J Comput Neurosci (2010) 29:107–126
DOI 10.1007/s10827-009-0179-x

A new look at state-space models for neural data

Liam Paninski · Yashar Ahmadian · Daniel Gil Ferreira · Shinsuke Koyama · Kamiar Rahnama Rad · Michael Vidne · Joshua Vogelstein · Wei Wu

Received: 22 December 2008 / Revised: 6 July 2009 / Accepted: 16 July 2009 / Published online: 1 August 2009

Abstract State space methods have proven indispensable in neural data analysis. However, common methods for performing inference in state-space models with non-Gaussian observations rely on certain approximations which are not always accurate. Here we review direct optimization methods that avoid these approximations, but that nonetheless retain the computational efficiency of the approximate methods. We discuss a variety of examples, applying these direct optimization techniques to problems in spike train smoothing, stimulus decoding, parameter estimation, and inference of synaptic properties. Along the way, we point out connections to some related standard statistical methods, including spline smoothing and isotonic regression. Finally, we note that the computational methods reviewed here do not in fact depend on the state-space setting at all; instead, the key property we are exploiting involves the bandedness of certain matrices. We close by discussing some applications of this more general point of view, including Markov chain Monte Carlo methods for neural decoding and efficient estimation of spatially-varying firing rates.

Keywords Neural coding · State-space models · Hidden Markov model · Tridiagonal matrix

Action Editor: Israel Nelken

L. Paninski (✉) · Y. Ahmadian · D. G. Ferreira · K. Rahnama Rad · M. Vidne
Department of Statistics and Center for Theoretical Neuroscience, Columbia University, New York, NY, USA
e-mail: liam@stat.columbia.edu
URL: http://www.stat.columbia.edu/∼liam

S. Koyama
Department of Statistics, Carnegie Mellon University, Pittsburgh, PA, USA

J. Vogelstein
Department of Neuroscience, Johns Hopkins University, Baltimore, MD, USA

W. Wu
Department of Statistics, Florida State University, Tallahassee, FL, USA

1 Introduction; forward-backward methods for inference in state-space models

A wide variety of neuroscientific data analysis problems may be attacked fruitfully within the framework of hidden Markov ("state-space") models. The basic idea is that the underlying system may be described as a stochastic dynamical process: a (possibly multidimensional) state variable q_t evolves through time according to some Markovian dynamics $p(q_t|q_{t-1}, \theta)$, as specified by a few model parameters θ. Now in many situations we do not observe the state variable q_t directly (this Markovian variable is "hidden"); instead, our observations y_t are a noisy, subsampled version of q_t, summarized by an observation distribution $p(y_t|q_t)$.

Methods for performing optimal inference and estimation in these hidden Markov models are very well-developed in the statistics and engineering literature (Rabiner 1989; Durbin and Koopman 2001; Doucet et al. 2001). For example, to compute the conditional distribution $p(q_t|Y_{1:T})$ of the state variable q_t given all the observed data on the time interval $(0, T]$, we need only apply two straightforward recursions: a *forward* recursion that computes the conditional dis-

tribution of q_t given only the observed data up to time t,

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

$$\text{for } t = 1, 2, \ldots, T \quad (1)$$

and then a *backward* recursion that computes the desired distribution $p(q_t|Y_{1:T})$,

$$p(q_t|Y_{1:T}) = p(q_t|Y_{1:t}) \int \frac{p(q_{t+1}|Y_{1:T}) p(q_{t+1}|q_t)}{\int p(q_{t+1}|q_t) p(q_t|Y_{1:t}) dq_t} dq_{t+1},$$

$$\text{for } t = T-1, T-2, \ldots, 1, 0. \quad (2)$$

Each of these recursions may be derived easily from the Markov structure of the state-space model. In the classical settings, where the state variable q is discrete (Rabiner 1989; Gat et al. 1997; Hawkes 2004; Jones et al. 2007; Kemere et al. 2008; Herbst et al. 2008; Escola and Paninski 2009), or the dynamics $p(q_t|q_{t-1})$ and observations $p(y_t|q_t)$ are linear and Gaussian, these recursions may be computed exactly and efficiently: note that a full forward-backward sweep requires computation time which scales just linearly in the data length T, and is therefore quite tractable even for large T. In the linear-Gaussian case, this forward-backward recursion is known as the Kalman filter-smoother (Roweis and Ghahramani 1999; Durbin and Koopman 2001; Penny et al. 2005; Shumway and Stoffer 2006).

Unfortunately, the integrals in Eqs. (1) and (2) are not analytically tractable in general; in particular, for neural applications we are interested in cases where the observations y_t are point processes (e.g., spike trains, or behavioral event times), and in this case the recursions must be solved approximately. One straightforward idea is to approximate the conditional distributions appearing in (1) and (2) as Gaussian; since we can compute Gaussian integrals analytically (as in the Kalman filter), this simple approximation provides a computationally tractable, natural extension of the Kalman filter to non-Gaussian observations. Many versions of this recursive Gaussian approximation idea (with varying degrees of accuracy versus computational expediency) have been introduced in the statistics and neuroscience literature (Fahrmeir and Kaufmann 1991; Fahrmeir and Tutz 1994; Bell 1994; Kitagawa and Gersch 1996; West and Harrison 1997; Julier and Uhlmann 1997; Brown et al. 1998; Smith and Brown 2003; Ypma and Heskes 2003; Eden et al. 2004; Yu et al. 2006).

These methods have proven extremely useful in a wide variety of neural applications. Recursive estimation methods are especially critical in online applications, where estimates must be updated in real time as new information is observed. For example, state-space techniques achieve state-of-the-art performance decoding multineuronal spike train data from motor cortex (Wu et al. 2006; Truccolo et al. 2005; Wu et al. 2009) and parietal cortex (Yu et al. 2006; Kemere et al. 2008), and these methods therefore hold great promise for the design of motor neuroprosthetic devices (Donoghue 2002). In this setting, the hidden variable q_t corresponds to the desired position of the subject's hand, or a cursor on a computer screen, at time t; y_t is the vector of observed spikes at time t, binned at some predetermined temporal resolution; the conditional probability $p(y_t|q_t)$ is given by an "encoding" model that describes how the position information q_t is represented in the spike trains y_t; and $p(q_t|Y_{1:t+s})$ is the desired fixed-lag decoding distribution, summarizing our knowledge about the current position q_t given all of the observed spike train data Y from time 1 up to $t+s$, where s is a short allowed time lag (on the order of 100 ms or so in motor prosthetic applications). In this setting, the conditional expectation $E(q_t|Y_{1:t+s})$ is typically used as the optimal (minimum mean-square) estimator for q_t, while the posterior covariance $Cov(q_t|Y_{1:t+s})$ quantifies our uncertainty about the position q_t, given the observed data; both of these quantities are computed most efficiently using the forward-backward recursions (1–2). These forward-backward methods can also easily incorporate target or endpoint goal information in these online decoding tasks (Srinivasan et al. 2006; Yu et al. 2007; Kulkarni and Paninski 2008; Wu et al. 2009).

State-space models have also been applied successfully to track nonstationary neuron tuning properties (Brown et al. 2001; Frank et al. 2002; Eden et al. 2004; Czanner et al. 2008; Rahnama et al. 2009). In this case, the hidden state variable q_t represents a parameter vector which determines the neuron's stimulus-response function. Lewi et al. (2009) discusses an application of these recursive methods to perform optimal online experimental design — i.e., to choose the stimulus at time t which will give us as much information as possible about the observed neuron's response properties, given all the observed stimulus-response data from time 1 to t.

A number of offline applications have appeared as well: state-space methods have been applied to perform optimal decoding of rat position given multiple hippocampal spike trains (Brown et al. 1998; Zhang et al. 1998; Eden et al. 2004), and to model behavioral learning experiments (Smith and Brown 2003; Smith et al. 2004, 2005; Suzuki and Brown 2005); in the latter case, q_t represents the subject's certainty about

the behavioral task, which is not directly observable and which changes systematically over the course of the experiment. In addition, we should note that the forward-backward idea is of fundamental importance in the setting of sequential Monte Carlo ("particle-filtering") methods (Doucet et al. 2001; Brockwell et al. 2004; Kelly and Lee 2004; Godsill et al. 2004; Shoham et al. 2005; Ergun et al. 2007; Vogelstein et al. 2009; Huys and Paninski 2009), though we will not focus on these applications here.

However, the forward-backward approach is not always directly applicable. 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 may break down.[1] In this paper, we will review more general direct optimization methods for performing inference in state-space models. We discuss this approach in Section 2 below. This direct optimization approach also leads to more efficient methods for estimating the model parameters θ (Section 3). Finally, the state-space model turns out to be a special case of a richer, more general framework involving banded matrix computations, as we discuss at more length in Section 4.

2 A direct optimization approach for computing the maximum a posteriori path in state-space models

2.1 A direct optimization interpretation of the classical Kalman filter

We begin by taking another look at the classical Kalman filter-smoother (Durbin and Koopman 2001; Wu et al. 2006; Shumway and Stoffer 2006). The primary goal of the smoother is to compute the conditional expectation $E(Q|Y)$ of the hidden state path Q given the observations Y. (Throughout this paper, we will use Q and Y to denote the full collection of the hidden state variables $\{q_t\}$ and observations $\{y_t\}$, respectively.) Due to the linear-Gaussian structure of the Kalman model, (Q, Y) forms a jointly Gaussian random vector, and therefore $p(Q|Y)$ is itself Gaussian. Since the mean and mode of a Gaussian distribution coincide, this implies that $E(Q|Y)$ is equal to the maximum a posteriori (MAP) solution, the maximizer of the posterior $p(Q|Y)$. If we write out the linear-Gaussian Kalman model more explicitly,

$$q_t = Aq_{t-1} + \epsilon_t,\ \epsilon_t \sim \mathcal{N}(0, C_q);\ q_1 \sim \mathcal{N}(\mu_1, C_1)$$

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

(where $\mathcal{N}(0, C)$ denotes the Gaussian density with mean 0 and covariance C), we can gain some insight into the the analytical form of this maximizer:

$$\begin{aligned}
E(Q|Y) &= \arg\max_Q p(Q|Y) \\
&= \arg\max_Q \log p(Q, Y) \\
&= \arg\max_Q \Bigg(\log p(q_1) + \sum_{t=2}^{T} \log p(q_t|q_{t-1}) \\
&\qquad + \sum_{t=1}^{T} \log p(y_t|q_t) \Bigg) \\
&= \arg\max_Q \Bigg[-\frac{1}{2} \Bigg((q_1-\mu_1)^T C_1^{-1} (q_1-\mu_1) \\
&\qquad + \sum_{t=2}^{T} (q_t - Aq_{t-1})^T C_q^{-1} (q_t - Aq_{t-1}) \\
&\qquad + \sum_{t=1}^{T} (y_t - Bq_t)^T C_y^{-1} (y_t - Bq_t) \Bigg) \Bigg].
\end{aligned} \tag{3}$$

The right-hand-side here is a simple quadratic function of Q (as expected, since $p(Q|Y)$ is Gaussian, i.e., $\log p(Q|Y)$ is quadratic), and therefore $E(Q|Y)$ may be computed by solving an unconstrained quadratic program in Q; we thus obtain

$$\begin{aligned}
\hat{Q} &= \arg\max_Q \log p(Q|Y) = \arg\max_Q \left[\frac{1}{2} Q^T H Q + \nabla^T Q \right] \\
&= -H^{-1}\nabla,
\end{aligned}$$

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

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

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

The next key point to note is that the Hessian matrix H is block-tridiagonal, since $\log p(Q|Y)$ is a sum of sim-

[1] It is worth noting that other more sophisticated methods such as expectation propagation (Minka 2001; Ypma and Heskes 2003; Yu et al. 2006, 2007; Koyama and Paninski 2009) may be better-equipped to handle these strongly non-Gaussian observation densities $p(y_t|q_t)$ (and are, in turn, closely related to the optimization-based methods that are the focus of this paper); however, due to space constraints, we will not discuss these methods at length here.

ple 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

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

where

$$D_i = \frac{\partial^2}{\partial q_i^2} \log p(y_i|q_i) + \frac{\partial^2}{\partial q_i^2} \log p(q_i|q_{i-1}) + \frac{\partial^2}{\partial q_i^2} \log p(q_{i+1}|q_i), \tag{5}$$

and

$$R_{i,i+1} = \frac{\partial^2}{\partial q_i \partial q_{i+1}} \log p(q_{i+1}|q_i) \tag{6}$$

for $1 < i < N$. These quantities may be computed as simple functions of the Kalman model parameters; for example, $R_{i,i+1} = C_q^{-1} A$.

This block-tridiagonal form of H implies that the linear equation $\hat{Q} = H^{-1}\nabla$ may be solved in $O(T)$ time (e.g., by block-Gaussian elimination (Press et al. 1992); note that we never need to compute H^{-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 H is sparse and banded, the standard Matlab backslash command $\hat{Q} = H\backslash\nabla$ 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 (see, e.g., Shumway and Stoffer 2006 for details). These quantities may easily be computed in $O(T)$ time in the matrix formulation. For example, since the matrix H represents the inverse posterior covariance matrix of our Gaussian vector Q given Y, $Cov(q_t|Y)$ is given by the (t,t)-th block of H^{-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 H^{-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).

2.2 Extending the direct optimization method to non-Gaussian models

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 paper 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 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.

To apply Newton's method here, we simply iteratively solve the linear equation[2]

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

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

[2] In practice, the simple Newton iteration does not always increase the objective $\log p(Q|Y)$; we have found the standard remedy for this instability (perform a simple backtracking line-search along the Newton direction $\hat{Q}^{(i)} - \delta^{(i)} H^{-1}\nabla$ to determine a suitable stepsize $\delta^{(i)} \leq 1$) to be quite effective here.

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 here. 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 Section 1. 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 (though apparently less so in the neuroscience literature). See Fahrmeir and Kaufmann (1991), Fahrmeir and Tutz (1994), Bell (1994), Davis and Rodriguez-Yam (2005), Jungbacker and Koopman (2007) for further discussion from a statistical point of view, and Koyama and Paninski (2009) for applications to the integrate-and-fire model for spiking data. In addition, Yu et al. (2007) previously applied a related direct optimization approach in the context of neural decoding (though note that the conjugate gradients approach utilized there requires $O(T^3)$ time if the banded structure of the Hessian is not exploited).

2.3 Example: inferring common input effects in multineuronal spike train recordings

Recent developments in multi-electrode recording technology (Nicolelis et al. 2003; Litke et al. 2004) and fluorescence microscopy (Cossart et al. 2003; Ohki et al. 2005; Nikolenko et al. 2008) enable the simultaneous measurement of the spiking activity of many neurons. Analysis of such multineuronal data is one of the key challenges in computational neuroscience today (Brown et al. 2004), and a variety of models for these data have been introduced (Chornoboy et al. 1988; Utikal 1997; Martignon et al. 2000; Iyengar 2001; Schnitzer and Meister 2003; Paninski et al. 2004; Truccolo et al. 2005; Nykamp 2005; Schneidman et al. 2006; Shlens et al. 2006; Pillow et al. 2008; Shlens et al. 2009). Most of these models include stimulus-dependence terms and "direct coupling" terms representing the influence that the activity of an observed cell might have on the other recorded neurons. These coupling terms are often interpreted in terms of "functional connectivity" between the observed neurons; the major question now is how accurately this inferred functional connectivity actually reflects the true underlying anatomical connectivity in the circuit.

Fewer models, however, have attempted to include the effects of the population of neurons which are not directly observed during the experiment (Nykamp 2005, 2007; Kulkarni and Paninski 2007). Since we can directly record from only a small fraction of neurons in any physiological preparation, such unmeasured neurons might have a large collective impact on the dynamics and coding properties of the observed neural population, and may bias the inferred functional connectivity away from the true anatomical connectivity, complicating the interpretation of these multineuronal analyses. For example, while Pillow et al. (2008) found that neighboring parasol retinal ganglion cells (RGCs) in the macaque are functionally coupled — indeed, incorporating this functional connectivity in an optimal Bayesian decoder significantly amplifies the information we can extract about the visual stimulus from the observed spiking activity of large ensembles of RGCs — Khuc-Trong and Rieke (2008) recently demonstrated, via simultaneous pairwise intracellular recordings, that RGCs receive a significant amount of strongly correlated common input, with weak direct anatomical coupling between RGCs. Thus the strong functional connectivity observed in this circuit is in fact largely driven by common input, not direct anatomical connectivity.

Therefore it is natural to ask if it is possible to correctly infer the degree of common input versus direct coupling in partially-observed neuronal circuits, given only multineuronal spike train data (i.e., we do not want to rely on multiple simultaneous intracellular recordings, which are orders of magnitude more difficult to obtain than extracellular recordings). To this end, Kulkarni and Paninski (2007) introduced a state-space model in which the firing rates depend not only on the stimulus history and the spiking history of the observed neurons but also on common input effects (Fig. 1). In this model, the conditional firing intensity, $\lambda_i(t)$, of the i-th observed neuron is:

$$\lambda_i(t) = \exp\left(\mathbf{k}_i \cdot \mathbf{x}(t) + \mathbf{h}_i \cdot \mathbf{y}_i(t) + \left(\sum_{i \neq j} \mathbf{l}_{ij} \cdot \mathbf{y}_j(t)\right) + \mu_i + q_i(t)\right), \tag{7}$$

where $\mathbf{x}$ is the spatiotemporal visual stimulus, $\mathbf{y}_i$ is cell i's own spike-train history, μ_i is the cell's baseline log-firing rate, $\mathbf{y}_j$ are the spike-train histories of other cells

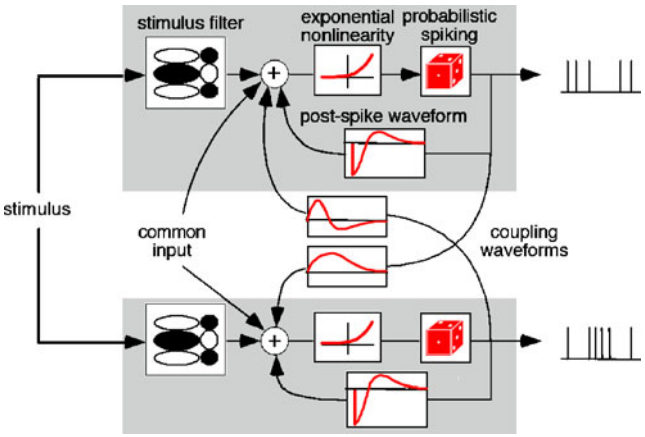

Fig. 1 Schematic illustration of the common-input model described by Eq. (7); adapted from Kulkarni and Paninski (2007)

at time t, $\mathbf{k}_i$ is the cell's spatiotemporal stimulus filter, $\mathbf{h}_i$ is the post-spike temporal filter accounting for past spike dependencies within cell i, and $\mathbf{l}_{ij}$ are direct coupling temporal filters, which capture the dependence of cell i's activity on the recent spiking of other cells j. The term $q_i(t)$, the hidden common input at time t, is modeled as a Gauss-Markov autoregressive process, with some correlation between different cells i which we must infer from the data. In addition, we enforce a nonzero delay in the direct coupling terms, so that the effects of a spike in one neuron on other neurons are temporally strictly causal.

In statistical language, this common-input model is a multivariate version of a Cox process, also known as a doubly-stochastic point process (Cox 1955; Snyder and Miller 1991; Moeller et al. 1998); the state-space models applied in Smith and Brown (2003), Truccolo et al. (2005), Czanner et al. (2008) are mathematically very similar. See also Yu et al. (2006) for discussion of a related model in the context of motor planning and intention.

As an example of the direct optimization methods developed in the preceding subsection, we reanalyzed the data from Pillow et al. (2008) with this common input model (Vidne et al. 2009). We estimated the model parameters $\theta = (\mathbf{k}_i, \mathbf{h}_i, \mathbf{l}_{ij}, \mu_i)$ from the spiking data by maximum marginal likelihood, as described in Koyama and Paninski (2009) (see also Section 3 below, for a brief summary); the correlation time of Q was set to ~ 5 ms, to be consistent with the results of Khuc-Trong and Rieke (2008). We found that this common-input model explained the observed cross-correlations quite well (data not shown), and the inferred direct-coupling weights were set to be relatively small (Fig. 2); in fact, the quality of the fits in our preliminary experiments is indistinguishable from those described in Pillow et al. (2008), where a model with strong direct-coupling terms and no common-input effects was used.

Given the estimated model parameters θ, we used the direct optimization method to estimate the subthreshold common input effects $q(t)$, on a single-trial basis (Fig. 2). The observation likelihood $p(y_t|q_t)$ here was given by the standard point-process likelihood (Snyder and Miller 1991):

$$\log p(Y|Q) = \sum_{it} y_{it} \log \lambda_i(t) - \lambda_i(t)dt, \tag{8}$$

where y_{it} denotes the number of spikes observed in time bin t from neuron i; dt denotes the temporal bin-width. We see in Fig. 2 that the inferred common input effect is strong relative to the direct coupling effects, in agreement with the intracellular recordings described in Khuc-Trong and Rieke (2008). We are currently working to quantify these common input effects $q_i(t)$ inferred from the full observed RGC population, rather than just the pairwise analysis shown here, in order to investigate the relevance of this strong common input effect on the coding and correlation properties of the full network of parasol cells. See also Wu et al. (2009) for applications of similar common-input state-space models to improve decoding of population neural activity in motor cortex.

2.4 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; as emphasized in the introduction, 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; Koyama and Paninski 2009). The idea is to replace the constrained concave problem

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

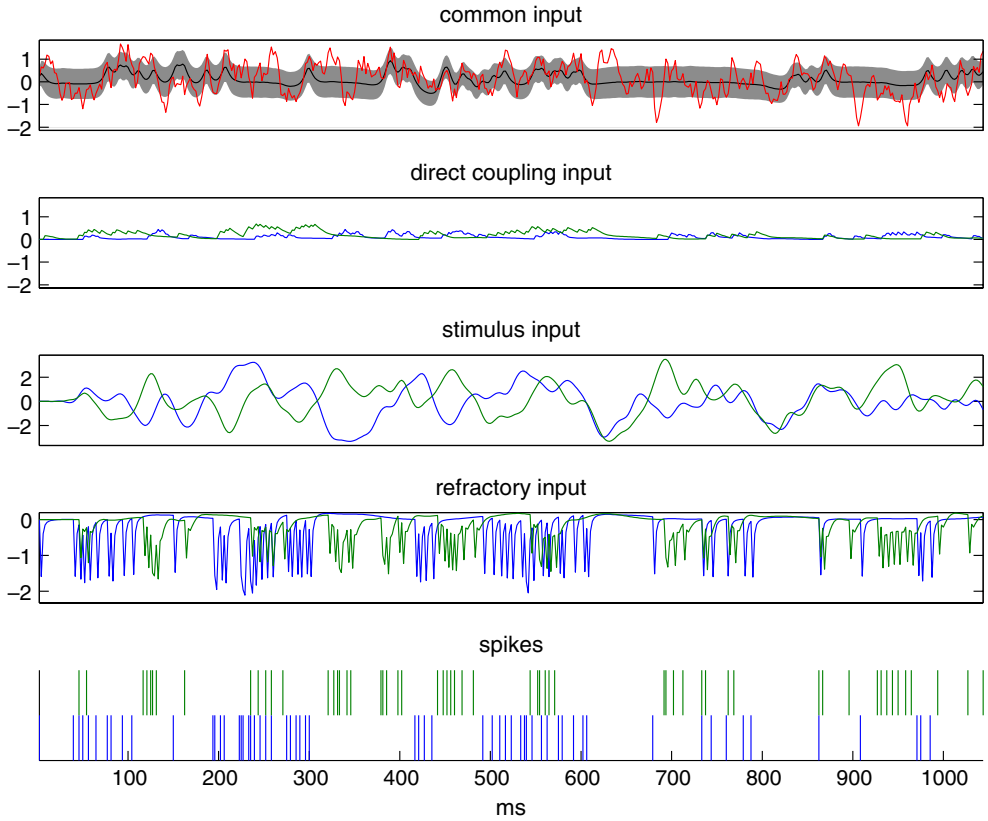

Fig. 2 Single-trial inference of the relative contribution of common, stimulus, direct coupling, and self inputs in a pair of retinal ganglion ON cells (Vidne et al. 2009); data from (Pillow et al. 2008). *Top panel*: Inferred linear common input, $\hat{Q}$: *red trace* shows a sample from the posterior distribution $p(Q|Y)$, *black trace* shows the conditional expectation $E(Q|Y)$, and *shaded region* indicates ± 1 posterior standard deviation about $E(Q|Y)$, computed from the diagonal of the inverse log-posterior Hessian H. 2^{nd} *panel*: Direct coupling input from the other cell, $\mathbf{l}_j \cdot \mathbf{y}_j$. (The *first two panels* are plotted on the same scale to facilitate comparison of the magnitudes of these effects.) *Blue trace* indicates cell 1; *green* indicates cell 2. 3^{rd} *panel*: The stimulus input, $\mathbf{k} \cdot \mathbf{x}$. 4^{th} *panel*: Refractory input, $\mathbf{h_i} \cdot \mathbf{y_i}$. Note that this term is strong but quite short-lived following each spike. All units are in log-firing rate, as in Eq. (7). *Bottom*: Observed paired spike trains Y on this single trial. Note the large magnitude of the estimated common input term $\hat{q}(t)$, relative to the direct coupling contribution $\mathbf{l}_j \cdot \mathbf{y}_j$

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 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 $\hat{Q}$. Note that the approximation $\arg\max_Q p(Q|Y) \approx E(Q|Y)$ will typically *not* hold in this constrained case, since the mean of a truncated Gaussian distribution will typically not coincide with the mode (unless the mode is sufficiently far from the nonnegativity constraint).

We give a few applications of this barrier approach below. See also Koyama and Paninski (2009) for a detailed discussion of an application to the integrate-and-fire model, and Vogelstein et al. (2008) for applications to the problem of efficiently deconvolving slow, noisy calcium fluorescence traces to obtain nonnegative estimates of spiking times. In addition, see Cunningham et al. (2008) for an application of the log-barrier method to infer firing rates given point process observations in a closely-related Gaussian process model (Rasmussen

and Williams 2006); these authors considered a slightly more general class of covariance functions on the latent stochastic process q_t, but the computation time of the resulting method scales superlinearly[3] with T.

2.4.1 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 $f(q_t)$ to be very large when spikes are observed (by making q_t very large).

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

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

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 2007, 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

[3] More precisely, Cunningham et al. (2008) introduce a clever iterative conjugate-gradient (CG) method to compute the MAP path in their model; this method requires $O(T \log T)$ time per CG step, with the number of CG steps increasing as a function of the number of observed spikes. (Note, in comparison, that the computation times of the state-space methods reviewed in the current work are insensitive to the number of observed spikes.)

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.

2.4.2 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 and Konishi 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 postsynaptic voltage and conductance:

$$\begin{aligned} V_{t+dt} &= V_t + dt\left[g^L(V^L - V_t) + g_t^E(V^E - V_t)\right. \\ &\quad \left. + g_t^I(V^I - V_t)\right] + \epsilon_t \\ g_{t+dt}^E &= g_t^E - \frac{g_t^E}{\tau_E}dt + N_t^E \\ g_{t+dt}^I &= g_t^I - \frac{g_t^I}{\tau_I}dt + N_t^I. \end{aligned} \tag{9}$$

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

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

in the case of white Gaussian current noise ϵ_t with variance $\sigma^2 dt$, for example,[4] we have

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

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. 3. 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 Eq. (9), and then used the $O(T)$ log-barrier method to simultaneously infer the presynaptic conductances

[4] The approach here can be easily generalized to the case that the input noise has a nonzero correlation timescale. For example, if the noise can be modeled as an autoregressive process of order p instead of the white noise process described here, then we simply include the unobserved p-dimensional Markov noise process in our state variable (i.e., our Markov state variable q_t will now have dimension $p + 2$ instead of 2), and then apply the $O(T)$ log-barrier method to this augmented state space.

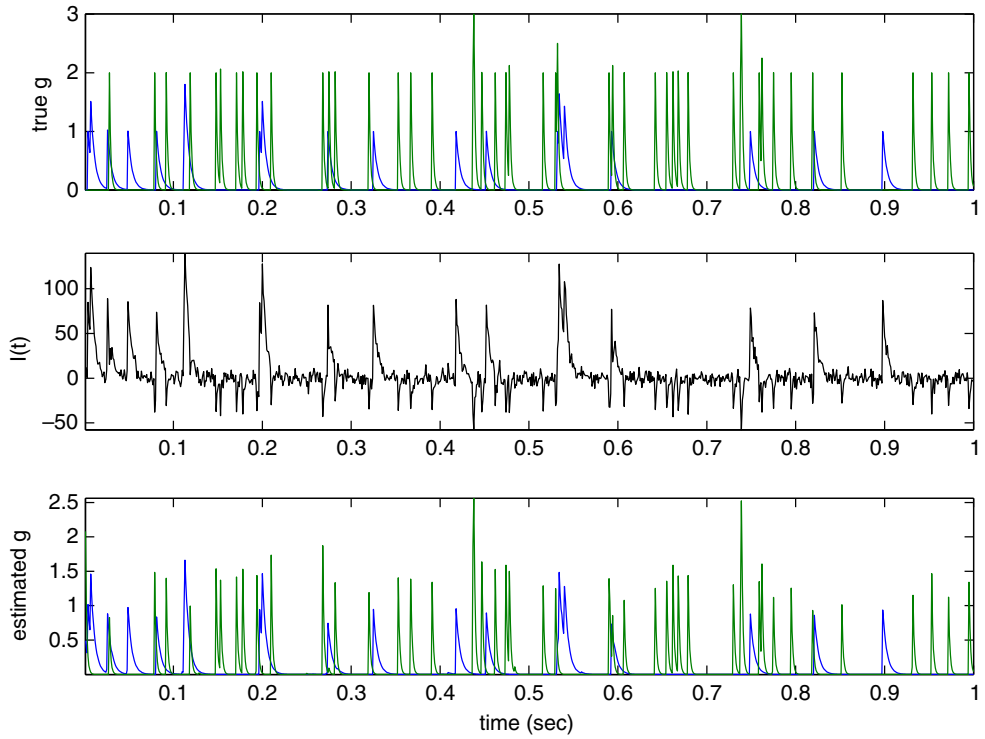

Fig. 3 Inferring presynaptic inputs given simulated postsynaptic voltage recordings. *Top*: true simulated conductance input (*green* indicates inhibitory conductance; *blue* excitatory). *Middle*: observed noisy current trace from which we will attempt to infer the input conductance. *Bottom*: Conductance 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

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 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 directly from the observed data.

3 Parameter estimation

In the previous sections we have discussed the inference of the hidden state path Q in the case that the system parameters are known. However, in most applications, we need to estimate the system parameters as well. As discussed in Kass et al. (2005), standard modern methods for parameter estimation are based on the likelihood $p(Y|\theta)$ of the observed data Y given the parameters θ. In the state-space setting, we need to compute the marginal likelihood by integrating out the hidden state path Q:[5]

$$p(Y|\theta) = \int p(Q, Y|\theta) dQ = \int p(Y|Q, \theta) p(Q|\theta) dQ. \quad (10)$$

[5] In some cases, Q may be observed directly on some subset of training data. If this is the case (i.e., direct observations of q_t are available together with the observed data Y), then the estimation problem simplifies drastically, since we can often fit the models $p(y_t|q_t, \theta)$ and $p(q_t|q_{t-1}, \theta)$ directly without making use of the more involved latent-variable methods discussed in this section.

This marginal likelihood is a unimodal function of the parameters θ in many important cases (Paninski 2005), making maximum likelihood estimation feasible in principle. However, this high-dimensional integral can not be computed exactly in general, and so many approximate techniques have been developed, including Monte Carlo methods (Robert and Casella 2005; Davis and Rodriguez-Yam 2005; Jungbacker and Koopman 2007; Ahmadian et al. 2009a) and expectation propagation (Minka 2001; Ypma and Heskes 2003; Yu et al. 2006; Koyama and Paninski 2009).

In the neural applications reviewed in Section 1, the most common method for maximizing the loglikelihood is the approximate Expectation-Maximization (EM) algorithm introduced in Smith and Brown (2003), in which the required expectations are approximated using the recursive Gaussian forward-backward method. This EM algorithm can be readily modified to optimize an approximate log-posterior $p(\theta|Y)$, if a useful prior distribution $p(\theta)$ is available (Kulkarni and Paninski 2007). While the EM algorithm does not require us to compute the likelihood $p(Y|\theta)$ explicitly, we may read this likelihood off of the final forward density approximation $p(q_T, y_{1:T})$ by simply marginalizing out the final state variable q_T. All of these computations are recursive, and may therefore be computed in $O(T)$ time.

The direct global optimization approach discussed in the preceding section suggests a slightly different strategy. Instead of making T local Gaussian approximations recursively—once for each forward density $p(q_t, y_{1:t})$—we make a single global Gaussian approximation for the full joint posterior:

$$\log p(Q|Y,\theta) \approx -\frac{T}{2}\log(2\pi) - \frac{1}{2}\log|-H_\theta| + \frac{1}{2}(Q-\hat{Q}_\theta)^T H_\theta (Q-\hat{Q}_\theta), \quad (11)$$

where the Hessian H_θ is defined as

$$H_\theta = \nabla\nabla_Q\left(\log p(Q|\theta) + \log p(Y|Q,\theta)\right)\Big|_{Q=\hat{Q}_\theta};$$

note the implicit dependence on θ through

$$\hat{Q}_\theta \equiv \arg\max_Q \left[\log p(Q|\theta) + \log p(Y|Q,\theta)\right].$$

Equation (11) corresponds to nothing more than a second-order Taylor expansion of $\log p(Q|Y,\theta)$ about the optimizer $\hat{Q}_\theta$.

Plugging this Gaussian approximation into Eq. (10), we obtain the standard "Laplace" approximation (Kass and Raftery 1995; Davis and Rodriguez-Yam 2005; Yu et al. 2007) for the marginal likelihood,

$$\log p(Y|\theta) \approx \log p(Y|\hat{Q}_\theta,\theta) + \log p(\hat{Q}_\theta|\theta) - \frac{1}{2}\log|-H_\theta| + const. \quad (12)$$

Clearly the first two terms here can be computed in $O(T)$, since we have already demonstrated that we can obtain $\hat{Q}_\theta$ in $O(T)$ time, and evaluating $\log p(Y|Q,\theta) + \log p(Q|\theta)$ at $Q = \hat{Q}_\theta$ is relatively easy. We may also compute $\log|-H_\theta|$ stably and in $O(T)$ time, via the Cholesky decomposition for banded matrices (Davis and Rodriguez-Yam 2005; Koyama and Paninski 2009). In fact, we can go further: Koyama and Paninski (2009) show how to compute the gradient of Eq. (12) with respect to θ in $O(T)$ time, which makes direct optimization of this approximate likelihood feasible via conjugate gradient methods.[6]

It is natural to compare this direct optimization approach to the EM method; again, see Koyama and Paninski (2009) for details on the connections between these two approaches. It is well-known that EM can converge slowly, especially in cases in which the so-called "ratio of missing information" is large; see Dempster et al. (1977); Meng and Rubin (1991); Salakhutdinov et al. (2003); Olsson et al. (2007) for details. In practice, we have found that direct gradient ascent of expression (12) is significantly more efficient than the EM approach in the models discussed in this paper; for example, we used the direct approach to perform parameter estimation in the retinal example discussed above in Section 2.3. One important advantage of the direct ascent approach is that in some special cases, the optimization of (12) can be performed in a single step (as opposed to multiple EM steps). We illustrate this idea with a simple example below.

3.1 Example: detecting the location of a synapse given noisy, intermittent voltage observations

Imagine we make noisy observations from a dendritic tree (for example, via voltage-sensitive imaging methods (Djurisic et al. 2008)) which is receiving synaptic inputs from another neuron. We do not know the strength

[6] It is worth mentioning the work of Cunningham et al. (2008) again here; these authors introduced conjugate gradient methods for optimizing the marginal likelihood in their model. However, their methods require computation time scaling superlinearly with the number of observed spikes (and therefore superlinearly with T, assuming that the number of observed spikes is roughly proportional to T).

or location of these inputs, but we do have complete access to the spike times of the presynaptic neuron (for example, we may be stimulating the presynaptic neuron electrically or via photo-uncaging of glutamate near the presynaptic cell (Araya et al. 2006; Nikolenko et al. 2008)). How can we determine if there is a synapse between the two cells, and if so, how strong the synapse is and where it is located on the dendritic tree?

To model this experiment, we assume that the neuron is in a stable, subthreshold regime, i.e., the spatiotemporal voltage dynamics are adequately approximated by the linear cable equation

$$\vec{V}_{t+dt} = \vec{V}_t + dt\big(A\vec{V}_t + \theta U_t\big) + \epsilon_t,\ \epsilon_t \sim \mathcal{N}(0, C_V dt). \tag{13}$$

Here the dynamics matrix A includes both leak terms and intercompartmental coupling effects: for example, in the special case of a linear dendrite segment with N compartments, with constant leak conductance g and intercompartmental conductance a, A is given by the tridiagonal matrix

$$A = -gI + aD^2,$$

with D^2 denoting the second-difference operator. For simplicity, we assume that U_t is a known signal:

$$U_t = h(t) * \sum_i \delta(t - t_i);$$

$h(t)$ is a known synaptic post-synaptic (PSC) current shape (e.g., an α-function (Koch 1999)), $*$ denotes convolution, and $\sum_i \delta(t - t_i)$ denotes the presynaptic spike train. The weight vector θ is the unknown parameter we want to infer: θ_i is the synaptic weight at the i-th dendritic compartment. Thus, to summarize, we have assumed that each synapse fires deterministically, with a known PSC shape (only the magnitude is unknown) at a known latency, with no synaptic depression or facilitation. (All of these assumptions may be relaxed significantly (Paninski and Ferreira 2008).)

Now we would like to estimate the synaptic weight vector θ, given U and noisy observations of the spatiotemporal voltage V. V is not observed directly here, and therefore plays the role of our hidden variable Q. For concreteness, we further assume that the observations Y are approximately linear and Gaussian:

$$y_t = B\vec{V}_t + \eta_t,\ \eta_t \sim \mathcal{N}(0, C_y).$$

In this case the voltage V and observed data Y are jointly Gaussian given U and the parameter θ, and furthermore V depends on θ linearly, so estimating θ can be seen as a rather standard linear-Gaussian estimation problem. There are many ways to solve this problem: we could, for example, use EM to alternatively estimate V given θ and Y, then estimate θ given our estimated V, alternating these maximizations until convergence. However, a more efficient approach (and one which generalizes nicely to the nonlinear case (Koyama and Paninski 2009)) is to optimize Eq. (12) directly. Note that this Laplace approximation is in fact exact in this case, since the posterior $p(Y|\theta)$ is Gaussian. Furthermore, the log-determinant term $\log|-H_\theta|$ is constant in θ (since the Hessian is constant in this Gaussian model), and so we can drop this term from the optimization. Thus we are left with

$$\begin{aligned}
&\arg\max_\theta \log p(Y|\theta) \\
&\quad = \arg\max_\theta \left\{\log p(Y|\hat{V}_\theta, \theta) + \log p(\hat{V}_\theta|\theta)\right\} \\
&\quad = \arg\max_\theta \left\{\log p(Y|\hat{V}_\theta) + \log p(\hat{V}_\theta|\theta)\right\} \\
&\quad = \arg\max_\theta \max_V \left\{\log p(Y|V) + \log p(V|\theta)\right\};
\end{aligned} \tag{14}$$

i.e., optimization of the marginal likelihood $p(Y|\theta)$ reduces here to joint optimization of the function $\log p(Y|V) + \log p(V|\theta)$ in (V, θ). Since this function is jointly quadratic and negative semidefinite in (V, θ), we need only apply a single step of Newton's method. Now if we examine the Hessian H of this objective function, we see that it is of block form:

$$\begin{aligned}
H &= \nabla\nabla_{(V,\theta)}\left[\log p(Y|V) + \log p(V|\theta)\right] \\
&= \begin{pmatrix} H_{VV} & H_{V\theta}^T \\ H_{\theta V} & H_{\theta\theta} \end{pmatrix},
\end{aligned}$$

where the H_{VV} block is itself block-tridiagonal, with T blocks of size n (where n is the number of compartments in our observed dendrite) and the $H_{\theta\theta}$ block is of size $n \times n$. If we apply the Schur complement to this block H and then exploit the fast methods for solving linear equations involving H_{VV}, it is easy to see that solving for $\hat{\theta}$ can be done in a single $O(T)$ step; see Koyama and Paninski (2009) for details.

Figure 4 shows a simulated example of this inferred θ in which $\vec{U}(t)$ is chosen to be two-dimensional (corresponding to inputs from two presynaptic cells, one excitatory and one inhibitory); given only half a second of intermittently-sampled, noisy data, the posterior $p(\theta|Y)$ is quite accurately concentrated about the true underlying value of θ.

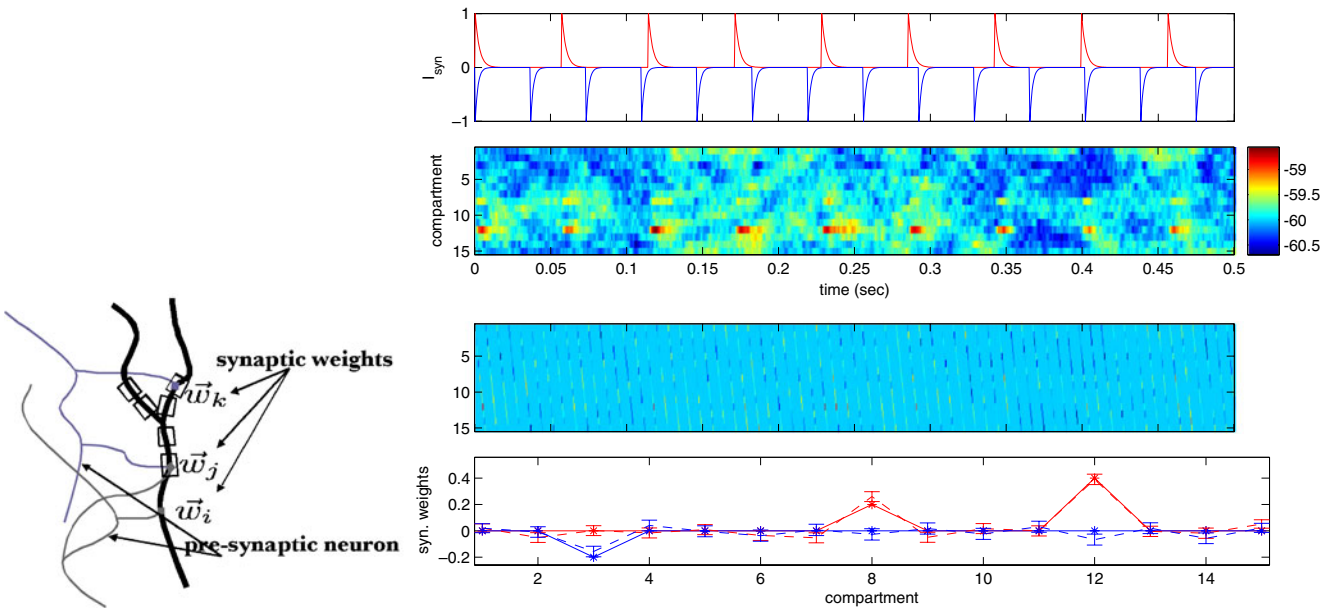

Fig. 4 Estimating the location and strength of synapses on a simulated dendritic branch. *Left*: simulation schematic. By observing a noisy, subsampled spatiotemporal voltage signal on the dendritic tree, we can infer the strength of a given presynaptic cell's inputs at each location on the postsynaptic cell's dendritic tree. *Right*: illustration of the method applied to simulated data (Paninski and Ferreira 2008). We simulated two presynapic inputs: one excitatory (*red*) and one inhibitory (*blue*). The (known) presynaptic spike trains are illustrated in the *top panel*, convolved with exponential filters of $\tau = 3$ ms (excitatory) and $\tau = 2$ ms (inhibitory) to form U_t. *Second panel*: True (unobserved) voltage (mV), generated from the cable Eq. (13). Note that each presynaptic spike leads to a post-synaptic potential of rather small magnitude (at most ≈ 1 mV), relative to the voltage noise level. In this case the excitatory presynaptic neuron synapses twice on the neuron, on compartment 12 and and a smaller synapse on compartment 8, while the inhibitory neuron synapses on compartment 3. *Third panel*: Observed (raster-scanned) voltage. The true voltage was not observed directly; instead, we only observed a noisy, spatially-rastered (linescanned), subsampled version of this signal. Note the very low effective SNR here. Bottom panel: True and inferred synaptic weights. The true weight of each synapse is indicated by an asterisk (*) and the errorbar shows the posterior mean $E(\theta_i|Y)$ and standard deviation $\sqrt{Var(\theta_i|Y)}$ of the synaptic weight given the observed data. Note that inference is quite accurate, despite the noisiness and the brief duration (500 ms) of the data

The form of the joint optimization in Eq. (14) sheds a good deal of light on the relatively slow behavior of the EM algorithm here. The E step here involves inferring V given Y and θ; in this Gaussian model, the mean and mode of the posterior $p(V|\theta, Y)$ are equal, and so we see that

$$\hat{V} = \arg\max_V \log p(V|\theta, Y)$$
$$= \arg\max_V \left\{ \log p(Y|V) + \log p(V|\theta) \right\}.$$

Similarly, in the M-step, we compute $\arg\max_\theta \log p(\hat{V}|\theta)$. So we see that EM here is simply coordinate ascent on the objective function in Eq. (14): the E step ascends in the V direction, and the M step ascends in the θ direction. (In fact, it is well-known that EM may be written as a coordinate ascent algorithm much more generally; see Neal and Hinton (1999) for details.) Coordinate ascent requires more steps than Newton's method in general, due to the "zigzag" nature of the solution path in the coordinate ascent algorithm (Press et al. 1992), and this is exactly our experience in practice.[7]

[7] An additional technical advantage of the direct optimization approach is worth noting here: to compute the E step via the Kalman filter, we need to specify some initial condition for $p(V(0))$. When we have no good information about the initial $V(0)$, we can use "diffuse" initial conditions, and set the initial covariance $Cov(V(0))$ to be large (even infinite) in some or all directions in the n-dimensional $V(0)$-space. A crude way of handling this is to simply set the initial covariance in these directions to be very large (instead of infinite), though this can lead to numerical instability. A more rigorous approach is to take limits of the update equations as the uncertainty becomes large, and keep separate track of the infinite and non-infinite terms appropriately; see Durbin and Koopman (2001) for details. At any rate, these technical difficulties are avoided in the direct optimization approach, which can handle infinite prior covariance easily (this just corresponds to a zero term in the Hessian of the log-posterior).

As emphasized above, the linear-Gaussian case we have treated here is special, because the Hessian H is constant, and therefore the $\log|H|$ term in Eq. (12) can be neglected. However, in some cases we can apply a similar method even when the observations are non-Gaussian; see Koyama and Paninski (2009), Vidne et al. (2009) for examples and further details.

4 Generalizing the state-space method: exploiting banded and sparse Hessian matrices

Above we have discussed a variety of applications of the direct $O(T)$ optimization idea. It is natural to ask whether we can generalize this method usefully beyond the state space setting. Let's look more closely at the assumptions we have been exploiting so far. We have restricted our attention to problems where the log-posterior $p(Q|Y)$ is log-concave, with a Hessian H such that the solution of the linear equation $\nabla = HQ$ can be computed much more quickly than the standard $O(\dim(Q)^3)$ required by a generic linear equation. In this section we discuss a couple of examples that do not fit gracefully in the state-space framework, but where nonetheless we can solve $\nabla = HQ$ quickly, and therefore very efficient inference methods are available.

4.1 Example: banded matrices and fast optimal stimulus decoding

The neural decoding problem is a fundamental question in computational neuroscience (Rieke et al. 1997): given the observed spike trains of a population of cells whose responses are related to the state of some behaviorally-relevant signal $x(t)$, how can we estimate, or "decode," $x(t)$? Solving this problem experimentally is of basic importance both for our understanding of neural coding (Pillow et al. 2008) and for the design of neural prosthetic devices (Donoghue 2002). Accordingly, a rather large literature now exists on developing and applying decoding methods to spike train data, both in single cell- and population recordings; see Pillow et al. (2009), Ahmadian et al. (2009a) for a recent review.

We focus our attention here on a specific example. Let's assume that the stimulus $x(t)$ is one-dimensional, with a jointly log-concave prior $p(X)$, and that the Hessian of this log-prior is banded at every point X. Let's also assume that the observed neural population whose responses we are attempting to decode may be well-modeled by the generalized linear model framework applied in Pillow et al. (2008):

$$\lambda_i(t) = \exp\left(\mathbf{k}_i * x(t) + \mathbf{h}_i \cdot \mathbf{y}_i + \left(\sum_{i \neq j} \mathbf{l}_{ij} \cdot \mathbf{y}_j\right) + \mu_i\right).$$

Here $*$ denotes the temporal convolution of the filter $\mathbf{k}_i$ against the stimulus $x(t)$. This model is equivalent to Eq. (7), but we have dropped the common-input term $q(t)$ for simplicity here.

Now it is easy to see that the loglikelihood $\log p(Y|X)$ is concave, with a banded Hessian, with bandwidth equal to the length of the longest stimulus filter $\mathbf{k}_i$ (Pillow et al. 2009). Therefore, Newton's method applied to the log-posterior $\log p(X|Y)$ requires just $O(T)$ time, and optimal Bayesian decoding here runs in time comparable to standard linear decoding (Warland et al. 1997). Thus we see that this stimulus decoding problem is a rather straightforward extension of the methods we have discussed above: instead of dealing with block-tridiagonal Hessians, we are now simply exploiting the slightly more general case of a banded Hessian. See Fig. 5.

The ability to quickly decode the input stimulus $x(t)$ leads to some interesting applications. For example, we can perform perturbation analyses with the decoder: by jittering the observed spike times (or adding or removing spikes) and quantifying how sensitive the decoded stimulus is to these perturbations, we may gain insight into the importance of precise spike timing and correlations in this multineuronal spike train data (Ahmadian et al. 2009b). Such analyses would be formidably slow with a less computationally-efficient decoder.

See Ahmadian et al. (2009a) for further applications to fully-Bayesian Markov chain Monte Carlo (MCMC) decoding methods; bandedness can be exploited quite fruitfully for the design of fast preconditioned Langevin-type algorithms (Robert and Casella 2005). It is also worth noting that very similar banded matrix computations arise naturally in spline models (Wahba 1990; Green and Silverman 1994), which are at the heart of the powerful BARS method for neural smoothing (DiMatteo et al. 2001; Kass et al. 2003); see Ahmadian et al. (2009a) for a discussion of how to exploit bandedness in the BARS context.

4.2 Smoothness regularization and fast estimation of spatial tuning fields

The applications discussed above all involve state-space models which evolve through time. However, these

ideas are also quite useful in the context of spatial models Moeller and Waagepetersen (2004). Imagine we would like to estimate some two-dimensional rate function from point process observations. Rahnama et al. (2009) discuss a number of distinct cases of this problem, including the estimation of place fields in hippocampus (Brown et al. 1998) or of tuning functions in motor cortex (Paninski et al. 2004); for concreteness, we focus here on the setting considered in Czanner et al. (2008). These authors analyzed repeated observations of a spike train whose mean rate function changed gradually from trial to trial; the goal of the analysis here is to infer the firing rate $\lambda(t,i)$, where t denotes the time within a trial and i denotes the trial number.

One convenient approach to this problem is to model $\lambda(t,i)$ as

$$\lambda(t,i) = f[q(t,i)],$$

and then to discretize (t,i) into two-dimensional bins in which $q(t,i)$ may be estimated by maximum likelihood in each bin. However, as emphasized in Kass et al. (2003); Smith and Brown (2003); Kass et al. (2005); Czanner et al. (2008), this crude histogram approach can lead to highly variable, unreliable estimates of the firing rate $\lambda(t,i)$ if the histogram bins are taken to be too small; conversely, if the bins are too large, then we may overcoarsen our estimates and lose valuable information about rapid changes in the firing rate as a function of time t or trial number i.

A better approach is to use a fine binwidth to estimate λ, but to regularize our estimate so that our inferred $\hat{\lambda}$ is not overly noisy. One simple approach is to compute a penalized maximum likelihood estimate

$$\hat{Q} = \arg\max_{Q} \left[\log p(Y|Q) - c_1 \sum_{it} [q(t,i) - q(t-dt,i)]^2 - c_2 \sum_{it} [q(t,i) - q(t,i-1)]^2 \right]; \quad (15)$$

the observation likelihood $p(Y|Q)$ is given by the standard point-process log likelihood

$$\log p(Y|Q) = \sum_{it} y_{it} \log f[q(t,i)] - f[q(t,i)]dt,$$

where dt denotes the temporal binwidth; y_{it} here denotes the number of spikes observed in time bin t during trial i (c.f. Eq. (8)). The constants c_1 and c_2 serve as regularizer weights: if c_1 is large, then we penalize strongly for fluctuations in $q(t,i)$ along the t-axis, whereas conversely c_2 sets the smoothness along the i-axis.

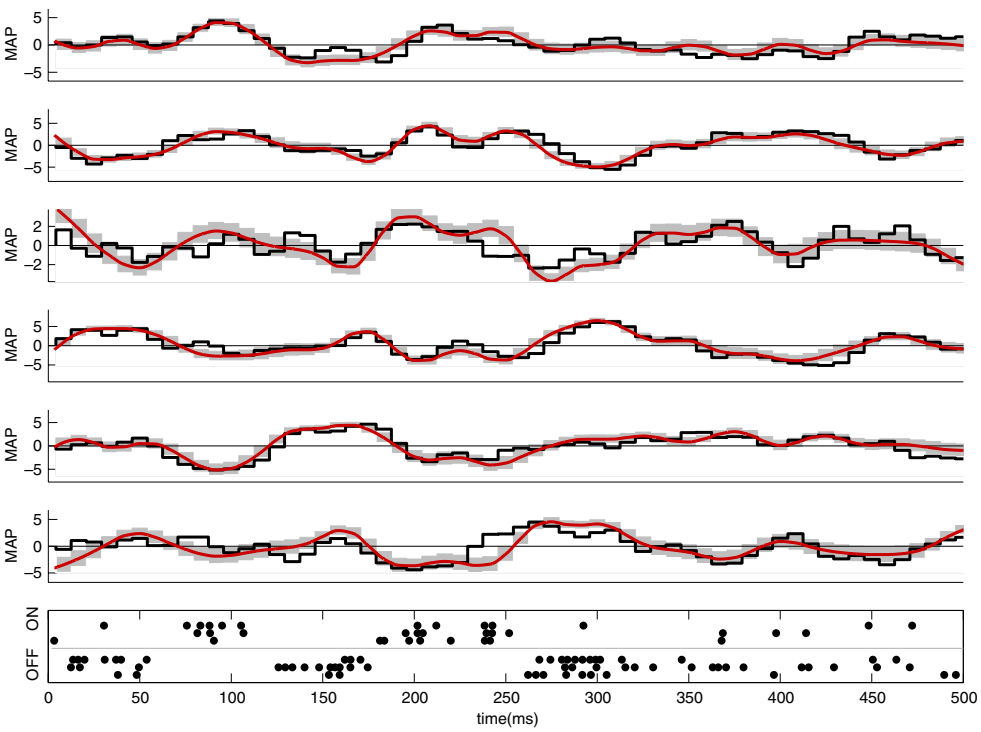

Fig. 5 MAP decoding of the spatio-temporal stimulus $\mathbf{k}_i * x(t)$ from the simultaneously recorded spike trains of three pairs of ON and OFF retinal ganglion cells, again from Pillow et al. (2008). The *top six panels* show the true input $\mathbf{k}_i * x(t)$ to each cell (*jagged black line*; the filters $\mathbf{k}_i$ were estimated by maximum likelihood from a distinct training set here), and the decoded MAP estimate (*smooth curve*) ± 1 posterior s.d. (*gray area*). The MAP estimate is quite accurate and is computed in $O(T)$ time, where T is the stimulus duration. In this example, fully-Bayesian Markov chain Monte Carlo decoding produced nearly identical results; see Ahmadian et al. (2009a) for details

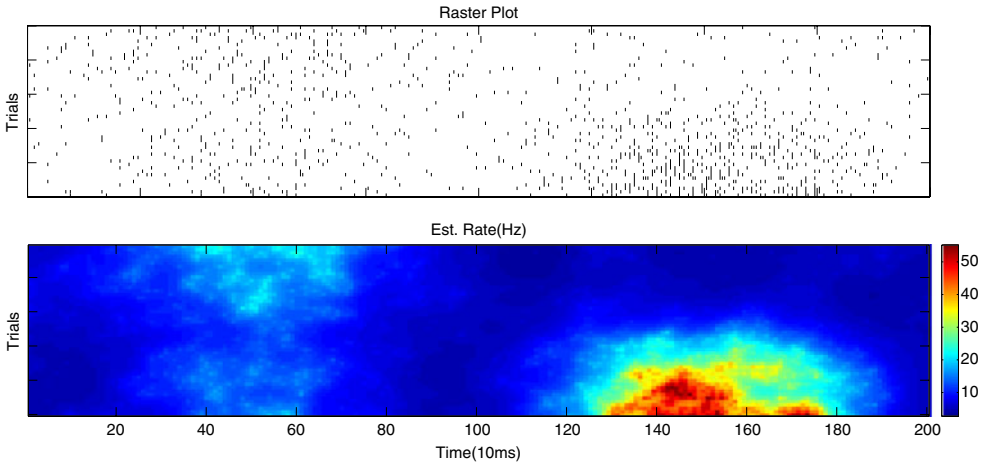

Fig. 6 An example of the fast spatial estimation method applied to data from Czanner et al. (2008), Fig. 3. *Top panel*: observed spike train data. Note that the firing rate qualitatively changes both as a function of time t and trial number i; 50 trials total are observed here. *Bottom*: $\lambda(t, i)$ estimated using the fast regularized methods described in Rahnama et al. (2009). See Rahnama et al. (2009) for further comparisons, e.g. to linear kernel smoothing methods

This quadratic penalty has a natural Bayesian interpretation: $\hat{Q}$ is the MAP estimate under a "smoothing" Gaussian prior of the form

$$\log p(Q) = -c_1 \sum_{it} [q(t,i) - q(t-dt,i)]^2 - c_2 \sum_{it} [q(t,i) - q(t,i-1)]^2 + const. \quad (16)$$

(Note that this specific Gaussian prior is improper, since the quadratic form in Eq. (16) does not have full rank—the sums in $\log p(Q)$ evaluate to zero for any constant Q—and the prior is therefore not integrable. This can be corrected by adding an additional penalty term, but in practice, given enough data, the posterior is always integrable and therefore this improper prior does not pose any serious difficulty.)

Now the key problem is to compute $\hat{Q}$ efficiently. We proceed as before and simply apply Newton's method to optimize Eq. (15). If we represent the unknown Q as a long vector formed by appending the columns of the matrix $q(t, i)$, then the Hessian with respect to Q still has a block-tridiagonal form, but now the size of the blocks scales with the number of trials observed, and so direct Gaussian elimination scales more slowly than $O(N)$, where N is the dimensionality (i.e., the total number of pixels) of $q(t, i)$. Nonetheless, efficient methods have been developed to handle this type of sparse, banded matrix (which arises, for example, in applied physics applications requiring discretized implementations of Laplace's or Poisson's equation); for example, in this case Matlab's built in $H \backslash \nabla$ code computes the solution to $Hx = \nabla$ in $O(N^{3/2})$ time, which makes estimation of spatial tuning functions with $N \sim 10^4$ easily feasible (on the order of seconds on a laptop). See Fig. 6 for an example of an estimated two-dimensional firing rate $\lambda(t, i)$, and Rahnama et al. (2009) for further details.

5 Conclusion

Since the groundbreaking work of Brown et al. (1998), state-space methods have been recognized as a key paradigm in neuroscience data analysis. These methods have been particularly dominant 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 2008) 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 et al. 2009), where the need for statistically rigorous and computationally efficient methods for tracking a dynamic "moving target" given noisy, indirect spike train observations has been particularly acute.

The direct optimization viewpoint discussed here (and previously in Fahrmeir and Kaufmann 1991, Fahrmeir and Tutz 1994, Bell 1994, Davis and Rodriguez-Yam 2005, Jungbacker and Koopman 2007, Koyama and Paninski 2009) opens up a number of additional interesting applications in neuroscience, and has a couple advantages, as we have emphasized. Pragmatically, this method is perhaps a bit conceptually simpler and easier to code, thanks to the efficient sparse matrix methods built into Matlab and other modern numerical software packages. The joint optimization approach makes the important extension to problems of constrained optimization quite transparent, as we saw in Section 2.4. We also saw that the direct techniques outlined in Section 3 can provide a much more efficient

algorithm for parameter estimation than the standard Expectation-Maximization strategy. In addition, the direct optimization approach makes the connections to other standard statistical methods (spline smoothing, penalized maximum likelihood, isotonic regression, etc.) quite clear, and can also serve as a quick initialization for more computationally-intensive methods that might require fewer model assumptions (e.g., on the concavity of the loglikelihoods $p(y_t|q_t)$). Finally, this direct optimization setting may be generalized significantly: we have mentioned extensions of the basic idea to constrained problems, MCMC methods, and spatial smoothing applications, all of which amply illustrate the flexibility of this approach. We anticipate that these state-space techniques will continue to develop in the near future, with widespread and diverse applications to the analysis of neural data.

Acknowledgements We thank J. Pillow for sharing the data used in Figs. 2 and 5, G. Czanner for sharing the data used in Fig. 6, and B. Babadi and Q. Huys for many helpful discussions. LP is supported by NIH grant R01 EY018003, an NSF CAREER award, and a McKnight Scholar award; YA by a Patterson Trust Postdoctoral Fellowship; DGF by the Gulbenkian PhD Program in Computational Biology, Fundacao para a Ciencia e Tecnologia PhD Grant ref. SFRH / BD / 33202 / 2007; SK by NIH grants R01 MH064537, R01 EB005847 and R01 NS050256; JV by NIDCD DC00109.

References

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

Ahmadian, Y., Pillow, J., Shlens, J., Chichilnisky, E., Simoncelli, E., & Paninski, L. (2009b). A decoder-based spike train metric for analyzing the neural code in the retina. *COSYNE09*.

Araya, R., Jiang, J., Eisenthal, K. B., & Yuste, R. (2006). The spine neck filters membrane potentials. *PNAS, 103*(47), 17961–17966.

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

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

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

Boyd, S., & Vandenberghe, L. (2004). *Convex optimization*. Oxford: Oxford University Press.

Brockwell, A., Rojas, A., & 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., & 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., Kass, R., & Mitra, P. (2004). Multiple neural spike train data analysis: State-of-the-art and future challenges. *Nature Neuroscience, 7*, 456–461.

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

Chornoboy, E., Schramm, L., & Karr, A. (1988). Maximum likelihood identification of neural point process systems. *Biological Cybernetics, 59,* 265–275.

Coleman, T., & 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., & Yuste, R. (2003). Attractor dynamics of network up states in the neocortex. *Nature, 423*, 283–288.

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

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

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

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

Dempster, A., Laird, N., & Rubin, D. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B, 39*, 1–38.

DiMatteo, I., Genovese, C., & Kass, R. (2001). Bayesian curve fitting with free-knot splines. *Biometrika, 88*, 1055–1073.

Djurisic, M., Popovic, M., Carnevale, N., & Zecevic, D. (2008). Functional structure of the mitral cell dendritic tuft in the rat olfactory bulb. *Journal of Neuroscience, 28*(15), 4057–4068.

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

Doucet, A., de Freitas, N., & Gordon, N. (Eds.) (2001). *Sequential Monte Carlo in practice*. New York: Springer.

Durbin, J., & Koopman, S. (2001). *Time series analysis by state space methods*. Oxford: Oxford University Press.

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

Ergun, A., Barbieri, R., Eden, U., Wilson, M., & Brown, E. (2007). Construction of point process adaptive filter algorithms for neural systems using sequential Monte Carlo methods. *IEEE Transactions on Biomedical Engineering, 54*, 419–428.

Escola, S., & Paninski, L. (2009). Hidden Markov models applied toward the inference of neural states and the improved estimation of linear receptive fields. *Neural Computation* (under review).

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

Fahrmeir, L., & Tutz, G. (1994). *Multivariate statistical modelling based on generalized linear models*. New York: Springer.

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

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

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

Godsill, S., Doucet, A., & West, M. (2004). Monte Carlo smoothing for non-linear time series. *Journal of the American Statistical Association, 99*, 156–168.

Green, P., & Silverman, B. (1994). *Nonparametric regression and generalized linear models*. Boca Raton: CRC.

Hawkes, A. (2004). Stochastic modelling of single ion channels. In J. Feng (Ed.), *Computational neuroscience: A comprehensive approach* (pp. 131–158). Boca Raton: CRC.

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

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

Huys, Q., & Paninski, L. (2009). Model-based smoothing of, and parameter estimation from, noisy biophysical recordings. *PLOS Computational Biology, 5*, e1000379.

Iyengar, S. (2001). The analysis of multiple neural spike trains. In *Advances in methodological and applied aspects of probability and statistics* (pp. 507–524). New York: Gordon and Breach.

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

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

Kass, R., & Raftery, A. (1995). Bayes factors. *Journal of the American Statistical Association, 90*, 773–795.

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

Kass, R. E., Ventura, V., & Brown, E. N. (2005). Statistical issues in the analysis of neuronal data. *Journal of Neurophysiology, 94*, 8–25.

Kelly, R., & Lee, T. (2004). Decoding V1 neuronal activity using particle filtering with Volterra kernels. *Advances in Neural Information Processing Systems, 15*, 1359–1366.

Kemere, C., Santhanam, G., Yu, B. M., Afshar, A., Ryu, S. I., Meng, T. H., et al. (2008). Detecting neural-state transitions using hidden Markov models for motor cortical prostheses. *Journal of Neurophysiology, 100*, 2441–2452.

Khuc-Trong, P., & Rieke, F. (2008). Origin of correlated activity between parasol retinal ganglion cells. *Nature Neuroscience, 11*, 1343–1351.

Kitagawa, G., & Gersch, W. (1996). Smoothness priors analysis of time series. *Lecture notes in statistics* (Vol. 116). New York: Springer.

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

Koyama, S., & 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* doi:10.1007/s10827-009-0150-x.

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

Kulkarni, J., & Paninski, L. (2008). Efficient analytic computational methods for state-space decoding of goal-directed movements. *IEEE Signal Processing Magazine, 25*(special issue on brain-computer interfaces), 78–86.

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

Litke, A., Bezayiff, N., Chichilnisky, E., Cunningham, W., Dabrowski, W., Grillo, A., et al. (2004). What does the eye tell the brain? Development of a system for the large scale recording of retinal output activity. *IEEE Transactions on Nuclear Science*, 1434–1440.

Martignon, L., Deco, G., Laskey, K., Diamond, M., Freiwald, W., & Vaadia, E. (2000). Neural coding: Higher-order temporal patterns in the neuro-statistics of cell assemblies. *Neural Computation, 12*, 2621–2653.

Meng, X.-L., & Rubin, D. B. (1991). Using EM to obtain asymptotic variance-covariance matrices: The SEM algorithm. *Journal of the American Statistical Association, 86*(416), 899–909.

Minka, T. (2001). *A family of algorithms for Approximate Bayesian Inference*. PhD thesis, MIT.

Moeller, J., Syversveen, A., & Waagepetersen, R. (1998). Log-Gaussian Cox processes. *Scandinavian Journal of Statistics, 25*, 451–482.

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

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

Neal, R., & Hinton, G. (1999). A view of the EM algorithm that justifies incremental, sparse, and other variants. In M. Jordan (Ed.), *Learning in graphical models* (pp. 355–368). Cambridge: MIT.

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

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

Nykamp, D. (2005). Revealing pairwise coupling in linear-nonlinear networks. *SIAM Journal on Applied Mathematics, 65*, 2005–2032.

Nykamp, D. (2007). A mathematical framework for inferring connectivity in probabilistic neuronal networks. *Mathematical Biosciences, 205*, 204–251.

Ohki, K., Chung, S., Ch'ng, Y., Kara, P., & Reid, C. (2005). Functional imaging with cellular resolution reveals precise micro-architecture in visual cortex. *Nature, 433*, 597–603.

Olsson, R. K., Petersen, K. B., & Lehn-Schioler, T. (2007). State-space models: From the EM algorithm to a gradient approach. *Neural Computation, 19*, 1097–1111.

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

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

Paninski, L., & Ferreira, D. (2008). State-space methods for inferring synaptic inputs and weights. *COSYNE*.

Peña, J.-L. & 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.

Penny, W., Ghahramani, Z., & Friston, K. (2005). Bilinear dynamical systems. *Philosophical Transactions of the Royal Society of London, 360*, 983–993.

Pillow, J., Ahmadian, Y., & Paninski, L. (2009). Model-based decoding, information estimation, and change-point detection in multi-neuron spike trains. *Neural Computation* (under review).

Pillow, J., Shlens, J., Paninski, L., Sher, A., Litke, A., Chichilnisky, E., et al. (2008). Spatiotemporal correlations and visual signaling in a complete neuronal population. *Nature, 454*, 995–999.

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

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

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

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

Rasmussen, C., & Williams, C. (2006). *Gaussian processes for machine learning*. Cambridge: MIT.

Rieke, F., Warland, D., de Ruyter van Steveninck, R., & Bialek, W. (1997). *Spikes: Exploring the neural code*. Cambridge: MIT.

Robert, C., & Casella, G. (2005). *Monte Carlo statistical methods*. New York: Springer.

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

Rybicki, G., & 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., & Press, W. H. (1995). Class of fast methods for processing irregularly sampled or otherwise inhomogeneous one-dimensional data. *Physical Review Letters, 74*(7), 1060–1063.

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

Schneidman, E., Berry, M., Segev, R., & Bialek, W. (2006). Weak pairwise correlations imply strongly correlated network states in a neural population. *Nature, 440*, 1007–1012.

Schnitzer, M., & Meister, M. (2003). Multineuronal firing patterns in the signal from eye to brain. *Neuron, 37*, 499–511.

Shlens, J., Field, G. D., Gauthier, J. L., Grivich, M. I., Petrusca, D., Sher, A., et al. (2006). The structure of multi-neuron firing patterns in primate retina. *Journal of Neuroscience, 26*, 8254–8266.

Shlens, J., Field, G. D., Gauthier, J. L., Greschner, M., Sher, A., Litke, A. M., et al. (2009). The structure of large-scale synchronized firing in primate retina. *Journal of Neuroscience, 29*, 5022–5031.

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

Shumway, R., & Stoffer, D. (2006). *Time series analysis and its applications*. New York: Springer.

Silvapulle, M., & Sen, P. (2004). *Constrained statistical inference: Inequality, order, and shape restrictions*. New York: Wiley-Interscience.

Smith, A., & 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., et al. (2004). Dynamic analysis of learning in behavioral experiments. *Journal of Neuroscience, 24*(2), 447–461.

Smith, A. C., Stefani, M. R., Moghaddam, B., & Brown, E. N. (2005). Analysis and design of behavioral experiments to characterize population learning. *Journal of Neurophysiology, 93*(3), 1776–1792.

Snyder, D., & Miller, M. (1991). *Random point processes in time and space*. New York: Springer.

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

Suzuki, W. A., & Brown, E. N. (2005). Behavioral and neurophysiological analyses of dynamic learning processes. *Behavioral & Cognitive Neuroscience Reviews, 4*(2), 67–95.

Truccolo, W., Eden, U., Fellows, M., Donoghue, J., & 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.

Utikal, K. (1997). A new method for detecting neural interconnectivity. *Biological Cyberkinetics, 76*, 459–470.

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

Vogelstein, J., Babadi, B., Watson, B., Yuste, R., & Paninski, L. (2008). 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., & Paninski, L., (2009). Model-based optimal inference of spike times and calcium dynamics given noisy and intermittent calcium-fluorescence imaging. *Biophysical Journal*. http://www.stat.columbia.edu/ liam/research/abstracts/vogelsteinbj08- abs.html.

Wahba, G. (1990). *Spline models for observational data*. Philadelphia: SIAM.

Wang, X., Wei, Y., Vaingankar, V., Wang, Q., Koepsell, K., Sommer, F., & 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.

Warland, D., Reinagel, P., & Meister, M. (1997). Decoding visual information from a population of retinal ganglion cells. *Journal of Neurophysiology, 78*, 2336–2350.

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

West, M., & Harrison, P., (1997). *Bayesian forecasting and dynamic models*. New York: Springer.

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

Wu, W., Kulkarni, J., Hatsopoulos, N., & Paninski, L. (2009). Neural decoding of goal-directed movements using a linear statespace model with hidden states. *IEEE Transactions on Biomedical Engineering* (in press).

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

Ypma, A., & Heskes, T., (2003). Iterated extended Kalman smoothing with expectation-propagation. *Neural Networks for Signal Processing, 2003*, 219–228.

Yu, B., Afshar, A., Santhanam, G., Ryu, S., Shenoy, K., & Sahani, M. (2006). Extracting dynamical structure embedded in neural activity. *NIPS*.

Yu, B. M., Cunningham, J. P., Shenoy, K. V., & Sahani, M. (2007). Neural decoding of movements: From linear to nonlinear trajectory models. *ICONIP*, 586–595.

Yu, B. M., Kemere, C., Santhanam, G., Afshar, A., Ryu, S. I., Meng, T. H., et al. (2007). Mixture of trajectory models for neural decoding of goal-directed movements. *Journal of Neurophysiology, 97*(5), 3763–3780.

Yu, B. M., Shenoy, K. V., & Sahani, M. (2006). Expectation propagation for inference in non-linear dynamical models with Poisson observations. In *Proceedings of the nonlinear statistical signal processing workshop* (pp. 83–86). Piscataway: IEEE.

Zhang, K., Ginzburg, I., McNaughton, B., & Sejnowski, T. (1998). Interpreting neuronal population activity by reconstruction: Unified framework with application to hippocampal place cells. *Journal of Neurophysiology, 79*, 1017–1044.