Fast spike train inference from calcium imaging

Joshua T. Vogelstein, others, Liam Paninski

June 11, 2009

Contents

Abstract

Experiments often yield measurements of variables that are naturally constrained to be nonnegative. In such scenarios, it may be desirable to filter (or deconvolve) the observations to find the most likely trajectory of the nonnegative variable, given the noisy observations. Here, we develop a computationally-efficient optimal filter for a certain subset of nonnegatively constrained deconvolutions. Specifically, for any nonnegative variable that is filtered by a matrix linear differential equation and observed with independent, log-concave noise, we can infer the optimal nonnegative trajectory via straightforward interior-point methods in $O(T)$ (linear) time, as opposed to more standard approaches requiring $O(T^3)$ time (where T is the total number of time steps). The key is to make use of the tridiagonal structure of the Hessian of the log-posterior here, which allows us to perform each Newton iteration in linear time. We apply this filter to an important problem in neuroscience: inferring a spike train from noisy calcium fluorescence observations. We demonstrate the filter's improved performance on simulated and real data. In conclusion, we propose that this filter is readily applicable for a number of real-time applications, including spike inference from simultaneously-observed large neural populations.

1 Introduction

Simultaneously imaging large populations of neurons using calcium sensors is becoming increasingly popular [1, 2, 3, 4], especially as the signal-to-noise-ratio (SNR) of genetic sensors continues to improve [5, 6, 7]. While this technology facilitates acquiring data from an unprecedented number of neurons simultaneously, it is not yet a panacea: there is a trade-off between the quantity of neurons, and the quality of data. While one can now acquire data from ~ 100 neurons within a single experimental session, the data for each neuron is relatively poor. In comparison with extracellular electrophysiology, the data from calcium imaging has reduced (i) SNR and (ii) temporal resolution. Thus, relatively few studies have been able to use this approach to date to ask *quantitative* questions about the relationship between spike trains, and, for example, sensory stimuli [8, 9, 10, 11, 12, 13, 14].

A number of groups have proposed spike inference algorithms to facilitate using calcium imaging to ask quantitative questions about spike timing. Finding the most likely spike train, given a calcium movie, is computationally intractable because we have to perform a search over all possible spike trains, and the number of possible spike trains scales exponentially with the number of image frames, T (more specifically, assuming only a single spike can occur per frame, we have 2^T possible spike trains). To circumvent this problem, various groups have developed distinct strategies. For instance, Greenberg et al. [15] reduced the search space by establishing heuristics to preprocess the data, and effectively exclude many image frames from the search space. In our previous work [16], we developed a sequential Monte Carlo method to efficiently sample spike trains given the fluorescence data, which is guaranteed to perform optimally given our model, which incorporates saturation, refractoriness, and stimulus dependent effects (this approach has the added advantage of providing a measure of uncertainty as well). Unfortunately, both these approaches are relatively slow, as they still require searches over large spaces. Holekamp et al. [17] took a very different strategy, by performing the optimal linear deconvolution (i.e., the Wiener filter) on the fluorescence data. This approach may be thought of as finding the *maximum a posteriori* (MAP) spike train.

The present work also develops a deconvolution filter, but has several advantages over the Wiener filter. First, we impose a non-negative constraint on the solution. Because we know that spikes are non-negative events, this is a desirable constraint to impose on our inference procedure. In practice, this constraint regularizes the resulting inference [18, 19, 20], by combating against "ringing" overfitting effects. Second, the computational time of our filter scales linearly with T, whereas the Wiener filter typically scales according to $T \log T$. Thus, we have named our approach the FAst Non-negative Deconvolution (FAND) filter. This FAND filter could be applicable in a number of scientific investigations from myriad fields, and is closely related to the problem of non-negative matrix factorization, a problem currently receiving much attention from the machine learning community [21, 18, 19, 22, 23].

The remainder of this paper is organized as follows. In Section 2, we describe our parametric model, and the FAND filter. Section 3 first provides our main result: we can utilize FAND to filter calcium fluorescence data, with better results than the optimal linear filter. We then generalize the FAND filter in a number of directions. First, instead of assuming the data is a 1-dimensional fluorescence time series, we operate on all the pixels within a region-of-interest (ROI), which can significantly improve the SNR of our inference. Second, by embedding our FAND filter into a pseudo-expectation-maximization algorithm, we can learn the parameters of our model without any training data (i.e., without requiring simultaneous imaging and electrophysiology). To demonstrate the utility of this approach, we compare our filter and the Wiener filter's output on in vitro data. Third, when imaging a large population of neurons simultaneously, sometimes segmenting the image to obtain one neuron per ROI is difficult. Therefore, we show how approach can be further generalized to deal with such a scenario. We then apply our filter to a movie of ~ 50 neurons recorded simultaneously in vitro, to indicate that this approach works "out-of-the-box" on segmented data. Fourth, we modify our model to handle data collected in vivo using genetic sensors. Specifically, this means allowing for a slow rise time and poisson observations. Finally, in the discussion, we show how to incorporate non-linear saturation into the FAND filter, and discuss how the output of our FAND filter can be used to initialize our more powerful (but slower) algorithms.

2 Methods

2.1 Model

We start by assuming a very simple generative model, relating the hidden neural activity (spike trains and intracellular calcium concnetrations) and the observations (fluorescence movies). More specifically, we start by assuming we have collected a 1-dimensional fluorescence trace, $\boldsymbol{F} = (F_1, \ldots, F_T)$ from a neuron. At time t, the fluorescence measurement, F_t is a linear-Gaussian function of the intracellular calcium concentration at that time, C_t:

$$F_t = \alpha(C_t + \beta) + \sigma\varepsilon_t, \qquad \varepsilon_t \overset{iid}{\sim} \mathcal{N}(0,1) \tag{1}$$

The scale, α, absorbs all experimental variables impacting the scale of the signal, including number of sensors within the cell, photon/$\Delta[\text{Ca}^{2+}]$ per sensor, amplification of imaging system, etc. Similarly, the offset, β, absorbs baseline calcium concentration of the cell, background fluorescence of the fluorophore, imaging system offset, etc. The standard deviation, σ, results from calcium fluctuations independent of spiking activity, fluorescence fluctuations independent of calcium, and imaging noise. These three parameters therefore correspond to a number of simplifying assumptions, that we will relax in Section 3.

We further assume that the intracellular calcium concentration, C_t, jumps after each spike, and subsequently decays back down to rest with time constant, τ, yielding $\tau(C_t - C_{t-1})/\Delta = -C_{t-1} + n_t$. Rearranging (and rescaling) a bit, we have:

$$C_t = \gamma C_{t-1} + n_t, \qquad n_t \overset{iid}{\sim} \text{Poisson}(\lambda\Delta) \tag{2}$$

where n_t indicates the number of times the neuron spiked at time t, and $\gamma = 1 - \Delta/\tau$. [1] Note that C_t does not refer to absolute intracellular concentration of calcium but rather, a relative measure. The assumed linearity of our model precludes the possibility of determining calcium in absolute terms (but see Section **??** for a modified model). Figure 1 depicts a schematic illustration of this model. Note that the model, Eqs. (1) and (2) is a discrete-time state-space model. Importantly, this model differs from the standard models in that we have a non-Gaussian noise term, n_t. Thus, standard tools, such as the Kalman filter-smoother, cannot be applied directly. We therefore develop a novel algorithm to handle problems of this nature, as described below.

[1] This follows from writing (2) as $\tau \frac{C_t - C_{t-1}}{\Delta} = -C_{t-1} + n_t$.

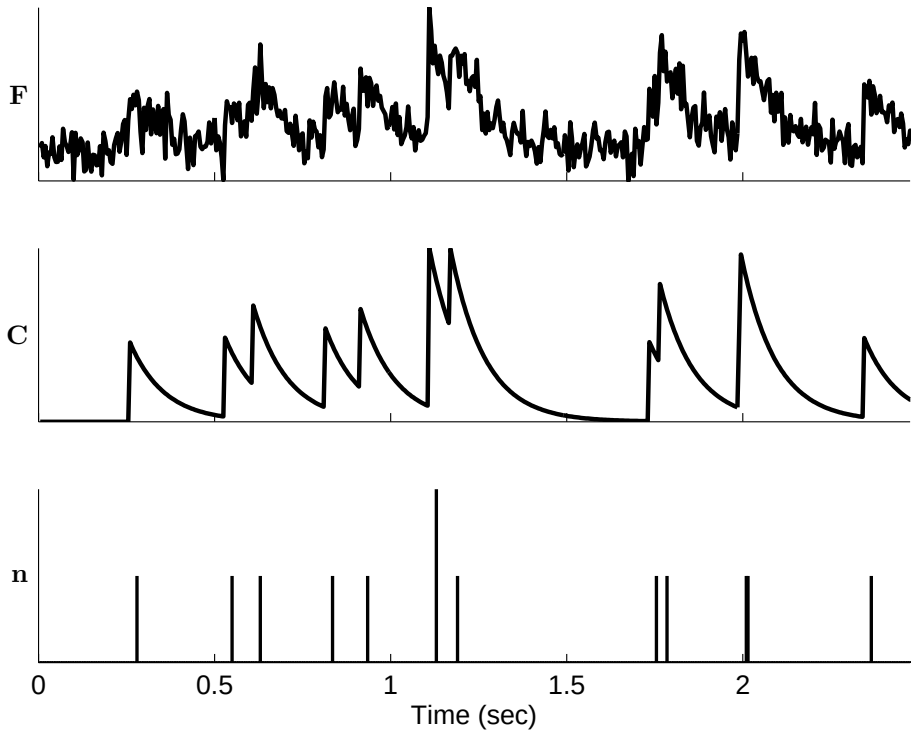

Figure 1: A schematic illustrating our model. The spike train (bottom panel) is convolved with an exponential with time constant τ to obtain the calcium dynamics (middle panel). The fluorescence observations are simply the calcium dynamics, plus zero mean Gaussian noise (with variance σ^2). Parameters: $\Delta = 5$ msec, $\alpha = 1$ photons/μM, $\beta = 0$ unitless, $\sigma = 0.25$ photons, $\tau = 100$ msec, $\lambda = 5$ Hz.

2.2 Inference

Given the above model, our goal is to find the maximum *a posteriori* (MAP) spike train, i.e., the most likely spike train, $\boldsymbol{n}$, given the fluorescence measurements, $\boldsymbol{F}$. Formally, we have:

$$\boldsymbol{n}_{MAP} = \underset{n_t \in \mathbb{N}_0 \forall t}{\operatorname{argmax}} P[\boldsymbol{n}|\boldsymbol{F}] \tag{3a}$$

$$= \underset{n_t \in \mathbb{N}_0 \forall t}{\operatorname{argmax}} P[\boldsymbol{F}|\boldsymbol{n}]P[\boldsymbol{n}] \tag{3b}$$

$$= \underset{n_t \in \mathbb{N}_0 \forall t}{\operatorname{argmax}} \prod_{t=1}^{T} P[F_t|C_t]P[n_t] = \underset{n_t \in \mathbb{N}_0 \forall t}{\operatorname{argmax}} \sum_{t=1}^{T} \big(\log P[F_t|C_t] + \log P[n_t] \big) \tag{3c}$$

$$= \underset{n_t \in \mathbb{N}_0 \forall t}{\operatorname{argmin}} \sum_{t=1}^{T} \left(\frac{1}{2\sigma^2} (F_t - \alpha(C_t + \beta))^2 - n_t \log \lambda\Delta + \log n_t! \right), \tag{3d}$$

where 3a is the definition of the MAP spike train, 3b follows from Bayes' Rule, 3c and 3d follow from the state-space nature of our problem, $\mathbb{N}_0$ is the set of natural numbers (i.e., $\mathbb{N}_0 = \{0, 1, 2, \ldots\}$), T is the number of time steps in the recording, and C_t is implicitly a function of n_t. Unfortunately, the computational complexity of solving Eq. (3) scales exponentially with T, since finding the most likely spike train requires searching over all possible spike trains. Even if n_t were constrained to be some finite number, k, the number of possible spike trains would be k^T, which, for any reasonably sized T, is too many to search over. Thus, instead of an exact solution, we propose to make an approximation that reduces the complexity to be *polynomial* in T. In particular, we relax the assumption that we must

have an integer number of spikes at any time step, by approximating the Poisson distribution with an exponential. Note that this is a common approximation technique in the machine learning literature [24], as it is the closest convex relaxation to its non-convex counterpart. The constraint on n_t in Eq. (3) is therefore relaxed from $n_t \in \mathbb{N}_0$ to $n_t \geq 0$:

$$\widehat{\boldsymbol{n}} \approx \underset{n_t>0\,\forall t}{\operatorname{argmax}} \sum_{t=1}^{T} \left(\frac{1}{2\sigma^2}(F_t - \alpha(C_t+\beta))^2 + n_t \lambda \Delta \right), \qquad \widehat{\boldsymbol{n}} \sim \text{Exponential}(\lambda\Delta) \tag{4}$$

While this convex relaxation makes the problem tractable, the "sharp" threshold imposed by the nonnegativity constraint prohibits the use of standard gradient ascent techniques [25]. We therefore take an "interior-point" (or "barrier") approach, in which we drop the sharp threshold, and add a barrier term, which must approach $-\infty$ as n_t approaches zero (e.g., $-\log n_t$) [25]. By iteratively reducing the weight of the barrier term, we are guaranteed to converge to the correct solution [25]. Thus, our goal is to efficiently solve:

$$\widehat{\boldsymbol{n}}_z = \underset{n_t \forall t}{\operatorname{argmin}} \sum_{t=1}^{T} \left(\frac{1}{2\sigma^2}(F_t - \alpha(C_t+\beta))^2 + n_t\lambda\Delta - z\log(n_t) \right), \tag{5}$$

Since spikes and calcium are related to one another via a simple linear transformation, namely, $n_t = C_t - \gamma C_{t-1}$. Thus, we may rewrite Eq. (5) in terms of $\boldsymbol{C}$:

$$\widehat{\boldsymbol{C}}_z = \underset{C_t - \gamma C_{t-1} \geq 0 \forall t}{\operatorname{argmin}} \sum_{t=1}^{T} \left(\frac{1}{2\sigma^2}(F_t - \alpha(C_t+\beta))^2 + (C_t - \gamma C_{t-1})\lambda\Delta - z\log(C_t - \gamma C_{-1}) \right). \tag{6}$$

Since Eq. (6) is concave, we can use any number of techniques guaranteed to find the global optimum. And because the argument of Eq. (6) is twice differentiable, we elect to use the Newton-Raphson technique. Importantly, the state-space nature of this problem yields a particularly efficient approach. Specifically, note that the Hessian is *tridiagonal*, which is clear upon rewriting (6) in matrix notation:

$$\widehat{\boldsymbol{C}}_z = \underset{\boldsymbol{MC} \geq \boldsymbol{0}}{\operatorname{argmin}} \frac{1}{2\sigma^2} \left\| \boldsymbol{F} - \alpha(\boldsymbol{C}+\beta) \right\|^2 + (\boldsymbol{MC})^\mathsf{T}\boldsymbol{\lambda} - z\log(\boldsymbol{MC})^\mathsf{T}\boldsymbol{1}, \tag{7}$$

where $\boldsymbol{M} \in \mathbb{R}^{T\times T}$ is a bidiagonal matrix, $\boldsymbol{MC} \geq \boldsymbol{0}$ indicates that every element of $\boldsymbol{MC}$ is greater than or equal to zero, $^\mathsf{T}$ indicates transpose, $\boldsymbol{1}$ is a T dimensional column vector, $\boldsymbol{\lambda} = \lambda\Delta\boldsymbol{1}^\mathsf{T}$, and $\log(\cdot)$ indicates an element-wise logarithm. Note that Eq. (7) follows from writing $\boldsymbol{n}$ in terms of $\boldsymbol{M}$ and $\boldsymbol{C}$:

$$\boldsymbol{MC} = \begin{bmatrix} 1 & 0 & 0 & \cdots & \cdots \\ 1 & -\gamma & 0 & \cdots & \cdots \\ 0 & 1 & -\gamma & 0 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 1 & -\gamma \end{bmatrix} \begin{bmatrix} C_1 \\ C_2 \\ \vdots \\ \vdots \\ C_T \end{bmatrix} = \begin{bmatrix} n_1 \\ n_2 \\ \vdots \\ \vdots \\ n_T \end{bmatrix} = \boldsymbol{n} \tag{8}$$

Thus, a little bit of calculus yields our update algorithm for $\boldsymbol{C}_z$:

$$\widehat{\boldsymbol{C}}_z \leftarrow \widehat{\boldsymbol{C}}_z + s\boldsymbol{d} \tag{9a}$$

$$\boldsymbol{H}\boldsymbol{d} = \boldsymbol{g} \tag{9b}$$

$$\boldsymbol{g} = -\frac{\alpha}{\sigma^2}(\boldsymbol{F} - \alpha(\widehat{\boldsymbol{C}}_z^\mathsf{T} + \beta)) + \boldsymbol{M}^\mathsf{T}\boldsymbol{\lambda} - z\boldsymbol{M}^\mathsf{T}(\boldsymbol{M}\widehat{\boldsymbol{C}}_z)^{-1} \tag{9c}$$

$$\boldsymbol{H} = \frac{\alpha^2}{\sigma^2}\boldsymbol{I} + z\boldsymbol{M}^\mathsf{T}(\boldsymbol{M}\widehat{\boldsymbol{C}}_z)^{-2}\boldsymbol{M} \tag{9d}$$

where s is the step size, $\boldsymbol{d}$ is the step direction, and $\boldsymbol{g}$ and $\boldsymbol{H}$ are the gradient (first derivative) and Hessian (second derivative) of the argument in Eq. (7) with respect to $\boldsymbol{C}$, respectively, and the exponents indicate element-wise operations. Note that we use "backtracking linesearches", meaning that for each iteration, we find the maximal s that is (i) between 0 and 1 and (ii) decreases the likelihood.

Typically, implementing Newton-Raphson requires inverting the Hessian, i.e., $\boldsymbol{d} = \boldsymbol{H}^{-1}\boldsymbol{g}$, a computation consuming $O(T^3)$ time. Here, because $\boldsymbol{M}$ is bidiagonal, the Hessian is tridiagonal, so the solution may be found in $O(T)$ time via standard banded Gaussian elimination techniques (which can be implemented efficiently in Matlab using $\boldsymbol{H}\backslash\boldsymbol{g}$). Thus, by recursively solving for $\widehat{\boldsymbol{C}}_z$ using Eq. (9), we obtain $\widehat{\boldsymbol{n}}$, which is an approximation to to $\boldsymbol{n}_{MAP}$. We refer to this fast algorithm for solving (4) the FAst Nonnegative Deconvolution (FAND) filter.

3 Results

3.1 Main Result

The main result of this paper is that we can approximately compute $\boldsymbol{n}_{MAP}$ in *linear* time, whereas an exact solution would require exponential time. Fig. 1 shows an example of running the FANSI filter on simulated data. The top three panels show bF, $\boldsymbol{C}$, and $\boldsymbol{n}$, simulated according to our model, Eqs. (1) and (2). The bottom panel shows the output of our FANSI filter. Note that the FANSI filter does not provide the most likely spike train, $\boldsymbol{n}_{MAP}$, but rather an approximation to it, $\widetilde{\boldsymbol{n}}$ (see Eq. (4) for definition), that is computable exactly in *linear* time.

3.1.1 Comparison with Wiener filter

According to the above model (Eq. (2)), the neuron spikes according to a Poisson distribution, which is well approximated by an exponential distribution, when the rate is low. However, when the rate is high, e.g., several spikes per time bin, the distribution of spikes would be better approximated by a Gaussian distribution. The optimal linear filter, upon making this assumption, is typically called a Wiener filter [26].[2] Importantly, the Wiener admits negative spikes as viable solutions, as a Gaussian distribution has support on the entire real number line (our FANSI filter, however, imposes a non-negative constraint). Thus, a natural question to ask is how our FANSI filter compares with the Wiener filter on both slow and fast spike trains. Figure 2 depicts the performance of both the FANSI filter and Wiener filter for these two scenarios: the top panels show the fluorescence traces provided to the two filters, the middle panel shows the outputs of the FANSI filter, and the bottom panel shows the outputs of the Wiener filter.

On the left, it is clear that the FANSI filter significantly outperforms the Wiener filter, in terms of signal-to-noise ratio (SNR). Specifically, the Wiener filter infers *negative* spikes throughout the time course, resulting in the so-called "ringing" effect. Intuitively, the ringing results from the most likely explanation of a strong downward shift in fluorescence is a corresponding downward shift in spiking. The FANSI filter prohibits downward spiking events (with the non-negative constraint), and therefore, does not suffer from ringing.

On the right, both filters seem to perform very well, suggesting that even when the true spike distribution is better approximated by a Gaussian distribution, the exponential distribution is sufficient. Because the Wiener filter naïvely requires $O(T \log T)$, whereas the FANSI filter requires only $O(T)$, this suggests that the FANSI filter performs both (i) at least as well, and (ii) faster than the Wiener filter.

[2] See Appexdix A for a derivation of the Wiener filter for this model.

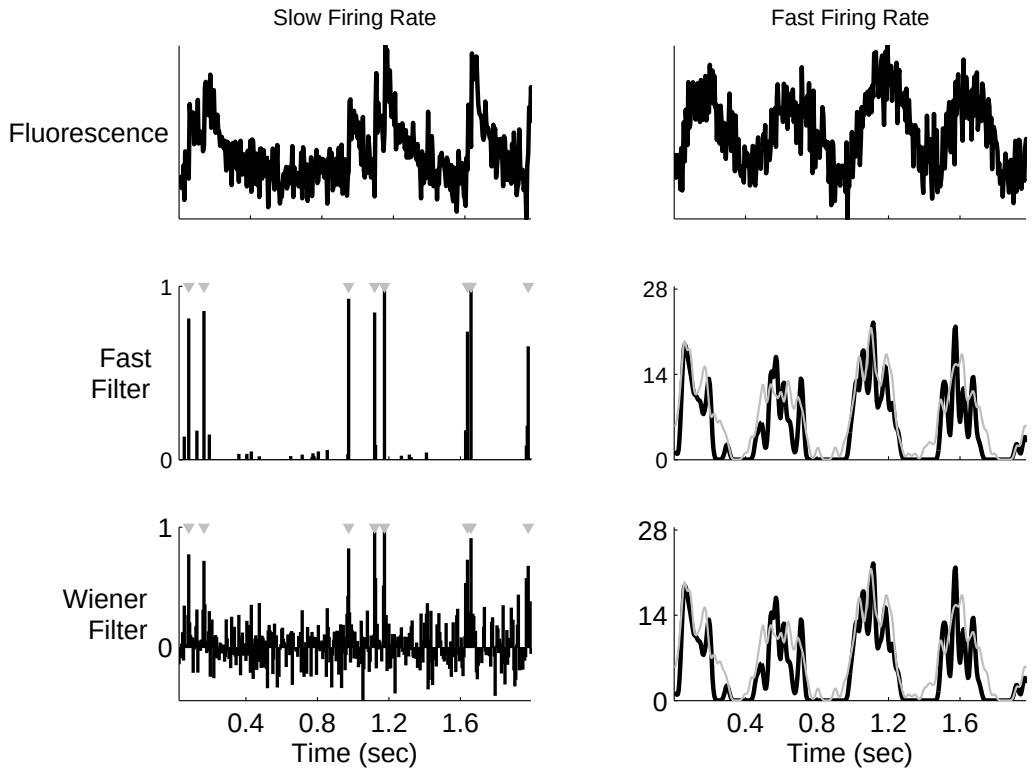

Figure 2: A simulation demonstrating that the FANSI filter performs at least as well as the optimal linear (i.e., Wiener) filter. The left panels show that in the slow firing regime, the FANSI filter outperforms the Wiener filter in terms of SNR. This "makes sense", given that if a neuron is spiking according to a Poisson distribution with a slow rate, an exponential distribution is a better approximation than a Gaussian distribution, and the FANSI filter approximates the spike train distribution with an exponential, versus the Wiener filter's Gaussian. The right panels show that both approximations are sufficient in the fast firing regime. Top left panel: fluorescence time series for a neuron with a slow firing rate. Middle left panel: the FANSI filter's inferred spike train. Bottom left panel: Wiener filter's inferred spike train. Note that (i) the Wiener filter does not impose a non-negativity constraint, and (ii) the effective SNR of the Wiener filter in this example is worse than the FANSI filter's. Top right panel: same as top left panel, for a neuron with a high firing rate. Middle right panel: the FANSI filter's inferred spike train smoothed with a Gaussian kernel for visualization purposes (black line), and the true spike train smoothed with the same Gaussian kernel (gray line). Bottom right panel: same as middle right panel, but with the Wiener filter. Parameters for left panels: same as above. Parameters for right panels: same as above, except: $\sigma = 8$ photons, $\lambda = 500$ Hz.

3.1.2 Spatial Filtering

In the previous sections, we implicitly assumed that the raw movie of fluorescence measurements collected by the experimenter had undergone two stages of pre-processing. First, the movie was segmented, to determine regions-of-interest (ROIs). This yields a vector, $\vec{F}_t$, corresponding to the fluorescence intensity at time t for each of the N_p pixels in the ROI. Second, we projected that vector into a scalar, yielding F_t, the assumed input. In this section, we still assume that somebody has gone through our movies and performed some segmentation, but we do not assume that they have projected the vector $\vec{F}_t$ into a scalar F_t. Formally, we posit a more general model:

$$\vec{F}_t = \vec{\alpha}(C_t + \beta) + \sigma\vec{\varepsilon}_t, \qquad \vec{\varepsilon}_t \sim \mathcal{N}(\vec{0}, \boldsymbol{I}) \tag{10}$$

where $\vec{F}_t$, $\vec{\alpha}$, $\vec{\varepsilon}_t$, and $\vec{0}$ are all column vectors of length N_p, and $\boldsymbol{I}$ is an $N_p \times N_p$ identity matrix. This model follows from the observation that the fluorescence at any individual pixel is composed of a static element, β, and a dynamic element, that we assume is purely due to calcium fluctuations, C_t. Further, we have assumed that the noise is uncorrelated and has the same variance, σ^2, in each pixel (an assumption that can be relaxed quite easily). Performing inference in this more general model proceeds nearly identical as before,

$$\widehat{\boldsymbol{C}}_z = \underset{\boldsymbol{MC}\geq\boldsymbol{0}}{\operatorname{argmin}} \frac{1}{2\sigma^2} \left\| \vec{\boldsymbol{F}} - \vec{\alpha}(\boldsymbol{C}^\mathsf{T} + \beta\boldsymbol{1}^\mathsf{T}) \right\|^2 + (\boldsymbol{MC})^\mathsf{T}\boldsymbol{\lambda} - z\log(\boldsymbol{MC})^\mathsf{T}\boldsymbol{1}, \tag{11}$$

$$\boldsymbol{g} = -\frac{\boldsymbol{\alpha}}{\sigma^2}(\boldsymbol{F} - \boldsymbol{\alpha}(\widehat{\boldsymbol{C}}_z^\mathsf{T} + \beta)) + \boldsymbol{M}^\mathsf{T}\boldsymbol{\lambda} - z\boldsymbol{M}^\mathsf{T}(\boldsymbol{M}\widehat{\boldsymbol{C}}_z)^{-1} \tag{12}$$

$$\boldsymbol{H} = \frac{\boldsymbol{\alpha}^\mathsf{T}\boldsymbol{\alpha}}{\sigma^2}\boldsymbol{I} + z\boldsymbol{M}^\mathsf{T}(\boldsymbol{M}\widehat{\boldsymbol{C}}_z)^{-2}\boldsymbol{M} \tag{13}$$

Figure 3 demonstrates the utility of this generalization. The top row shows different depictions of an ROI containing a single neuron, all using the same color scale. On the far left panel is the "true" spatial filter. Importantly, some pixels are *anti-correlated* with others, as indicated by certain pixels being black, and others white. Such a scenario can often arise in both in vitro and in vivo recordings, as the concentration of calcium immediately outside the cell often decreases after a spike, as the calcium ions flow into the cell. Unsurprisingly, the mean frame looks very similar to the true spatial filter, as individual frames are effectively just modulating the magnitude of the spatial filter, and adding noise. Typically, one would simply identify the pixels with high positive values from the mean frame, and average them together. Such an approach yields the 1-dimensional fluorescence projection depicted on the left middle panel, with its associated inferred spike train beneath. Using the true spatial filter to project $\vec{\boldsymbol{F}}$ onto a 1-dimensional fluorescence time series results in the middle right panel, with its associated inferred spike train beneath. It should be clear that using the true spatial filter improves the SNR of the fluorescence signal, and therefore, the inferred spike train accuracy.

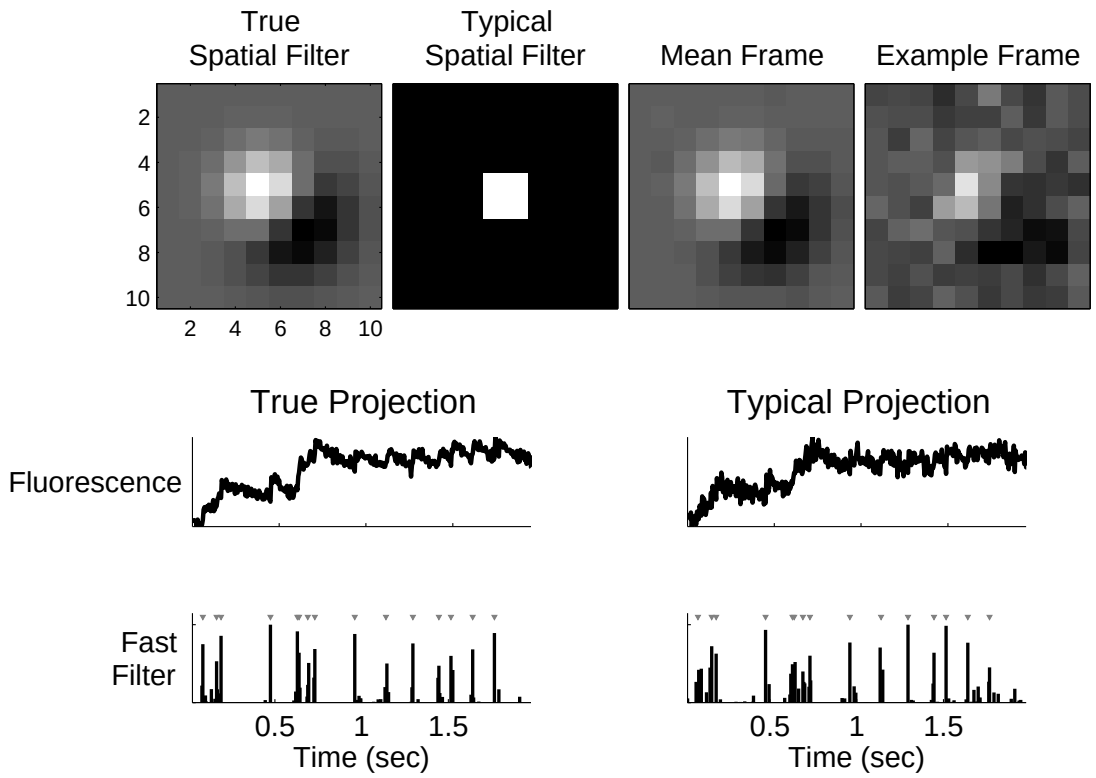

Figure 3: A simulation demonstrating that using a better spatial filter can significantly enhance the effective SNR (see Supplementary Movie 1 for the full movie associated with this simulation). Top: four movie frames. Middle left: projection of the entire movie onto the optimal spatial filter, yielding a 1D fluorescence time series, with a large SNR. Bottom left: FANSI filter's inferred spike train using the optimal spatial filter. Middle right: projection of the entire movie onto the "mean" spatial filter, yielding a 1D fluorescence time series with a small SNR. Bottom right: FANSI filter's inferred spike train using the mean spatial filter. Parameters different from Fig 1: $\boldsymbol{\alpha}$ is a mixture of two 2D-Gaussians, with mean $\mu_1 = \mu_2 = [0, 0]$ and covariance matrices, $\Sigma_i = \sigma_i^2 \boldsymbol{I}$, with $\sigma_1 = 2$ and $\sigma_2 = 4$ and $\boldsymbol{I}$ is the identity matrix. $\boldsymbol{\beta} = 1$.

3.2 Learning

In the above, we assumed that the parameters governing our model, $\boldsymbol{\theta} = \{\boldsymbol{\alpha}, \beta, \sigma, \gamma, \lambda\} \in \boldsymbol{\Theta}$, were known. In general, however, these parameters must be estimated from the data. To find the maximum likelihood estimator for the parameters, $\widehat{\boldsymbol{\theta}}$, we must integrate over all possible spike trains. Unfortunately, it is not currently known how to perform this integral exactly, and approximating this integral using Monte Carlo methods is relatively time consuming (see [16] for details). Thus, we resort to a more drastic approximation, commonly used in state-space models. For specifically, instead of integrating over all possibly spike trains, we only consider the most likely sequence (ofter referred to as the Viterbi path [27]):

$$\widehat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta}\in\boldsymbol{\Theta}}{\operatorname{argmax}} \int P[\boldsymbol{F}|\boldsymbol{C},\boldsymbol{\theta}]P[\boldsymbol{C}|\boldsymbol{\theta}]d\boldsymbol{C} \approx \underset{\boldsymbol{\theta}\in\boldsymbol{\Theta}}{\operatorname{argmax}} P[\boldsymbol{F}|\widehat{\boldsymbol{C}},\boldsymbol{\theta}]P[\widehat{\boldsymbol{C}},|\boldsymbol{\theta}] \tag{14}$$

where $\widehat{\boldsymbol{C}}$ is determined using the above described inference algorithm. The approximation in (14) is good whenever the likelihood is very peaky, meaning that most of the mass is around the MAP sequence.[3]

Due to the state space nature of the above model (Eqs (10) and (2)), the optimization in Eq (14) simplifies significantly. More specifically, we can write the argument from the right-hand-side of Eq (14) as a product of terms that we have defined in our model:

$$P[\boldsymbol{F}|\widehat{\boldsymbol{C}},\boldsymbol{\theta}]P[\widehat{\boldsymbol{C}}|\boldsymbol{\theta}] = \prod_{t=1}^{T} P[F_t|\widehat{C}_t,\boldsymbol{\alpha},\beta,\sigma]P[\widehat{C}_t|\widehat{C}_{t-1},\widehat{n}_t,\gamma]P[\widehat{n}_t|\lambda] \tag{15}$$

Fortunately, this optimization simplifies into several separable problems: (1) $\{\boldsymbol{\alpha}, \beta\}$, (2) σ, (3) γ, and (4) λ.

Because solving for $\boldsymbol{\alpha}$ and β jointly is non-concave, we solve for each separately. First consider $\boldsymbol{\alpha}$ only:

$$\widehat{\boldsymbol{\alpha}} = \underset{\boldsymbol{\alpha}}{\operatorname{argmax}} \prod_{t,x=1}^{T,N_p} P_{\boldsymbol{\theta}}[F_{t,x}|C_t] = \underset{\boldsymbol{\alpha}}{\operatorname{argmax}} \prod_{t,x=1}^{T,N_p} \mathcal{N}(F_{x,t}; \alpha_x(C_t+\beta), \sigma^2) \tag{16a}$$

$$= \underset{\boldsymbol{\alpha}}{\operatorname{argmax}} \sum_{t,x=1}^{T,N_p} \log \mathcal{N}(F_{x,t}; \alpha_x(C_t+\beta), \sigma^2) \tag{16b}$$

$$= \underset{\boldsymbol{\alpha}}{\operatorname{argmax}} -\frac{N_p T}{2}\log(2\pi\sigma^2) - \frac{1}{2\sigma^2}\sum_{t,x=1}^{T,N} (F_{t,x} - \alpha_x(C_t+b))^2 \tag{16c}$$

$$= \underset{\boldsymbol{\alpha}}{\operatorname{argmin}} \sum_{t,x=1}^{T,N_p} (F_{t,x} - \alpha_x(C_t+\beta))^2. \tag{16d}$$

Therefore, we can solve for each of the N_p $\widehat{\alpha}_x$'s separately and efficiently using Matlab's `mldivide`,

$$\widehat{\alpha}_x = (\boldsymbol{C} + \beta\mathbf{1}) \backslash \boldsymbol{F}_x, \tag{17}$$

where $\boldsymbol{F}_x = [F_{1,x}, \ldots, F_{T,x}]^{\mathsf{T}}$. Given $\widehat{\boldsymbol{\alpha}}$, we can estimate β:

$$\widehat{\beta} = \underset{\beta>0}{\operatorname{argmax}} \sum_{t,x=1}^{T,N} (F_{t,x} - \widehat{\alpha}_x(C_t+\beta))^2 = \underset{\beta>0}{\operatorname{argmax}} \sum_{t,x=1}^{T,N} (F_{t,x} - \widehat{\alpha}_x C_t + \beta\widehat{\alpha}_x)^2 \tag{18}$$

which can be solved efficiently again using Matlab's `mldivide`: $\widetilde{\beta} = \widehat{\alpha}_x \backslash \left(\sum_{t=1}^{T} F_{t,x} - \widehat{\alpha}_x C_t\right)$. If $\widetilde{\beta} < 0$, we simply let $\widehat{\beta} = 0$, else, $\widehat{\beta} = \widetilde{\beta}$.

[3] The approximation in (14) may be considered a first-order Laplace approximation

Given $\widehat{\boldsymbol{\alpha}}$ and $\widehat{\beta}$, we can now estimate σ using the residuals, i.e.,

$$\widehat{\sigma}^2 = \frac{1}{TN_p} \left\| \vec{\boldsymbol{F}} - \widehat{\boldsymbol{\alpha}}(\boldsymbol{C}^{\mathsf{T}} + \beta \mathbf{0}) \right\|^2 \tag{19}$$

Alternately, we could simply find a segment of time in which it is clear that no spikes occurred, $\vec{F}_{s:t}$, and then let $\widehat{\sigma} = \mathrm{Var}(\vec{F}_{s:t})$, which is independent of our inference algorithm.

Although estimating γ is rather straightforward, previous work [11] and unpublished results suggest that errors in estimating γ do not significantly impact inference quality, and that deviations in γ are relatively small across a population of neurons, so it can typically be "eyeballed".

Estimating λ is also totally straightforward:

$$\widehat{\lambda} = \frac{1}{T\Delta} \widehat{\boldsymbol{n}}' \mathbf{1} \tag{20}$$

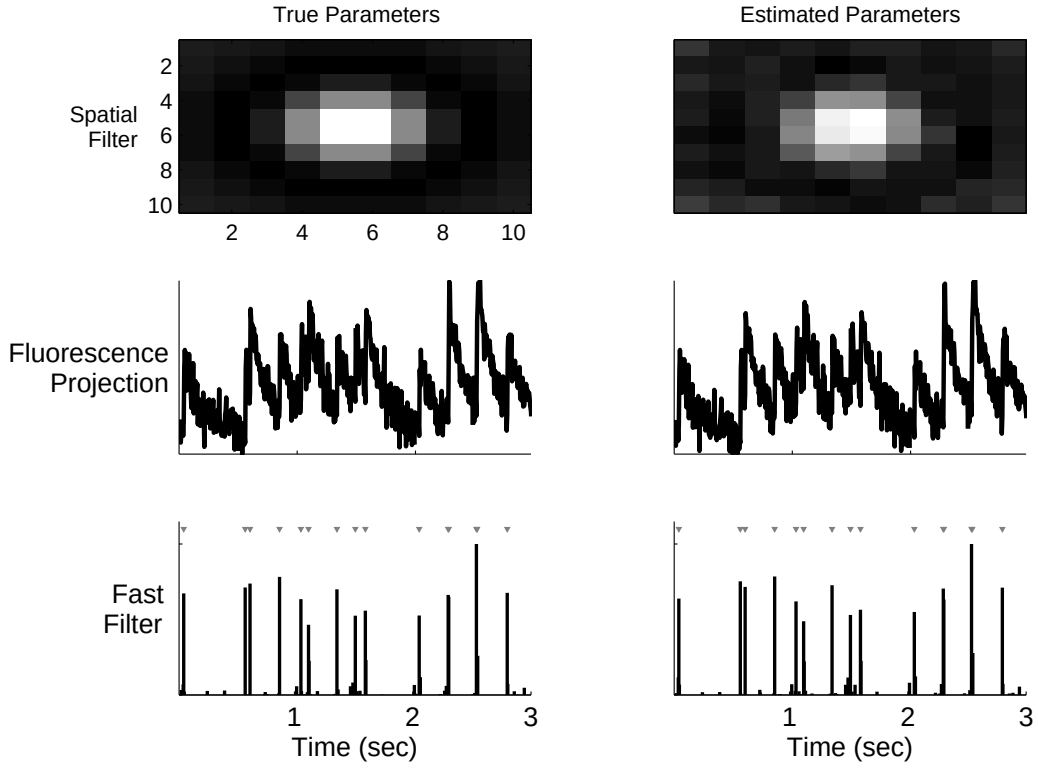

Figure 4: A simulation demonstrating that given only the fluorescence movie, the parameters may be estimated, and the spike train inferred (c.f. Supplementary Movie 2). Top left panel: true spatial filter. Middle left panel: projection of movie onto true spatial filter. Bottom left panel: inferred spike train using true parameters. Right panels: same as left except estimating parameters. $\boldsymbol{\alpha}$ initialized with the first singular value of $\boldsymbol{F}$. $\boldsymbol{\beta}$ initialized with $\mathbf{0}$. λ and σ were initialized at double their true value. τ was assumed known. Parameters same as above.

3.3 in vitro data

3.4 Overlapping spatial filters

Model

$$\boldsymbol{F}_t = \sum_{i=1}^{N_c} \boldsymbol{\alpha}_i (C_{i,t} + \beta_i) + \boldsymbol{\varepsilon}_t, \qquad \boldsymbol{\varepsilon}_t \sim \mathcal{N}(\mathbf{0}, \sigma^2 \boldsymbol{I}) \tag{21}$$

$$C_{i,t} = \gamma_i C_{i,t-1} + n_{i,t}, \qquad n_{i,t} \sim \text{Poisson}(n_{i,t}; \lambda_i \Delta) \tag{22}$$

implicit assumption that $\boldsymbol{n}_i \perp \boldsymbol{n}_j$, $\forall i \neq j$

Inference let $\boldsymbol{n} = (n_{1,1}, n_{2,1}, \ldots, n_{N_c,1}, n_{1,2}, \ldots, n_{N_c,T})^\mathsf{T}$. similar def of $\boldsymbol{C}$.

$$\boldsymbol{M} = \begin{bmatrix} 1 & 1 & 0 & 0 & \cdots & \cdots & & & & \\ 1 & -\gamma_1 & & 1 & -\gamma_2 & \ldots & 1 & -\gamma_{N_c} & 0 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & & & & & \\ 0 & 0 & 0 \ldots & 1 & -\gamma_{N_c-1} & 1 & -\gamma_{N_c} & & & \end{bmatrix} \tag{23}$$

inference as before, but replacing the scalar β with $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_{N_c})^\mathsf{T}$, and making minor adjustments to deal with dimensionality issues.

Learning estimating $\boldsymbol{\alpha} = (\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_{N_c})^\mathsf{T}$ is similar. now, we compute $\widehat{\boldsymbol{\alpha}}_x = (\widehat{\alpha}_{1,x}, \ldots, \widehat{\alpha}_{N_c,x})^\mathsf{T}$ using

$$\widehat{\boldsymbol{\alpha}}_x = (\boldsymbol{C} + \widetilde{\boldsymbol{\beta}}) \backslash \boldsymbol{F}_x, \tag{24}$$

where $\widetilde{\boldsymbol{\beta}}$ is $\boldsymbol{\beta}$ reparameterized to be the same size as $\boldsymbol{C}$.

estimating $\boldsymbol{\beta}$ proceeds as before, but since it is a vector, we use Matlab's `quadprog`, imposing the constraint that $\beta_i > 0 \, \forall i$.

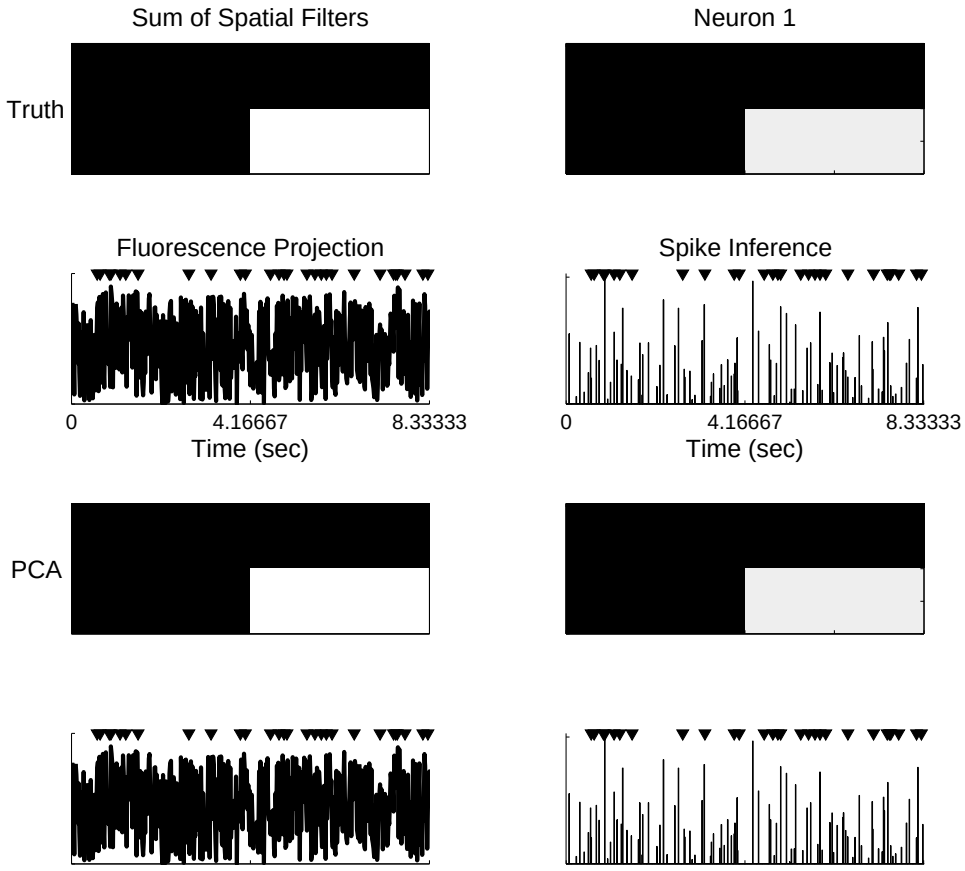

Figure 5: Simulation showing that even when two neuron's spatial filters are largely overlapping, our inferencsame as spatial EM, but with 2 cells in ROI

3.5 Population imaging

Figure 6: full movie, fully automated

Figure 7: full movie, fully automated, real data

3.6 Slow rise time

3.7 Poisson observation model

Model

$$\boldsymbol{F}_{x,t} \sim \text{Poisson}(\alpha_x(C_t + \beta)) \tag{25}$$

Inference

$$\mathcal{L}_{x,t} = -\alpha_x(C_t + \beta) + F_{x,t}\log(\alpha_x(C_t + \beta)) - \log(F_{x,t}!) \tag{26a}$$
$$g_{x,t} = -\alpha_x + F_{x,t}(C_t + \beta)^{-1} \tag{26b}$$
$$H_{x,t} = -F_{x,t}(C_t + \beta)^{-2} \tag{26c}$$

where $\mathcal{L} = \sum_{x,t} \mathcal{L}_{x,t}$, $\boldsymbol{g} = \sum_x (g_{x,1}, \ldots, g_{x,T})^\mathsf{T}$, and $\boldsymbol{H} = \text{diag}(\sum_x H_{x,t})$.

3.8 Incorporating a nonlinear observation model

$$\boldsymbol{F}_t = \boldsymbol{\alpha} S(C_t + \beta) + \boldsymbol{\varepsilon}_t, \qquad \boldsymbol{\varepsilon}_t \sim \mathcal{N}(\mathbf{0}, \sigma^2 \boldsymbol{I}) \tag{27}$$

where $S(x) = \frac{x^{n_d}}{x^{n_d} + k_d}$

note: initialize with linear result, but add a constant wherever constraint is not satisfied

3.9 in vivo data

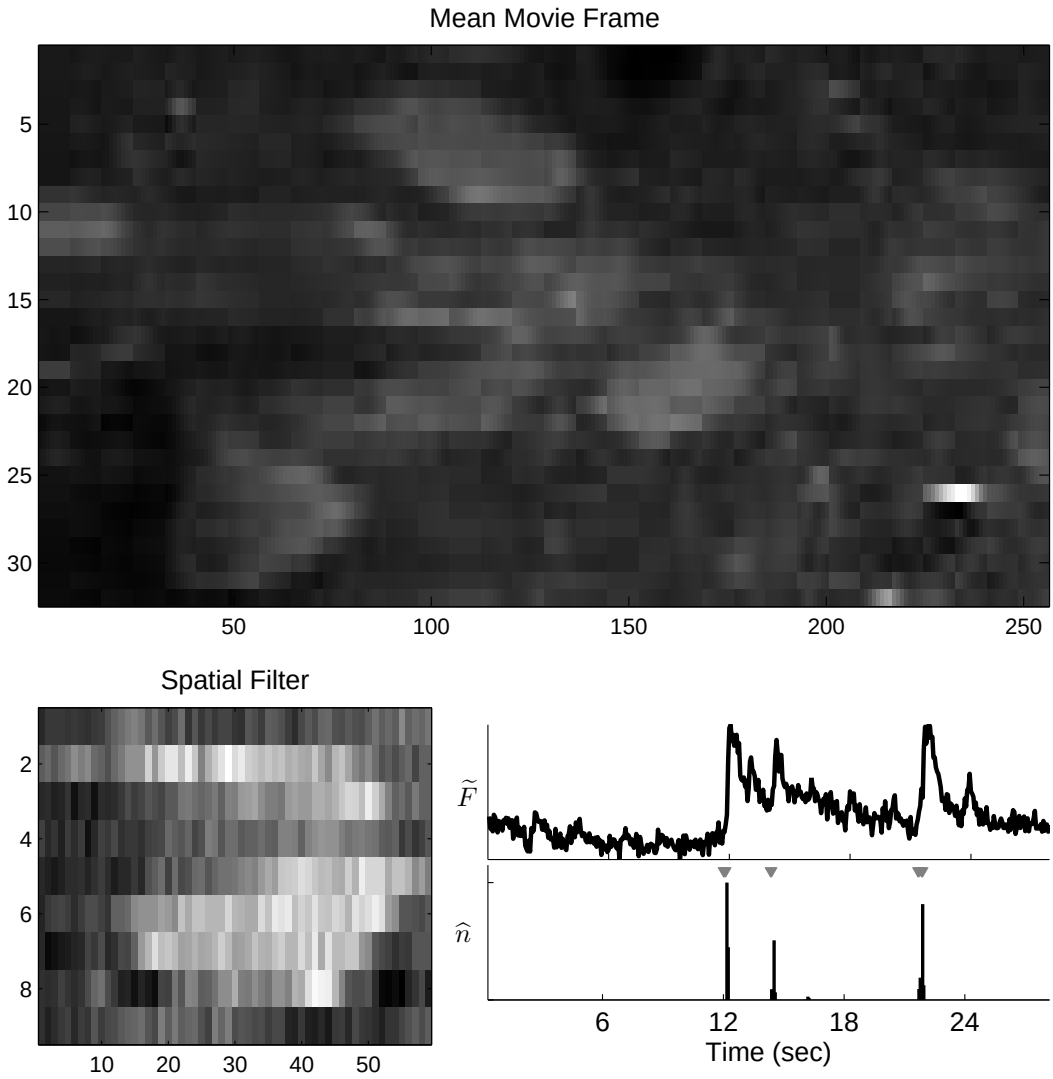

Figure 8: Given only a fluorescence movie, recorded in vivo, we can learn the parameters necessary to correctly infer the spike trains. Left: mean frame. Left: projection of movie onto mean frame. Left: the FANSI filter's inference.

Figure 9: distribution of errors in spike inference from real data

4 Discussion

Summary We show here that for certain nonnegative deconvolution problems, we can derive an algorithm that is both optimal and efficient. More specifically, our algorithm may be applied to any model with a nonnegative signal that is linearly filtered by a matrix linear ordinary differential equation. We apply this approach to the problem of inferring the most likely spike train given noisy calcium sensitive fluorescence observations (c.f. Fig. 1), and demonstrate, in simulations, that the optimal nonnegative filter outperforms the optimal linear (i.e., Wiener) filter in both slow and fast firing rate regimes (c.f. Fig. 2). Furthermore, when applied to data from a live cell, the optimal nonnegative filter outperforms a fast projection pursuit regression filter, which constrains the inferred spike train to be nonnegative integers (c.f. Fig. 8). On the other hand, the nonnegative filter is based on a linear observation model, and therefore suffers a loss of precision in the presence of strong saturation effects, in contrast to the optimal nonlinear particle filter (c.f. Fig. 8).

The implications of these results are severalfold. First, it seems as if there is no reason to use the Wiener filter for scenarios in which our algorithm may apply. Second, as our filter is so efficient, it may be used for many real-time processing applications. Specifically, upon simultaneously imaging a population of neurons [28, 29, 9, 30, 12], our filter may be applied essentially online. This could greatly expedite the tuning of important experimental parameters — such as laser intensity — to optimize signal-to-noise ratio for inferring spikes. Third, the parameters estimated from this filter may be used to initialize the parameters of the optimal nonlinear particle filter, which may then be used offline, to further refine the spike train inference.

Extensions

Thresholding

4.1 Dynamic prior

Model let $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_T)^\mathsf{T}$

$$C_t = \gamma C_{t-1} + n_t, \qquad n_t \sim \text{Poisson}(n_t; \lambda_t \Delta) \tag{28}$$

Inference

$$\mathcal{L} = \frac{1}{2\sigma^2} \left\| \boldsymbol{F} - \boldsymbol{\alpha}(\boldsymbol{C}^\mathsf{T} + \beta + \mathbf{1}^\mathsf{T}) \right\|^2 + (\boldsymbol{MC})^\mathsf{T} \boldsymbol{\lambda} \Delta - z \log(\boldsymbol{MC})^\mathsf{T} \mathbf{1} \tag{29a}$$

$$\boldsymbol{g} = -\frac{\boldsymbol{\alpha}}{\sigma^2}(\boldsymbol{F} - \boldsymbol{\alpha}(\widehat{\boldsymbol{C}}_z^\mathsf{T} + \beta)) + \boldsymbol{M}^\mathsf{T} \boldsymbol{\lambda} \Delta - z \boldsymbol{M}^\mathsf{T} (\boldsymbol{M} \widehat{\boldsymbol{C}}_z)^{-1} \tag{29b}$$

Acknowledgments Support for JTV was provided by NIDCD DC00109. LP is supported by an NSF CAREER award, by an Alfred P. Sloan Research Fellowship, and the McKnight Scholar Award. BOW was supported by NDS grant F30 NS051964. The authors would like to thank A. Packer for helpful discussions.

References

[1] R. Yuste and A. Konnerth. *Imaging in Neuroscience and Development, A Laboratory Manual*, 2006.

[2] Shin Nagayama, Shaoqun Zeng, Wenhui Xiong, Max L Fletcher, Arjun V Masurkar, Douglas J Davis, Vincent A Pieribone, and Wei R Chen. In vivo simultaneous tracing and Ca^{2+} imaging of local neuronal circuits. *Neuron*, 53(6):789–803, Mar 2007.

[3] Werner Göbel and Fritjof Helmchen. In vivo calcium imaging of neural network function. *Physiology (Bethesda)*, 22:358–365, Dec 2007.

[4] Liqun Luo, Edward M Callaway, and Karel Svoboda. Genetic dissection of neural circuits. *Neuron*, 57(5):634–660, Mar 2008.

[5] Olga Garaschuk, Oliver Griesbeck, and Arthur Konnerth. Troponin c-based biosensors: a new family of genetically encoded indicators for in vivo calcium imaging in the nervous system. *Cell Calcium*, 42(4-5):351–361, 2007.

[6] Marco Mank and Oliver Griesbeck. Genetically encoded calcium indicators. *Chem Rev*, 108(5):1550–1564, May 2008.

[7] Damian J Wallace, Stephan Meyer zum Alten Borgloh, Simone Astori, Ying Yang, Melanie Bausen, Sebastian Kgler, Amy E Palmer, Roger Y Tsien, Rolf Sprengel, Jason N D Kerr, Winfried Denk, and Mazahir T Hasan. Single-spike detection in vitro and in vivo with a genetic ca2+ sensor. *Nat Methods*, 5(9):797–804, Sep 2008.

[8] Christoph Stosiek, Olga Garaschuk, Knut Holthoff, and Arthur Konnerth. In vivo two-photon calcium imaging of neuronal networks. *Proceedings of The National Academy Of Sciences Of The United States Of America*, 100(12):7319–7324, Jun 2003.

[9] Kenichi Ohki, Sooyoung Chung, Yeang H Ch'ng, Prakash Kara, and R Clay Reid. Functional imaging with cellular resolution reveals precise micro-architecture in visual cortex. *Nature*, 433(7026):597–603, Feb 2005.

[10] Kenichi Ohki, Sooyoung Chung, Prakash Kara, Mark H??bener, Tobias Bonhoeffer, and R. Clay Reid. Highly ordered arrangement of single neurons in orientation pinwheels. *Nature*, 442(7105):925–928, Aug 2006.

[11] Emre Yaksi, Benjamin Judkewitz, and Rainer W Friedrich. Topological reorganization of odor representations in the olfactory bulb. *PLoS Biol*, 5(7):e178, Jul 2007.

[12] Takashi R Sato, Noah W Gray, Zachary F Mainen, and Karel Svoboda. The functional microarchitecture of the mouse barrel cortex. *PLoS Biol*, 5(7):e189, Jul 2007.

[13] J.N.D. Kerr, C.P.J. de Kock, D.S. Greenberg, R.M. Bruno, B. Sakmann, and F. Helmchen. Spatial organization of neuronal population responses in layer 2/3 of rat barrel cortex. *Journal of Neuroscience*, 27(48):13316, 2007.

[14] Ilker Ozden, H. Megan Lee, Megan R Sullivan, and Samuel S-H Wang. Identification and clustering of event patterns from in vivo multiphoton optical recordings of neuronal ensembles. *J Neurophysiol*, 100(1):495–503, Jul 2008.

[15] David S Greenberg, Arthur R Houweling, and Jason N D Kerr. Population imaging of ongoing neuronal activity in the visual cortex of awake rats. *Nat Neurosci*, 11(7):749 – 751, Jun 2008.

[16] JT Vogelstein, BO Watson, Packer AM, R Yuste, Jedynak B, and L Paninski. Spike inference from calcium imaging using sequential monte carlo methods. *Biophysical Journal*, 2009.

[17] Terrence F Holekamp, Diwakar Turaga, and Timothy E Holy. Fast three-dimensional fluorescence imaging of activity in neural populations by objective-coupled planar illumination microscopy. *Neuron*, 57(5):661–672, Mar 2008.

[18] D. D. Lee and H. S. Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788–791, Oct 1999.

[19] D.D. Lee and H.S. Seung. Algorithms for non-negative matrix factorization. *Advances in Neural Information Processing Systems*, 13:556–562, 2001.

[20] Quentin J M Huys, Misha B Ahrens, and Liam Paninski. Efficient estimation of detailed single-neuron models. *J Neurophysiol*, 96(2):872–890, Aug 2006.

[21] L.F. Portugal, J.J. Judice, and L.N. Vicente. A comparison of block pivoting and interior-point algorithms for linear least squares problems with nonnegative variables. *Mathematics of Computation*, 63(208):625–643, 1994.

[22] H. Lee, A. Battle, R. Raina, and A. Ng. Efficient sparse coding algorithms. *Advances in Neural Information Processing Systems*, 2006.

[23] O?Grady, P.D. and Pearlmutter, B.A. Convolutive non-negative matrix factorisation with a sparseness constraint. *Machine Learning for Signal Processing, 2006. Proceedings of the 2006 16th IEEE Signal Processing Society Workshop on*, pages 427–432, 2006.

[24] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, 2001.

[25] S. Boyd and L. Vandenberghe. *Convex Optimization*. Oxford University Press, 2004.

[26] N. Wiener. *Extrapolation, interpolation, and smoothing of stationary time series: with engineering applications*. MIT Press, Cambridge, Mass., 1949.

[27] Lawrence R Rabiner. A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. *Proceedings of the IEEE*, 72(2):257–286, February 1989.

[28] Yuji Ikegaya, Gloster Aaron, Rosa Cossart, Dmitriy Aronov, Ilan Lampl, David Ferster, and Rafael Yuste. Synfire chains and cortical songs: temporal modules of cortical activity. *Science*, 304(5670):559–564, Apr 2004.

[29] Cristopher M Niell and Stephen J Smith. Functional imaging reveals rapid development of visual response properties in the zebrafish tectum. *Neuron*, 45(6):941–951, Mar 2005.

[30] Emre Yaksi and Rainer W Friedrich. Reconstruction of firing rate changes across neuronal populations by temporally deconvolved Ca^{2+} imaging. *Nat Methods*, 3(5):377–383, May 2006.

A Wiener Filter

Sections 2.2 outline one approach to solving Eq. (3), by approximating the Poisson distribution with an exponential distribution, and imposing a non-negative constraint on the inferred $\widehat{\boldsymbol{n}}$. Perhaps a more straightforward approach would be to approximate the Poission distribution with a Gaussian distribution. In fact, as rate increases above about 10 spikes/sec, a Poisson distribution with rate $\lambda\Delta$ is well approximated by a Gaussian with mean and variance $\lambda\Delta$. Given such an approximation, instead of Eq. (4), we would obtain:

$$\widehat{\boldsymbol{n}}_w \approx \underset{n_t \in \mathbb{R}, \forall t}{\operatorname{argmin}} \sum_{t=1}^{T} \left(\frac{1}{2\sigma^2}(F_t - C_t)^2 + \frac{1}{2\lambda\Delta}(n_t - \lambda\Delta)^2 \right) \tag{30}$$

As above, we can rewrite Eq. (30) in matrix notation in terms of $\boldsymbol{C}$:

$$\widehat{\boldsymbol{C}}_w = \underset{C_t \in \mathbb{R}, \forall t}{\operatorname{argmin}} \frac{1}{2\sigma^2} \left\| \boldsymbol{F} - \boldsymbol{C} \right\|^2 + \frac{1}{2\lambda\Delta} \left\| \boldsymbol{M}\boldsymbol{C} - \lambda\Delta\boldsymbol{1} \right\|^2 \tag{31}$$

which is quadratic in $\boldsymbol{C}$, and may therefore be solved analytically using quadratic programming, $\widehat{\boldsymbol{C}}_w = \widehat{\boldsymbol{C}}_0 + \boldsymbol{d}_w$, where $\widehat{\boldsymbol{C}}_0$ is the initial guess and $\boldsymbol{d}_w = \boldsymbol{H}_w \backslash \boldsymbol{g}_w$, where

$$\boldsymbol{g}_w = \frac{1}{\sigma^2}(\boldsymbol{C}_0' - \boldsymbol{F}) + \frac{1}{\lambda\Delta}((\boldsymbol{M}\widehat{\boldsymbol{C}}_0)'\boldsymbol{M} - \lambda\Delta\boldsymbol{M}'\boldsymbol{1}) \tag{32}$$

$$\boldsymbol{H}_w = \frac{1}{\sigma^2}\boldsymbol{I} + \frac{1}{\lambda\Delta}\boldsymbol{M}'\boldsymbol{M} \tag{33}$$

Note that this solution is the optimal linear solution, under the assumption that spikes follow a Gaussian distribution, and if often referred to as the Wiener filter, regression with a smoothing prior, or ridge regression. To estimate the parameters for the Wiener filter, we take the same approach as above:

$$\widehat{\boldsymbol{\theta}}_w \approx \underset{\boldsymbol{\theta}_w}{\operatorname{argmax}} P[\boldsymbol{F}|\widehat{\boldsymbol{n}}_w, \boldsymbol{\theta}_w) P[\widehat{\boldsymbol{n}}_w|\boldsymbol{\theta}_w) \tag{34a}$$

$$= \underset{\boldsymbol{\theta}_w}{\operatorname{argmax}} -\frac{T}{2}\log(4\pi^2\sigma^2\lambda\Delta) - \frac{1}{2\sigma^2}\left\| \boldsymbol{Y}_w + \boldsymbol{\eta}_w \boldsymbol{X}_w \right\|^2 - \frac{1}{2\lambda\Delta}\left\| \widehat{\boldsymbol{n}}_w - \lambda\Delta\boldsymbol{1} \right\|^2 \tag{34b}$$

where $\boldsymbol{Y}_w$, $\boldsymbol{\eta}_w$, and $\boldsymbol{X}_w$ are defined as their subscriptless counterparts in Eq. (14).