed by (Gao et al., 2002). Finally, it is interesting to note that we may also easily enforce monotonicity constraints on q_t, by choosing the penalty function $b(u)$ to apply a barrier at $u = 0$; this is a form of isotonic regression (Silvapulle and Sen, 2004), and is useful in cases where we believe that a cell's firing rate should be monotonically increasing or decreasing throughout the course of a behavioral trial, or as a function of the magnitude of an applied stimulus.

3.2.3 Example: fast nonnegative deconvolution methods for inferring spike times from noisy, intermittent calcium traces

Calcium-imaging methods have become popular recently for investigating the behavior of populations of neurons (Cossart et al., 2003; Kerr et al., 2005; Ohki et al., 2006). However, calcium dynamics evolve over a much slower timescale than do spike trains; hence deconvolution techniques are necessary to recover the spike train from the available noisy, intermittent calcium observations.

A reasonable first-pass model of calcium dynamics is as follows. Let y_t denote the observed fluorescence signal, and q_t the underlying, unobserved calcium signal. (Note that q_t might be measured on a finer time scale than y_t.) For the dynamics, we use a first-order model,

$$q_{t+dt} = q_t - \frac{q_t}{\tau}dt + bn_t,$$

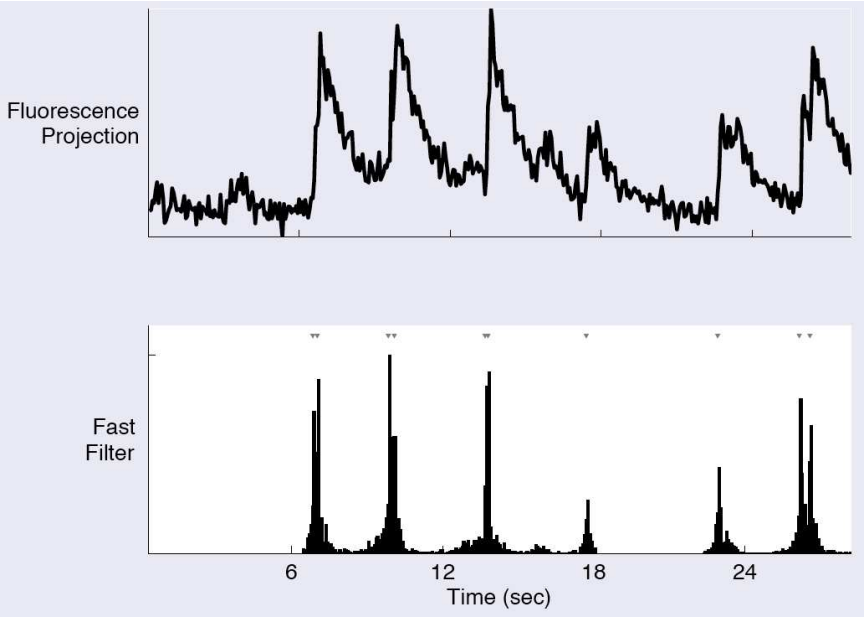

Figure 8: Example of fast nonnegative calcium deconvolution method applied to data recorded *in vivo*. Top: observed calcium fluorescence signal. Bottom: spiking signal n_t inferred via fast nonnegative method. Gray triangles mark true spike times (recorded via intracellular electrode); note that spike times are inferred accurately. The intracellular recording was used only to check the accuracy of the inference; all model parameters were fit from the observed calcium fluorescence data only. See (Vogelstein et al., 2008a) for details.

for some positive constants τ, b; here n_t denotes the spike train which is driving these calcium dynamics, so the calcium jumps up with each spike and then decays exponentially with time constant τ. Finally, assume linear Gaussian observations

$$y_t = q_t + e_t;\ e_t \sim \mathcal{N}(0, \sigma^2).$$

Finally, we assume a log-concave, nonnegative prior on n_t; a convenient example is $\log p(n) = -\lambda \sum_t n_t dt$, which corresponds to an i.i.d. exponential prior on n. See (Vogelstein et al., 2008b) for a much more detailed discussion of the assumptions involved with this simple model.

Now maximizing $p(Q|Y)$ is nearly routine. The likelihood term $\log p(Y|Q)$ contributes a diagonal term to the Hessian, while the dynamics term $p(q_{t+dt}|q_t)$ (including a barrier term to enforce the nonnegativity of n_t) contributes a tridiagonal term. Each Newton iteration may once again be done in $O(T)$ time. See Fig. 8 for an example application to *in vivo* data.

3.2.4 Example: inferring presynaptic inputs given postsynaptic voltage recordings

To further illustrate the flexibility of this method, let's look at a multidimensional example. Consider the problem of identifying the synaptic inputs a neuron is receiving: given voltage recordings at a postsynaptic site, is it possible to recover the time course of the presynaptic

conductance inputs? This question has received a great deal of experimental and analytical attention (Borg-Graham et al., 1996; Peña, J.-L. and Konishi, M., 2000; Wehr and Zador, 2003; Priebe and Ferster, 2005; Murphy and Rieke, 2006; Huys et al., 2006; Wang et al., 2007; Xie et al., 2007; Paninski, 2009), due to the importance of understanding the dynamic balance between excitation and inhibition underlying sensory information processing.

We may begin by writing down a simple state-space model for the evolution of the post-synaptic voltage and conductance:

$$\begin{aligned} V_{t+dt} &= V_t + dt\left[g^L(V^L - V_t) + g_t^E(V^E - V_t) + g_t^I(V^I - V_t)\right] + \epsilon_t \\ g_{t+dt}^E &= g_t^E - \frac{g_t^E}{\tau_E} + N_t^E \\ g_{t+dt}^I &= g_t^I - \frac{g_t^I}{\tau_I} + N_t^I. \end{aligned} \tag{19}$$

Here g_t^E denotes the excitatory presynaptic conductance at time t, and g_t^I the inhibitory conductance; V^L, V^E, and V^I denote the leak, excitatory, and inhibitory reversal potentials, respectively. Finally, ϵ_t denotes an unobserved i.i.d. current noise with a log-concave density, and N_t^E and N_t^I denote the presynaptic excitatory and inhibitory inputs (which must be nonnegative on physical grounds); we assume these inputs also have a log-concave density.

Assume V_t is observed noiselessly for simplicity. Then let our observed variable $y_t = V_{t+dt} - V_t$ and our state variable $q_t = (g_t^E \; g_t^I)^T$. Now, since g_t^I and g_t^E are linear functions of N_t^I and N_t^E (for example, g_t^I is given by the convolution $g_t^I = N_t^I * \exp(-t/\tau_I)$), the log-posterior may be written as

$$\begin{aligned} \log p(Q|Y) &= \log p(Y|Q) + \log p(N_t^I, N_t^E) + const. \\ &= \log p(Y|Q) + \sum_{t=1}^{T} \log p(N_t^E) + \sum_{t=1}^{T} \log p(N_t^I) + const.,\ N_t^E, N_t^I \geq 0; \end{aligned}$$

in the case of Gaussian current noise ϵ_t with variance $\sigma^2 dt$, for example, we have

$$\log p(Y|Q) = -\frac{1}{2\sigma^2 dt} \sum_{t=2}^{T} \left[V_{t+dt} - \Big(V_t + dt\big(-gV_t + g_t^I(V^I - V_t) + g_t^E(V^E - V_t)\big)\Big)\right]^2 + const.;$$

this is a quadratic function of Q.

Now applying the $O(T)$ log-barrier method is straightforward; the Hessian of the log-posterior $\log p(Q|Y)$ in this case is block-tridiagonal, with blocks of size two (since our state variable q_t is two-dimensional). The observation term $\log p(Y|Q)$ contributes a block-diagonal term to the Hessian; in particular, each observation y_t contributes a rank-1 matrix of size 2×2 to the t-th diagonal block of H. (The low-rank nature of this observation matrix reflects the fact that we are attempting to extract two variables — the excitatory and inhibitory conductances at each time step — given just a single voltage observation per time step.)

Some simulated results are shown in Fig. 9. We generated Poisson spike trains from both inhibitory and excitatory presynaptic neurons, then formed the postsynaptic current signal I_t by contaminating the summed synaptic and leak currents with white Gaussian noise as in equation (19), and then used the $O(T)$ log-barrier method to simultaneously infer the presynaptic conductances from the observed current I_t. The current was recorded at 1 KHz (1 ms bins), and we reconstructed the presynaptic activity at the same time resolution. We

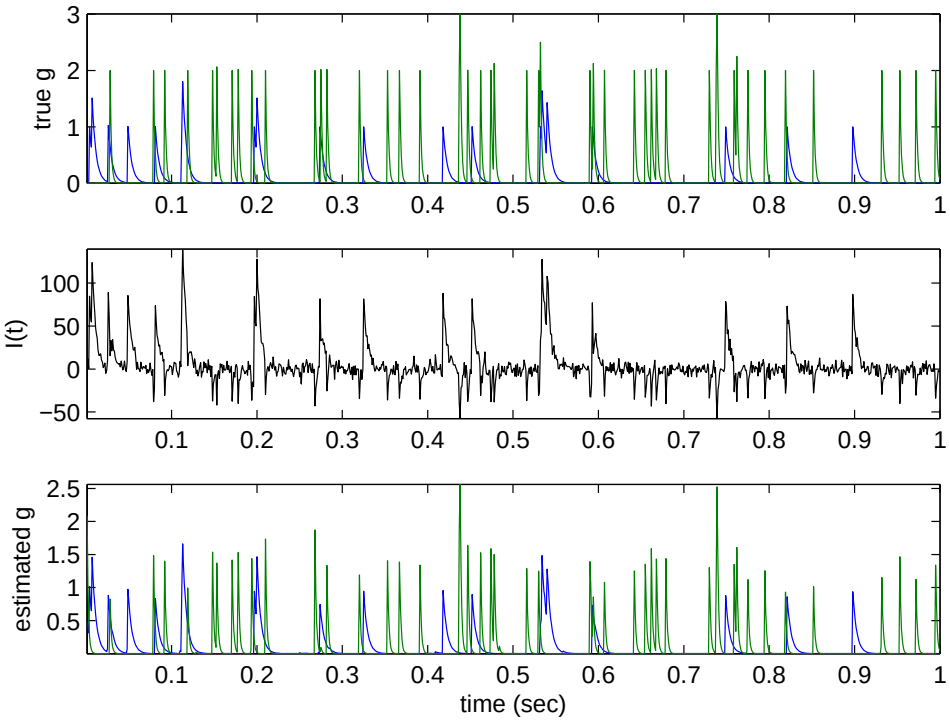

Figure 9: Inferring presynaptic inputs given simulated postsynaptic voltage recordings. **Top**: true simulated conductance input (green indicates inhibitory conductance; blue excitatory). **Middle**: observed noisy current trace from which we will attempt to infer the input conductance. **Bottom**: Conductance inferred by nonnegative MAP technique. Note that inferred conductance is shrunk in magnitude compared to true conductance, due to the effects of the prior $p(N_t^E)$ and $p(N_t^I)$, both of which peak at zero here; shrinkage is more evident in the inferred inhibitory conductance, due to the smaller driving force (the holding potential in this experiment was -62 mV, which is quite close to the inhibitory reversal potential; as a result, the likelihood term is much weaker for the inhibitory conductance than for the excitatory term). Inference here required about one second on a laptop computer per second of data (i.e., real time), at a sampling rate of 1 KHz.

see that the estimated $\hat{Q}$ here does a good job extracting both excitatory and inhibitory synaptic activity given a single trace of observed somatic current; there is no need to average over multiple trials. It is worth emphasizing that we are inferring two presynaptic signals here given just one observed postsynaptic current signal, with limited "overfitting" artifacts; this is made possible by the sparse, nonnegatively constrained nature of the inferred presynaptic signals. For simplicity, we assumed that the membrane leak, noise variance, and synaptic time constants τ_E and τ_I were known here; we used exponential (sparsening) priors $p(N_t^E)$ and $p(N_t^I)$, but the results are relatively robust to the details of these priors (data not shown). See (Huys et al., 2006; Huys and Paninski, 2009; Paninski, 2009) for further details and extensions, including methods for inferring the membrane parameters

3.2.5 Example: nonparametric methods for estimating the current emission densities in hidden Markov models of ion channel data

In the previous chapter we discussed hidden Markov models for ion channel data. Here the hidden state q_t represented the physical configuration of the ion channel molecule, which is assumed to evolve in a Markovian manner between a few discrete quasistable states. Given the configuration q_t, the channel passes an observed current y_t, which may be considered a random variable with some conditional distribution $p(y_t|q_t)$. We discussed Gaussian models for the observation densities $p(y_t|q_t)$: we saw that the M-step is quite straightforward in this case. However, there is little biophysical justification for this Gaussian assumption, and in fact we can fit much richer nonparametric models with little extra computational effort in this setting. Recall that the E-step in this model is performed using the standard discrete-space forward-backward recursion, which does not make any special use of the form of the observation density $p(y_t|q_t)$; thus we may assume that the observations are drawn from a conditional distribution with a continuous density function $f(y_t|q_t)$ without any loss of computational efficiency in the E-step.

Of course, if we allow $f(y_t|q_t)$ to be completely arbitrary, we will overfit: the "optimal" density (in the sense of maximizing the loglikelihood on the training data) will place all of its mass on the observed data points $\{y_t\}$, and no mass on any unobserved points y; of course, such a density will typically generalize very poorly, and will therefore be very far from optimal when used to perform inference on any test data. We can use a penalized maximum nonparametric likelihood approach to avoid overfitting here. Recall that the EM algorithm can always be modified to (locally) maximize a log-posterior, instead of the log-likelihood. Thus we modify the standard M-step in this HMM slightly, to optimize

$$\left(\sum_t p(q_t = i|Y) \log f(y_t|i) \right) + \log P[f(.|i)]$$

with respect to $f(.|i)$, where $P[f]$ represents our prior distribution on f. (See (Robinson et al., 2008) for an application of a very similar model in the context of change-point detection.) In this setting it is natural to choose a prior $P(f)$ that penalizes the roughness of f: one convenient prior is of the truncated tridiagonal Gaussian form (Paninski, 2005),

$$\log P(f) = -\lambda \int \left(\frac{\partial f}{\partial y} \right)^2 dy + const.,$$

under the constraints

$$f \geq 0, \int f = 1.$$

Note that this log-prior is concave as a function of f, and the Hessian of the integral term $\int \left(\frac{\partial f}{\partial y} \right)^2 dy$ is tridiagonal, when we discretize in y. In addition, the loglikelihood $\log f(.|i)$ is concave in f, with a diagonal Hessian; a log-barrier term enforcing the nonnegativity constraint $f \geq 0$ is similarly concave in f, with a diagonal Hessian. The only slight complication is the normalization constraint $\int f = 1$, but this may be written in simple vector form as $1^T f dy = 1$, and handling this simple rank-one linear constraint is easy via the Woodbury lemma. Thus we may perform the M-step here in $O(d)$ time, where d is the number of bins we use to represent the densities $f(.|i)$. See (Good and Gaskins, 1971; Bialek et al., 1996; Holy, 1997) for discussions of some closely related approaches to penalized density estimation.

Two small technical points are worth mentioning. First, as mentioned in (Paninski, 2005), this Gaussian form for $\log P(f)$ makes rigorous sense only if f is defined on a compact interval. We can easily finesse this by nonlinearly rescaling y (which would typically be considered an unbounded variable in this ion channel application) onto a compact (bounded) interval via some convenient continuous squashing function, and then constraining our estimated f to approach zero continuously on the endpoints of this interval, with a straightforward modification of our penalty function $\log P(f)$. Second, it is often a good idea to constrain our densities $f(.|i)$ to not approach zero too sharply, since if $f(y|i) = 0$ at some observed point $y = y_t$, the E-step will conclude with perfect confidence that $q_t \neq i$, which is typically not a justified conclusion except in very high-SNR recordings. One crude method for preventing this zero-f problem is to simply stop our interior-point algorithm early, before the penalty term on the log-barrier function enforcing the nonnegativity constraint $f \geq 0$ becomes negligible. This will effectively modify our objective function to include an additional small penalty preventing the the log-density $\log f$ from "running away" to $-\infty$.

3.2.6 Example: optimal control of spike timing

In the last section we discussed the problem of optimal spike train decoding in the context of neural motor prosthetic devices: we want to read out a neural signal and decode this information to drive a useful control signal. On the other hand, sensory neural prosthetics, such as cochlear (Loizou, 1998) or retinal implants (Weiland et al., 2005), require us to solve the converse problem: given an auditory or visual sensory signal, how can we transduce this information by stimulating sensory neurons (e.g., in the cochlear or optic nerve) to fire in such a way that some percept of the sensory environment can be restored to the patient?

More precisely, any sensory neural prosthetic must in principle solve the following sequence of signal-processing problems in real time:

1. Sense the stimulus (e.g., via a microphone or video camera).
2. Transduce the sensory signal into a desired multineuronal spike train r_{target} (e.g., via a model of cochlear nerve or retinal ganglion cell response properties).
3. Stimulate neurons to fire with the desired output pattern r_{target}.

We will focus on the third step here: how can we choose a stimulus that will cause a neuron to output the target spike train r_{target} with optimal precision? Of course, this problem as stated is ill-posed: if we can inject any arbitrary current into a cell, for example, we can simply make the cell fire with any desired pattern. Instead, we need to solve a constrained optimization problem, since there are limitations on the stimuli we can safely apply in any physiological preparation.

Thus we want to solve a problem of the form

$$\min_{s \in \mathcal{S}} E_{p(r|s)} C(r, r_{target}),$$

where s denotes the applied stimulus we are optimizing over, $\mathcal{S}$ denotes a constraint space of allowed stimuli s, and $C(r, r_{target})$ denotes some cost function measuring the discrepancy between the desired output spike train r_{target} and the actual output resopnse r. The expectation is taken over $p(r|s)$, the conditional probability of the response r given s; in general, we must supply a concrete response model to compute $p(r|s)$.

Unfortunately, depending on the on the precise form of the cost function $C(.,.)$ and the details of the conditional distribution $p(r|s)$, it may be quite challenging to solve this optimization problem to choose the best stimulus s in real time. Thus it is reasonable to search for a more tractable approximation to this problem. One natural approach is to optimize a likelihood-based objective function instead. We will discuss two examples here.

First, let's consider direct electrical stimulation of the circuit, via single or multiple stimulating electrodes. We start with a tractable model of the responses $p(r|s)$. Let's assume that the response of the stimulated neuron may be modeled as a point process with rate given by our usual leaky, noisy integrator:

$$\begin{aligned} \lambda_t &= f(V_t + h_t) \\ V_{t+dt} &= V_t + dt\,(-gV_t + aI_t) + \sqrt{dt}\sigma\epsilon_t, \quad \epsilon_t \sim \mathcal{N}(0,1). \end{aligned}$$

Here the stimulation gain parameter a must be calibrated (in general, this gain factor will depend on the membrane resistance, the electrode impedance, and the distance from the electrode tip to the cell). The spike history effect h_t is formed by convolving a refractory term against the target spike train r_{target}; note that we consider h_t fixed (i.e., we will not attempt to average over history-dependent variability in h_t here (Toyoizumi et al., 2009)).

Now to choose the best stimulus current I_t, we optimize the penalized loglikelihood of r_{target} given I_t, under some physiological constraints:

$$\begin{aligned} &\max_{(V,I):|I_t|<c} \log p(D|V) + \log p(V|I) - R(I) \\ = &\max_{(V,I):|I_t|<c} \sum_t \left(n_t \log \lambda_t - \lambda_t dt\right) - \frac{1}{2\sigma^2 dt}\sum_t \left[V_{t+dt} - V_t - dt\,(-gV_t + aI_t)\right]^2 - c_2 \sum_t I_t^2 dt - c_3 \sum_t J_t^2 dt, \end{aligned}$$

where J_t is a low-pass filtered version of the injected current I:

$$J_{t+dt} = J_t - dt\left(\frac{J_t}{\tau_J} + I_t\right).$$

The current is bounded to remain within an acceptable safety range $|I_t| < c$, while the quadratic cost $R(I)$ encourages the optimizer to find a low-energy current, and in addition ensures that no significant charge builds up on the electrode over timescales τ_J (since high charge densities lead to cell damage; see, e.g., (Sekirnjak et al., 2006) for further discussion). The constraint parameters (c, c_2, c_3) are set according to physiological safety margins and determine, along with the gain parameter a, the tradeoff between the size of the applied current and the reliability of the output spike train: making c larger, for example, leads to larger allowed currents and therefore more reliable spike outputs.

Now, to optimize this objective efficiently, note that our loglikelihood $\log p(D|V) + \log p(V|I)$ is jointly concave in (V, I) when $f(.)$ is convex and log-concave. If we introduce a barrier function to enforce the boundedness constraint $|I_t| < c$, the Hessian of the objective with respect to (V, I) is easily seen to be block-banded, with blocks of size 2×2 (one dimension each for V_t and I_t) and we can apply our fast interior-point algorithm easily. In particular, this optimization can in principle be performed in real time, with online updating of the optimal I_t as new information about the target spike train (and, if it is possible to record from the stimulated neurons, about the resulting stimulated spike trains) becomes available. This procedure could in principle also be generalized to stimulate multiple neurons on multiple

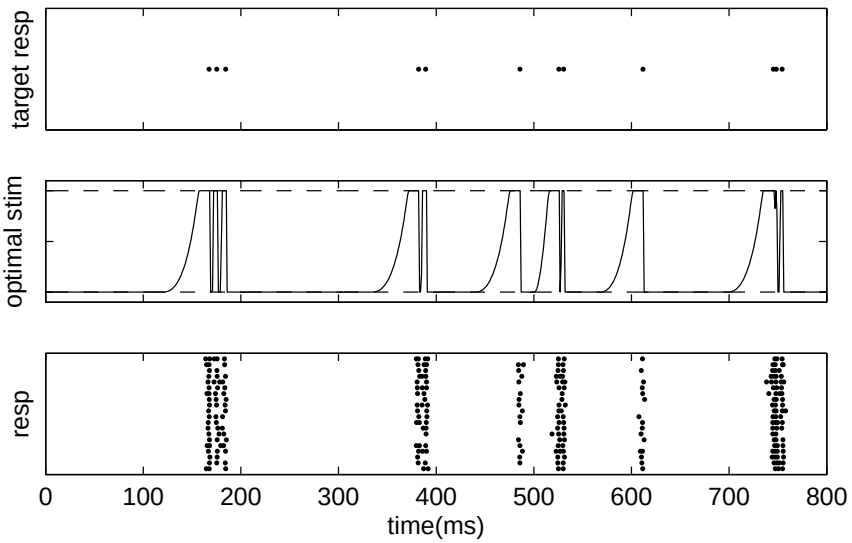

Figure 10: Example of an inferred optimal current. Top: target spike response. Middle: inferred optimal current. The regularization parameters c_2 and c_3 were set to zero here, for simplicity. Dashed lines show upper and lower current bounds ($\pm c$). Bottom: twenty sample responses, to test the reliability of the resulting spike train given the optimal current injection.

electrodes simultaneously: in this case we would need to introduce a matrix of gain factors a_{ij} describing the influence of electrode i on neuron j.

The second setting we will consider involves photostimulation of the circuit instead of direct electrical stimulation. Optical methods for perturbing neural circuits with high spatiotemporal precision have developed dramatically over the last decade (Callaway and Yuste, 2002): for example, neurons can be excited locally via optical uncaging of caged glutamate compounds (Nikolenko et al., 2007; Matsuzaki et al., 2008; Nikolenko et al., 2008), or via photostimulation following membrane expression of channelrhodopsin-2- and halorhodopsin channels (Boyden et al., 2005; Han and Boyden, 2007; Mohanty et al., 2008). To develop optimal stimulation protocols in this setting, we again need to start with a model of the neural response $p(r|s)$. We begin here with a simple first-order model for the dynamics of conductances driven by photostimulation (although, again, more complicated models may be handled using similar techniques). We model the responses as a point process driven by a conductance-based leaky integrator model:

$$\begin{aligned}
\lambda_t &= f(V_t + h_t) \\
V_{t+dt} &= V_t + dt\left(-gV_t + g_t^i(V^i - V_t) + g_t^e(V^e - V_t)\right) + \sqrt{dt}\sigma\epsilon_t, \quad \epsilon_t \sim \mathcal{N}(0,1) \\
g_{t+dt}^i &= g_t^i + dt\left(-\frac{g_t^i}{\tau_i} + a_{ii}L_t^i + a_{ie}L_t^e\right) \\
g_{t+dt}^e &= g_t^e + dt\left(-\frac{g_t^e}{\tau_i} + a_{ee}L_t^e + a_{ei}L_t^i\right)
\end{aligned}$$

Here g_t^e and g_t^i are the excitatory and inhibitory conductances at time t, with reversal potentials V^i and V^e, respectively; L_t^e and L_t^i are the applied light intensities in the excitatory

and inhibitory channel, and the weights a_{ie}, etc., represent a 2×2 gain matrix summarizing how influential the optical pulse L_t^e is upon the conductace g_t^i (this is a function of the optical qualities of the stimulating laser, and the overlap in the absorption spectra of the optical sensors, as well as the concentration of the caged compound in the neuropil or opsin channel expression in the membrane).

These system dynamics are nonlinear in the state variables (V, g^e, g^i) (due to the multiplication of the conductances g_t with the voltage V_t), and therefore the resulting optimization problem is nonconvex. We can obtain a good initializer by solving the following approximate current-based system instead:

$$V_{t+dt} = V_t + dt\left(-gV_t + g_t^i(V^i - V_*) + g_t^e(V^e - V_*)\right) + \sqrt{dt}\sigma\epsilon_t, \quad \epsilon_t \sim \mathcal{N}(0,1),$$

where V_* is an averaged (non-time-varying) voltage and the conditional intensity function λ_t and conductances g_t follow the same dynamics defined above.

Now we may write down our optimization problem. Our major constraint in the electrical stimulation setting was that the applied current I_t was bounded in absolute value; here, the applied light intensities L_t^i and L_t^e are constrained to be nonnegative and bounded above. So we want to solve the problem

$$\begin{aligned}
&\max_{(V,L):0\le L_t^i<c_i;0\le L_t^e<c_e;V_t\ge V^i} \log p(D|V) + \log p(V|L) - R(L) \\
&= \max_{(V,L):0\le L_t^i<c_i;0\le L_t^e<c_e;V_t\ge V^i} \sum_t \left(n_t \log \lambda_t - \lambda_t dt\right) - \frac{1}{2\sigma^2 dt}\sum_t \left[V_{t+dt} - V_t - dt\left(-gV_t + g_t^i(V^i - V_*) + g_t^e(V^e - V_*)\right)\right]^2 \\
&\qquad -c_1\sum_t (L_t^e)^2 dt - c_2 \sum_t (L_t^i)^2 dt.
\end{aligned}$$

Note that we have also included a lower bound on the voltage V_t, since in the deterministic conductance-based model V_t will never dip below the inhibitory equilibrium potential V^i (this bound is not included automatically in the current-based model). We have found that it is beneficial to include this bound explicitly, to prevent the optimizer from attempting to over-hyperpolarize the neuron during silent periods in the target spike train r_{target}; instead, the inhibition in this bounded model acts more like a shunt, becoming effective only when excitation is applied.

This problem can be solved with our usual fast barrier techniques once we write the Hessian in block-tridiagonal form, with each block of size 3×3 (one dimension each for V_t, g_t^e, and g_t^i). Again, generalizations to the multineuronal case are straightforward.

References

Ahmadian, Y., Pillow, J., and Paninski, L. (2009). Efficient Markov Chain Monte Carlo methods for decoding population spike trains. *Under review, Neural Computation*.

Asif, A. and Moura, J. (2005). Block matrices with l-block banded inverse: Inversion algorithms. *IEEE Transactions on Signal Processing*, 53:630–642.

Badel, L., Gerstner, W., and Richardson, M. J. E. (2008a). Spike-triggered averages for passive and resonant neurons receiving filtered excitatory and inhibitory synaptic drive. *Physical Review E (Statistical, Nonlinear, and Soft Matter Physics)*, 78:011914.

Badel, L., Lefort, S., Berger, T. K., Petersen, C. C. H., Gerstner, W., and Richardson, M. J. E. (2008b). Extracting non-linear integrate-and-fire models from experimental data using dynamic I-V curves. *Biological Cybernetics*, 99:361–370.

Badel, L., Lefort, S., Brette, R., Petersen, C. C. H., Gerstner, W., and Richardson, M. J. E. (2008c). Dynamic I-V Curves Are Reliable Predictors of Naturalistic Pyramidal-Neuron Voltage Traces. *J Neurophysiol*, 99(2):656–666.

Barbieri, R., Frank, L., Nguyen, D., Quirk, M., Solo, V., Wilson, M., and Brown, E. (2004). Dynamic analyses of information encoding in neural ensembles. *Neural Computation*, 16:277–307.

Bell, B. M. (1994). The iterated Kalman smoother as a Gauss–Newton method. *SIAM Journal on Optimization*, 4:626–636.

Bialek, W., Callan, C., and Strong, S. (1996). Field theories for learning probability distributions. *Physical Review Letters*, 77:4693–4697.

Borg-Graham, L., Monier, C., and Fregnac, Y. (1996). Voltage-clamp measurements of visually-evoked conductances with whole-cell patch recordings in primary visual cortex. *J. Physiology [Paris]*, 90:185–188.

Boyd, S. and Vandenberghe, L. (2004). *Convex Optimization*. Oxford University Press.

Boyden, E. S., Zhang, F., Bamberg, E., Nagel, G., and Deisseroth, K. (2005). Millisecond-timescale, genetically targeted optical control of neural activity. *Nat Neurosci*, 8(9):1263–1268.

Brockwell, A., Rojas, A., and Kass, R. (2004). Recursive Bayesian decoding of motor cortical signals by particle filtering. *Journal of Neurophysiology*, 91:1899–1907.

Brown, E., Frank, L., Tang, D., Quirk, M., and Wilson, M. (1998). A statistical paradigm for neural spike train decoding applied to position prediction from ensemble firing patterns of rat hippocampal place cells. *Journal of Neuroscience*, 18:7411–7425.

Brown, E., Nguyen, D., Frank, L., Wilson, M., and Solo, V. (2001). An analysis of neural receptive field plasticity by point process adaptive filtering. *PNAS*, 98:12261–12266.

Callaway, E. and Yuste, R. (2002). Stimulating neurons with light. *Current Opinion in Neurobiology*, 12:587–592.

Coleman, T. and Sarma, S. (2007). A computationally efficient method for modeling neural spiking activity with point processes nonparametrically. *IEEE Conference on Decision and Control*.

Cossart, R., Aronov, D., and Yuste, R. (2003). Attractor dynamics of network up states in the neocortex. *Nature*, 423:283–288.

Cover, T. and Thomas, J. (1991). *Elements of information theory*. Wiley, New York.

Cunningham, J. P., Shenoy, K. V., and Sahani, M. (2008). Fast Gaussian process methods for point process intensity estimation. *ICML*, pages 192–199.

Czanner, G., Eden, U., Wirth, S., Yanike, M., Suzuki, W., and Brown, E. (2008). Analysis of between-trial and within-trial neural spiking dynamics. *Journal of Neurophysiology*, 99:2672–2693.

Davis, R. and Rodriguez-Yam, G. (2005). Estimation for state-space models: an approximate likelihood approach. *Statistica Sinica*, 15:381–406.

Dayan, P. and Abbott, L. (2001). *Theoretical Neuroscience*. MIT Press.

de la Rocha, J., Doiron, B., Shea-Brown, E., Josic, K., and Reyes, A. (2007). Correlation between neural spike trains increases with firing rate. *Nature*, 448:802–806.

Donoghue, J. (2002). Connecting cortex to machines: recent advances in brain interfaces. *Nature Neuroscience*, 5:1085–1088.

Durbin, J. and Koopman, S. (2001). *Time Series Analysis by State Space Methods*. Oxford University Press.

Eden, U. T., Frank, L. M., Barbieri, R., Solo, V., and Brown, E. N. (2004). Dynamic analyses of neural encoding by point process adaptive filtering. *Neural Computation*, 16:971–998.

Fahrmeir, L. and Kaufmann, H. (1991). On Kalman filtering, posterior mode estimation and fisher scoring in dynamic exponential family regression. *Metrika*, 38:37–60.

Fahrmeir, L. and Tutz, G. (1994). *Multivariate Statistical Modelling Based on Generalized Linear Models*. Springer.

Frank, L., Eden, U., Solo, V., Wilson, M., and Brown, E. (2002). Contrasting patterns of receptive field plasticity in the hippocampus and the entorhinal cortex: An adaptive filtering approach. *J. Neurosci.*, 22(9):3817–3830.

Gao, Y., Black, M., Bienenstock, E., Shoham, S., and Donoghue, J. (2002). Probabilistic inference of arm motion from neural activity in motor cortex. *NIPS*, 14:221–228.

Gobel, W. and Helmchen, F. (2007). New angles on neuronal dendrites in vivo. *J Neurophysiol*, 98(6):3770–3779.

Good, I. and Gaskins, R. (1971). Nonparametric roughness penalties for probability densities. *Biometrika*, 58:255–277.

Han, X. and Boyden, E. S. (2007). Multiple-color optical activation, silencing, and desynchronization of neural activity, with single-spike temporal resolution. *PLoS ONE*, 2:e299.

Hochberg, L., Serruya, M., Friehs, G., Mukand, J., Saleh, M., Caplan, A., Branner, A., Chen, D., Penn, R., , and Donoghue, J. (2006). Neuronal ensemble control of prosthetic devices by a human with tetraplegia. *Nature*, 442:164–171.

Holy, T. (1997). The analysis of data from continuous probability distributions. *Physical Review Letters*, 79:3545–3548.

Howard, A. and Jebara, T. (2005). Square root propagation. *Columbia University Computer Science Technical Reports*, 040-05.

Humphrey, D., Schmidt, E., and Thompson, W. (1970). Predicting measures of motor performance from multiple cortical spike trains. *Science*, 170:758–762.

Huys, Q., Ahrens, M., and Paninski, L. (2006). Efficient estimation of detailed single-neuron models. *Journal of Neurophysiology*, 96:872–890.

Huys, Q. and Paninski, L. (2009). Model-based smoothing of, and parameter estimation from, noisy biophysical recordings. *PLOS Computational Biology*, In press.

Iyengar, S. (1985). Hitting lines with two-dimensional Brownian motion. *SIAM Journal on Applied Mathematics*, 45:983–989.

Izhikevich, E. (2001). Resonate-and-fire neurons. *Neural Networks*, 14:883–894.

Jolivet, R., Lewis, T., and Gerstner, W. (2004). Generalized integrate-and-fire models of neuronal activity approximate spike trains of a detailed model to a high degree of accuracy. *Journal of Neurophysiology*, 92:959–976.

Julier, S. and Uhlmann, J. (1997). A new extension of the Kalman filter to nonlinear systems. In *Int. Symp. Aerospace/Defense Sensing, Simul. and Controls, Orlando, FL*.

Jungbacker, B. and Koopman, S. (2007). Monte Carlo estimation for nonlinear non-Gaussian state space models. *Biometrika*, 94:827–839.

Kass, R., Ventura, V., and Cai, C. (2003). Statistical smoothing of neuronal data. *Network: Computation in Neural Systems*, 14:5–15.

Kerr, J. N. D., Greenberg, D., and Helmchen, F. (2005). Imaging input and output of neocortical networks in vivo. *PNAS*, 102(39):14063–14068.

Koch, C. (1999). *Biophysics of Computation*. Oxford University Press.

Koopman, S. (1993). Disturbance smoother for state space models. *Biometrika*, 80(1):117–126.

Koyama, S. and Paninski, L. (2009). Efficient computation of the maximum a posteriori path and parameter estimation in integrate-and-fire and more general state-space models. *Journal of Computational Neuroscience*, In press.

Kulkarni, J. and Paninski, L. (2007a). Common-input models for multiple neural spike-train data. *Network: Computation in Neural Systems*, 18:375–407.

Kulkarni, J. and Paninski, L. (2007b). A numerically efficient approach for constructing reach-trajectories conditioned on target. *In press, IEEE Signal Processing Magazine (special issue on brain-computer interfaces)*.

Larkum, M. E., Watanabe, S., Lasser-Ross, N., Rhodes, P., and Ross, W. N. (2008). Dendritic properties of turtle pyramidal neurons. *J Neurophysiol*, 99(2):683–694.

Lewi, J., Butera, R., and Paninski, L. (2009). Sequential optimal design of neurophysiology experiments. *Neural Computation*, 21:619–687.

Lewicki, M. (1998). A review of methods for spike sorting: the detection and classification of neural action potentials. *Network: Computation in Neural Systems*, 9:R53–R78.

Loizou, P. (1998). Mimicking the human ear: an introduction to cochlear implants. *IEEE Signal Processing Magazine*, 15:101–130.

Mainen, Z. and Sejnowski, T. (1995). Reliability of spike timing in neocortical neurons. *Science*, 268:1503–1506.

Matsuzaki, M., Ellis-Davies, G. C. R., and Kasai, H. (2008). Three-Dimensional Mapping of Unitary Synaptic Connections by Two-Photon Macro Photolysis of Caged Glutamate. *J Neurophysiol*, 99(3):1535–1544.

Minka, T. P. (1999). From hidden Markov models to linear dynamical systems. Technical report, Tech. Rep. 531, Vision and Modeling Group of Media Lab, MIT.

Mohanty, S. K., Reinscheid, R. K., Liu, X., Okamura, N., Krasieva, T. B., and Berns, M. W. (2008). In-Depth Activation of Channelrhodopsin 2-Sensitized Excitable Cells with High Spatial Resolution Using Two-Photon Excitation with a Near-Infrared Laser Microbeam. *Biophys. J.*, 95(8):3916–3926.

Murphy, G. and Rieke, F. (2006). Network variability limits stimulus-evoked spike timing precision in retinal ganglion cells. *Neuron*, 52:511–524.

Nicolelis, M., Dimitrov, D., Carmena, J., Crist, R., Lehew, G., Kralik, J., and Wise, S. (2003). Chronic, multisite, multielectrode recordings in macaque monkeys. *PNAS*, 100:11041–11046.

Nikolenko, V., Poskanzer, K., and Yuste, R. (2007). Two-photon photostimulation and imaging of neural circuits. *Nature Methods*, 4:943–950.

Nikolenko, V., Watson, B., Araya, R., Woodruff, A., Peterka, D., and Yuste, R. (2008). SLM microscopy: Scanless two-photon imaging and photostimulation using spatial light modulators. *Frontiers in Neural Circuits*, 2:5.

Nuriya, M., Jiang, J., Nemet, B., Eisenthal, K., and Yuste, R. (2006). Imaging membrane potential in dendritic spines. *PNAS*, 103:786–790.

Ohki, K., Chung, S., Kara, P., Hubener, M., Bonhoeffer, T., and Reid, R. C. (2006). Highly ordered arrangement of single neurons in orientation pinwheels. *Nature*, 442(7105):925–928.

Paninski, L. (2004). Maximum likelihood estimation of cascade point-process neural encoding models. *Network: Computation in Neural Systems*, 15:243–262.

Paninski, L. (2005). Log-concavity results on Gaussian process methods for supervised and unsupervised learning. *Advances in Neural Information Processing Systems*, 17.

Paninski, L. (2006). The most likely voltage path and large deviations approximations for integrate-and-fire neurons. *Journal of Computational Neuroscience*, 21:71–87.

Paninski, L. (2009). Inferring synaptic inputs given a noisy voltage trace via sequential Monte Carlo methods. *Journal of Computational Neuroscience*, Under review.

Paninski, L., Ahmadian, Y., Ferreira, D., Koyama, S., Rahnama, K., Vidne, M., Vogelstein, J., and Wu, W. (2009). A new look at state-space models for neural data. *Submitted*.

Paninski, L., Fellows, M., Shoham, S., Hatsopoulos, N., and Donoghue, J. (2004a). Superlinear population encoding of dynamic hand trajectory in primary motor cortex. *J. Neurosci.*, 24:8551–8561.

Paninski, L., Haith, A., and Szirtes, G. (2007). Integral equation methods for computing likelihoods and their derivatives in the stochastic integrate-and-fire model. *Journal of Computational Neuroscience*, 24:69–79.

Paninski, L., Iyengar, S., Kass, R., and Brown, E. (2008). Statistical models of spike trains. In *Stochastic Methods in Neuroscience*. Oxford University Press.

Paninski, L., Pillow, J., and Simoncelli, E. (2004b). Comparing integrate-and-fire-like models estimated using intracellular and extracellular data. *Neurocomputing*, 65:379–385.

Peña, J.-L. and Konishi, M. (2000). Cellular mechanisms for resolving phase ambiguity in the owl's inferior colliculus. *Proceedings of the National Academy of Sciences of the United States of America*, 97:11787–11792.

Petrusca, D., Grivich, M. I., Sher, A., Field, G. D., Gauthier, J. L., Greschner, M., Shlens, J., Chichilnisky, E. J., and Litke, A. M. (2007). Identification and characterization of a Y-like primate retinal ganglion cell type. *J. Neurosci.*, 27(41):11019–11027.

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

Priebe, N. and Ferster, D. (2005). Direction selectivity of excitation and inhibition in simple cells of the cat primary visual cortex. *Neuron*, 45:133–145.

Rahnama Rad, K. and Paninski, L. (2009). Efficient estimation of two-dimensional firing rate surfaces via Gaussian process methods. *Under review*.

Robinson, L., Wager, T., and Lindquist, M. (2008). Estimating distributions of onset times and durations from multi-subject fMRI studies. *Human Brain Mapping Annual Meeting*.

Roweis, S. and Ghahramani, Z. (1999). A unifying review of linear Gaussian models. *Neural Computation*, 11:305–345.

Rybicki, G. and Hummer, D. (1991). An accelerated lambda iteration method for multilevel radiative transfer, appendix b: Fast solution for the diagonal elements of the inverse of a tridiagonal matrix. *Astronomy and Astrophysics*, 245:171.

Rybicki, G. B. and Press, W. H. (1995). Class of fast methods for processing irregularly sampled or otherwise inhomogeneous one-dimensional data. *Phys. Rev. Lett.*, 74(7):1060–1063.

Santhanam, G., Ryu, S. I., Yu, B. M., Afshar, A., and Shenoy, K. V. (2006). A high-performance brain-computer interface. *Nature*, 442:195–198.

Sekirnjak, C., Hottowy, P., Sher, A., Dabrowski, W., Litke, A. M., and Chichilnisky, E. J. (2006). Electrical Stimulation of Mammalian Retinal Ganglion Cells With Multielectrode Arrays. *J Neurophysiol*, 95(6):3311–3327.

Sharia, T. (2008). Recursive parameter estimation: asymptotic expansion. *Annals of the Institute of Statistical Mathematics*, In press.

Shoham, S., Paninski, L., Fellows, M., Hatsopoulos, N., Donoghue, J., and Normann, R. (2005). Optimal decoding for a primary motor cortical brain-computer interface. *IEEE Transactions on Biomedical Engineering*, 52:1312–1322.

Shumway, R. and Stoffer, D. (2006). *Time Series Analysis and Its Applications*. Springer.

Silvapulle, M. and Sen, P. (2004). *Constrained Statistical Inference: Inequality, Order, and Shape Restrictions*. Wiley-Interscience.

Smith, A. and Brown, E. (2003). Estimating a state-space model from point process observations. *Neural Computation*, 15:965–991.

Smith, A. C., Frank, L. M., Wirth, S., Yanike, M., Hu, D., Kubota, Y., Graybiel, A. M., Suzuki, W. A., and Brown, E. N. (2004). Dynamic Analysis of Learning in Behavioral Experiments. *J. Neurosci.*, 24(2):447–461.

Smith, A. C., Stefani, M. R., Moghaddam, B., and Brown, E. N. (2005). Analysis and Design of Behavioral Experiments to Characterize Population Learning. *J Neurophysiol*, 93(3):1776–1792.

Srinivasan, L., Eden, U., Willsky, A., and Brown, E. (2006). A state-space analysis for reconstruction of goal-directed movements using neural signals. *Neural Computation*, 18:2465–2494.

Stevens, C. and Zador, A. (1998). Novel integrate-and-fire-like model of repetitive firing in cortical neurons. *Proc. 5th joint symp. neural computation, UCSD*.

Toyoizumi, T., Rahnama Rad, K., and Paninski, L. (2009). Mean-field approximations for coupled populations of generalized linear model spiking neurons with Markov refractoriness. *Neural Computation*, In press.

Truccolo, W., Eden, U., Fellows, M., Donoghue, J., and Brown, E. (2005). A point process framework for relating neural spiking activity to spiking history, neural ensemble and extrinsic covariate effects. *Journal of Neurophysiology*, 93:1074–1089.

Velliste, M., Perel, S., Spalding, M., Whitford, A., and Schwartz, A. (2008). Cortical control of a prosthetic arm for self-feeding. *Nature*, 453:1098–101.

Vidne, M., Kulkarni, J., Ahmadian, Y., Pillow, J., Shlens, J., Chichilnisky, E., Simoncelli, E., and Paninski, L. (2009). Inferring functional connectivity in an ensemble of retinal ganglion cells sharing a common input. *COSYNE*.

Vogelstein, J., Babadi, B., Watson, B., Yuste, R., and Paninski, L. (2008a). Fast nonnegative deconvolution via tridiagonal interior-point methods, applied to calcium fluorescence data. *Statistical analysis of neural data (SAND) conference*.

Vogelstein, J., Watson, B., Packer, A., Jedynak, B., Yuste, R., and Paninski, L. (2008b). Model-based optimal inference of spike times and calcium dynamics given noisy and intermittent calcium-fluorescence imaging. *Biophysical Journal*, In press; http://www.stat.columbia.edu/∼liam/research/abstracts/vogelstein-bj08-abs.html.

Wang, X., Wei, Y., Vaingankar, V., Wang, Q., Koepsell, K., Sommer, F., and Hirsch, J. (2007). Feedforward excitation and inhibition evoke dual modes of firing in the cat's visual thalamus during naturalistic viewing. *Neuron*, 55:465–478.

Wehr, M. and Zador, A. (2003). Balanced inhibition underlies tuning and sharpens spike timing in auditory cortex. *Nature*, 426:442–446.

Weiland, J., Liu, W., and Humayun, M. (2005). Retinal prosthesis. *Annual Review of Biomedical Engineering*, 7:361–401.

West, M. and Harrison, P. (1997). *Bayesian Forecasting and Dynamic Models*. Springer.

Wirth, S., Yanike, M., Frank, L., Smith, A., Brown, E., and Suzuki, W. (2003). Single neurons in the monkey hippocampus and learning of new associations. *Science*, 300:1578–1581.

Wu, W., Black, M. J., Mumford, D., Gao, Y., Bienenstock, E., and Donoghue, J. (2004). Modeling and decoding motor cortical activity using a switching Kalman filter. *IEEE Transactions on Biomedical Engineering*, 51:933–942.

Wu, W., Gao, Y., Bienenstock, E., Donoghue, J. P., and Black, M. J. (2006). Bayesian population coding of motor cortical activity using a Kalman filter. *Neural Computation*, 18:80–118.

Xie, R., Gittelman, J. X., and Pollak, G. D. (2007). Rethinking tuning: In vivo whole-cell recordings of the inferior colliculus in awake bats. *J. Neurosci.*, 27(35):9469–9481.