Fast non-negative deconvolution for spike train inference from population calcium imaging

Joshua T. Vogelstein, Adam M. Packer, Tim A. Machado,
Tanya Sippy, Baktash Babadi, Rafael Yuste, Liam Paninski

December 9, 2009

Abstract

Calcium imaging for observing spiking activity from large populations of neurons are quickly gaining popularity. While the raw data are fluorescence movies, the underlying spike trains are of interest. This work presents a fast non-negative deconvolution filter to infer the approximately most likely spike train for each neuron, given the fluorescence observations. This algorithm outperforms optimal linear deconvolution (Wiener filtering) on both simulated and biological data. The performance gains come from restricting the inferred spike trains to be positive (using an interior-point method), unlike the Wiener filter. The algorithm is fast enough that even when imaging over 100 neurons, inference can be performed on the set of all observed traces faster than real-time. Performing optimal spatial filtering on the images further refines the estimates. Importantly, all the parameters required to perform the inference can be estimated using only the fluorescence data, obviating the need to perform joint electrophysiological and imaging calibration experiments.

1 Introduction

Simultaneously imaging large populations of neurons using calcium sensors is becoming increasingly popular [38], both *in vitro* [33, 12] and *in vivo* [23, 7, 16], and will likely continue as the signal-to-noise-ratio (SNR) of genetic sensors continues to improve [6, 19, 35]. Whereas the data from these experiments are movies of time-varying fluorescence traces, the desired signal consists of spike trains of the observable neurons. Unfortunately, finding the most likely spike train is a challenging computational task, due to limitations of the SNR and temporal resolution, unknown parameters, and computational intractability.

A number of groups have therefore proposed algorithms to infer spike trains from calcium fluorescence data using very different approaches. Early approaches simply thresholded dF/F (e.g., [31, 20]) to obtain "event onset times." More recently, Greenberg et al. [8] developed a template matching algorithm to identify individual spikes. Holekamp et al. [9] then applied an optimal linear deconvolution (ie, the Wiener filter) to the fluorescence data. This approach is natural from a signal processing standpoint, but does not utilize the knowledge that spikes are always positive. Sasaki et al. [30] proposed using machine learning techniques to build a nonlinear supervised classifier, requiring many hundreds of examples of joint electrophysiological and imaging data to "train" the algorithm to learn what effect spikes have on fluorescence. Vogelstein et al. [34] proposed a biophysical model-based sequential Monte Carlo method to efficiently estimate the probability of a spike in each image frame, given the entire fluorescence time-series. While effective, that approach is not suitable for online analyses of populations of neurons, as the computations run in about real-time per neuron (ie, analyzing one minute of data requires about one minute of computational time on a standard laptop computer).

The present work starts by building a simple model relating spiking activity to fluorescence traces. Unfortunately, inferring the most likely spike train given this model is computationally intractable. Making some reasonable approximations leads to an algorithm that infers the approximately most likely spike train, given the fluorescence data. This algorithm has a few particularly noteworthy features, relative to other approaches. First, spikes are assumed to be positive. This assumption often improves filtering results when the underlying signal has this property [27, 21, 14, 15, 24, 11, 4, 25]. Second, the algorithm is fast: it can process a calcium trace from 50,000 images in about one second on a standard laptop computer. In fact, filtering the signals for an entire population of about 100 neurons runs faster than real-time. This speed facilitates using this filter online, as observations are being collected. In addition

to these two features, the model may be generalized in a number of ways, including incorporating spatial filtering of the raw movie. The efficacy of the proposed filter is demonstrated on several biological data-sets, suggesting that this algorithm is a powerful and robust tool for online spike train inference. The code (which is a simple Matlab script) is available from the authors upon request.

2   Methods

2.1   Data driven generative model

Figure 1 shows data from a typical *in vitro* epifluorescence experiment (see section 2.7 for data collection details). The top panel shows the mean frame of this movie, including 3 neurons, two of which are patched. To build the model, the pixels within a region-of-interest (ROI) are selected (white circle). Given the ROI, all the pixel intensities of each frame can be averaged, to get a one-dimensional fluorescence time-series, as shown in the bottom left panel (black line). By patching onto this neuron, the spike train can also be directly observed (black bars). Previous work suggests that this fluorescence signal might be well characterized by convolving the spike train with an exponential, and adding noise [38]. This model is confirmed by convolving the true spike train with an exponential (gray line, bottom left panel), and then looking at the distribution of the residuals. The bottom right panel shows a histogram of the residuals (solid line), and the best fit Gaussian distribution (dashed line).

Figure 1: Typical *in vitro* data suggest that a reasonable first order model may be constructed by convolving the spike train with an exponential and adding Gaussian noise. Top panel: the average (over frames) of a typical field-of-view. Bottom left: true spike train recorded via a patch electrode (black bars), convolved with an exponential (gray line), superimposed on the fluorescence trace (black line). While the spike train and fluorescence trace are measured data, the calcium is not directly measured, but rather, inferred. Bottom right: a histogram of the residual error between the gray and black lines from the bottom left panel (dashed line), and the best fit Gaussian (solid line). Note that the Gaussian model provides a good fit for the residuals here.

The above observations may be formalized as follows. Assume there is a one-dimensional fluorescence trace, $\boldsymbol{F}$ (throughout this text $\boldsymbol{X}$ indicates the vector $[X_1, \ldots, X_T]$, where T is the index of the final frame), from a neuron. At time t, the fluorescence measurement F_t is a linear-Gaussian function of the intracellular calcium concentration at that time, $[\mathrm{Ca}^{2+}]_t$:

$$F_t = \alpha([\mathrm{Ca}^{2+}]_t + \beta) + \sigma\varepsilon_t, \qquad \varepsilon_t \overset{iid}{\sim} \mathcal{N}(0, 1). \tag{1}$$

The scale, α, absorbs all experimental variables impacting the scale of the signal, including the number of sensors within the cell, photons per calcium ion, amplification of the imaging system, etc. Similarly, the offset, β, absorbs the 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. The noise at each time, ε_t, is independently and identically distributed according to a standard normal distribution (ie, Gaussian with zero mean and unit variance), as indicated by the notation $\overset{iid}{\sim} \mathcal{N}(0, 1)$.

Then, assuming that the intracellular calcium concentration, $[\mathrm{Ca}^{2+}]_t$, jumps by A μM after each spike, and subsequently decays back down to C_b μM with time constant τ, one can write:

$$[\mathrm{Ca}^{2+}]_{t+1} = (1 - \Delta/\tau)[\mathrm{Ca}^{2+}]_t + (\Delta/\tau)C_b + An_t \tag{2}$$

where Δ is the time step size — which is the frame duration, or $1/(\text{frame rate})$ — and n_t indicates the number of times the neuron spiked in frame t. Note that because $[\mathrm{Ca}^{2+}]_t$ and F_t are linearly related to one another, the fluorescence scale, α, and calcium scale, A, are not identifiable. In other words, either can be set to unity without loss of generality, as the other can absorb the scale entirely. Similarly, the fluorescence offset, β, and calcium baseline, C_b are not identifiable, so either can be set to zero without loss of generality. Finally, letting $\gamma = (1 - \Delta/\tau)$, Eq. (2) can be rewritten replacing $[\mathrm{Ca}^{2+}]_t$ with its non-dimensionalized counterpart, C_t:

$$C_{t+1} = \gamma C_t + n_t. \tag{3}$$

Note that C_t does not refer to absolute intracellular concentration of calcium, but rather, a relative measure (see [34] for a more general model). The gray line in the bottom left panel of Figure 1 corresponds to the putative C of the observed neuron.

To complete the "generative model" (ie, a model from which simulations can be generated), the distribution from which spikes are sampled must be defined. Perhaps the simplest first order description of spike trains is that at each time, spikes are sampled according to a Poisson distribution with some rate:

$$n_t \overset{iid}{\sim} \text{Poisson}(\lambda\Delta) \tag{4}$$

where $\lambda\Delta$ is the expected firing rate per bin, and Δ is included to ensure that the expected firing rate is independent of the frame rate. Thus, Eqs. (1), (3), and (4) complete the generative model.

2.2   Goal

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

$$\widehat{n} = \underset{n_t \in \mathbb{N}_0 \forall t}{\text{argmax}} P[n|F], \tag{5}$$

where $P[n|F]$ is the posterior probability of a spike train, n, given the fluorescent trace, F, and n_t is constrained to be an integer, $\mathbb{N}_0 = \{0, 1, 2, \ldots\}$, because of the above assumed Poisson distribution. From Bayes' rule, the posterior can be rewritten:

$$P[n|F] = \frac{P[n, F]}{P[F]} = \frac{1}{P[F]} P[F|n]P[n], \tag{6}$$

where $P[F]$ is the evidence of the data, $P[F|n]$ is the likelihood of observing a particular fluorescence trace F, given the spike train n, and $P[n]$ is the prior probability of a spike train. Plugging the far right-hand-side of Eq. (6) into Eq. (5), yields:

$$\widehat{n} = \underset{n_t \in \mathbb{N}_0 \forall t}{\text{argmax}} \frac{1}{P[F]} P[F|n]P[n] = \underset{n_t \in \mathbb{N}_0 \forall t}{\text{argmax}} P[F|n]P[n], \tag{7}$$

where the second equality follows because $P[F]$ merely scales the results, but does not change the relative quality of various spike trains. Both $P[F|n]$ and $P[n]$ are available from the above model:

$$P[F|n] = P[F|C] = \prod_{t=1}^{T} P[F_t|C_t], \tag{8a}$$

$$P[n] = \prod_{t=1}^{T} P[n_t], \tag{8b}$$

where the first equality in Eq. (8a) follows because C is deterministic given n, and the second equality follows from Eq. (1). Further, Eq. (8b) follows from the Poisson process assumption, Eq. (4). Both $P[F_t|C_t]$ and $P[n_t]$ can be written explicitly:

$$P[F_t|C_t] = \mathcal{N}(\alpha(C_t + \beta), \sigma^2), \tag{9a}$$
$$P[n_t] = \text{Poisson}(\lambda\Delta), \tag{9b}$$

where both equations follow from the above model. Now, plugging Eq. (9) back into (8), and plugging that result into Eq. (7), yields:

$$\widehat{n} = \underset{n_t \in \mathbb{N}_0 \forall t}{\text{argmax}} \prod_{t=1}^{T} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{-\frac{1}{2} \frac{(F_t - \alpha(C_t + \beta))^2}{\sigma^2}\right\} \frac{\exp\{-\lambda\Delta\}(\lambda\Delta)^{n_t}}{n_t!} \tag{10a}$$

$$= \underset{n_t \in \mathbb{N}_0 \forall t}{\text{argmax}} \sum_{t=1}^{T} \left\{-\frac{1}{2\sigma^2}(F_t - \alpha(C_t + \beta))^2 + n_t \log \lambda\Delta - \log n_t!\right\}, \tag{10b}$$

where the second equality follows from taking the logarithm of the right-hand-side and dropping terms that do not depend on n. Unfortunately, solving Eq. (10b) exactly is computationally intractable, as it requires a nonlinear search over an infinite number of possible spike trains. The search space could be restricted by imposing an upper bound, k, on the number of spikes within a frame. However, in that case, the computational complexity scales *exponentially* with the number of image frames — i.e., the number of computations required would scale with k^T — which for pragmatic reasons is intractable.

2.3   Inferring the most likely spike train, given a fluorescence trace

The goal here is to develop an algorithm to efficiently approximate $\widehat{n}$, the most likely spike train, given the fluorescence trace. Because of the computational intractability described above, Eq. (10) is approximated by modifying Eq. (4), replacing the Poisson distribution with an exponential distribution of the same mean. Modifying Eq. (10) to incorporate this approximation yields:

$$\widehat{n} \approx \underset{n_t > 0\, \forall t}{\operatorname{argmax}} \prod_{t=1}^{T} \left\{ \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{ -\frac{1}{2} \frac{(F_t - \alpha(C_t + \beta))^2}{\sigma^2} \right\} (\lambda\Delta) \exp\{-\lambda\Delta n_t\} \right\} \tag{11a}$$

$$= \underset{n_t > 0\, \forall t}{\operatorname{argmax}} \sum_{t=1}^{T} -\frac{1}{2\sigma^2}(F_t - \alpha(C_t + \beta))^2 - n_t\lambda\Delta \tag{11b}$$

where the constraint on n_t has been relaxed from $n_t \in \mathbb{N}_0$ to $n_t \geq 0$ (since the exponential distribution can yield any non-negative number). The advantage of this approximation is that the optimization problem becomes concave in C, meaning that any gradient ascent method guarantees achieving the global maximum (because there are no local maxima, other than the single global maximum). To see that Eq. (11b) is concave in C, rearrange Eq. (3) to obtain, $n_t = C_t - \gamma C_{t-1}$, so Eq. (11b) can be rewritten:

$$C = \underset{C_t - \gamma C_{t-1} > 0\, \forall t}{\operatorname{argmax}} \sum_{t=1}^{T} -\frac{1}{2\sigma^2}(F_t - \alpha(C_t + \beta))^2 - (C_t - \gamma C_{t-1})\lambda\Delta \tag{12}$$

which is a sum of terms that are concave in C_t, so the whole right-hand-side is concave. Unfortunately, the integer constraint has been lost, i.e., the answer could include "partial" spikes. This disadvantage can be remedied by thresholding (ie, setting $n_t = 1$ for all n_t greater than some threshold, and the rest setting to zero), or by considering the magnitude of a partial spike at time t as an indication of the probability of a spike occurring during frame t. Note that replacing a Poisson with an exponential is a common approximation technique in the machine learning literature [3, 25], as the exponential distribution is the closest log-concave relaxation to its non-log-concave counterpart, the Poisson distribution. More specifically, the probability mass function of a Poisson distributed random variable with low rate is very similar to the probability density function of a random variable with an exponential distribution. While this convex relaxation makes the problem tractable, the "sharp" threshold imposed by the non-negativity constraint prohibits the use of standard gradient ascent techniques. This may be rectified by dropping the sharp threshold, and adding a barrier term which must approach $-\infty$ as n_t approaches zero (this approach is often called an "interior-point" method). Iteratively reducing the weight of the barrier term guarantees convergence to the correct solution. Thus, the goal is to efficiently solve:

$$\widehat{C}_z = \underset{C}{\operatorname{argmax}} \sum_{t=1}^{T} \left(-\frac{1}{2\sigma^2}(F_t - \alpha(C_t + \beta))^2 - (C_t - \gamma C_{t-1})\lambda\Delta + z\log(C_t - \gamma C_{t-1}) \right). \tag{13}$$

where $\log(\cdot)$ is the "barrier term", and z is the weight of the barrier term. Iteratively solving for $\widehat{C}_z$ for z going from one down to nearly zero, guarantees convergence to $\widehat{C}$ [3]. The concavity of Eq. (13) facilitates utilizing any number of techniques guaranteed to find the global maximum. Because the argument of Eq. (13) is twice analytically differentiable, one can use the Newton-Raphson technique [28]. The special tridiagonal structure of the Hessian enables each Newton-Raphson step to be very efficient (as described below). To proceed, Eq. (13) can be rewritten in

matrix notation. Note that:

$$\boldsymbol{MC} = \begin{bmatrix} -\gamma & 1 & 0 & 0 & \cdots & 0 \\ 0 & -\gamma & 1 & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & -\gamma & 1 & 0 \\ 0 & \cdots & 0 & 0 & -\gamma & 1 \end{bmatrix} \begin{bmatrix} C_1 \\ C_2 \\ \vdots \\ C_{T-1} \\ C_T \end{bmatrix} = \begin{bmatrix} n_1 \\ n_2 \\ \vdots \\ n_{T-1} \end{bmatrix}, \tag{14}$$

where $\boldsymbol{M} \in \mathbb{R}^{(T-1) \times T}$ is a bidiagonal matrix. Then, letting $\boldsymbol{1}$ be a $T-1$ dimensional column vector and $\boldsymbol{\lambda} = \lambda \Delta \boldsymbol{1}$ yields the objection function, Eq. (13), in more compact matrix notation:

$$\widehat{\boldsymbol{C}}_z = \operatorname*{argmax}_{\boldsymbol{MC} \geq \boldsymbol{0}} -\frac{1}{2\sigma^2} \|\boldsymbol{F} - \alpha(\boldsymbol{C} + \beta)\|_2^2 - (\boldsymbol{MC})^\mathsf{T}\boldsymbol{\lambda} + z \log(\boldsymbol{MC})^\mathsf{T}\boldsymbol{1}, \tag{15}$$

where $\boldsymbol{MC} \geq \boldsymbol{0}$ indicates that every element of $\boldsymbol{MC}$ is greater than or equal to zero, $^\mathsf{T}$ indicates the transpose operation, $\log(\cdot)$ indicates an element-wise logarithm, and $\|x\|_2$ is the standard L_2 norm, i.e., $\|x\|_2^2 = \sum_i x_i^2$. When using Newton-Raphson to ascend a surface, one iteratively computes both the gradient (first derivative) and Hessian (second derivative) of the argument to be maximized, with respect to the variables of interest ($\boldsymbol{C}$ here). Then, the estimate is updated using $\boldsymbol{C}_z \leftarrow \boldsymbol{C}_z + s\boldsymbol{d}$, where s is the step size and $\boldsymbol{d}$ is the step direction obtained by solving $\boldsymbol{Hd} = \boldsymbol{g}$. The gradient, $\boldsymbol{g}$, and Hessian, $\boldsymbol{H}$, for this model, with respect to $\boldsymbol{C}$, are given by:

$$\boldsymbol{g} = -\frac{\alpha}{\sigma^2}(\boldsymbol{F} - \alpha(\boldsymbol{C}^\mathsf{T} + \beta)) + \boldsymbol{M}^\mathsf{T}\boldsymbol{\lambda} - z\boldsymbol{M}^\mathsf{T}(\boldsymbol{MC})^{-1} \tag{16a}$$

$$\boldsymbol{H} = \frac{\alpha^2}{\sigma^2}\boldsymbol{I} + z\boldsymbol{M}^\mathsf{T}(\boldsymbol{MC})^{-2}\boldsymbol{M} \tag{16b}$$

where the exponents on the vector $\boldsymbol{MC}$ indicate element-wise operations. The step size, s, is found using "backtracking linesearches", which finds the maximal s that increases the posterior and is between zero and one [28].

Typically, implementing Newton-Raphson requires inverting the Hessian, i.e., solving $\boldsymbol{d} = \boldsymbol{H}^{-1}\boldsymbol{g}$, a computation that scales *cubically* with T (requires on the order of T^3 operations). Already, this would be a drastic improvement over the most efficient algorithm assuming Poisson spikes, which would require k^T operations (where k is the maximum number of spikes per frame). Here, because $\boldsymbol{M}$ is bidiagonal, the Hessian is tridiagonal, so the solution may be found in about T operations, via standard banded Gaussian elimination techniques (which can be implemented efficiently in Matlab using $\boldsymbol{H}\backslash\boldsymbol{g}$, assuming $\boldsymbol{H}$ is represented as a sparse matrix) [25]. In other words, the above approximation and inference algorithm reduces computations from *exponential* time to *linear* time. Appendix A contains pseudocode for this algorithm, including learning the parameters, as described below.

2.4   Learning the parameters

We assumed above that the parameters governing the model, $\boldsymbol{\theta} = \{\alpha, \beta, \sigma, \gamma, \lambda\}$, were known, but in practice they are typically unknown. An algorithm to estimate the most likely parameters, $\widehat{\boldsymbol{\theta}}$, could proceed as follows: (i) initialize some estimate of the parameters, $\widehat{\boldsymbol{\theta}}$, then (ii) recursively compute $\widehat{\boldsymbol{n}}$ using those parameters, and update $\widehat{\boldsymbol{\theta}}$ given the new $\widehat{\boldsymbol{n}}$, until some convergence criteria is met. Below, details are provided for each step.

2.4.1   Initializing the parameters

Because the model introduced above is linear, the scale of $\boldsymbol{F}$ relative to $\boldsymbol{n}$ is arbitrary. Therefore, before filtering, $\boldsymbol{F}$ is linearly "squashed" between zero and one, ie $\boldsymbol{F} \leftarrow (\boldsymbol{F} - F_{min})/(F_{max} - F_{min})$, where F_{min} and F_{max} are the observed minimum and maximum of $\boldsymbol{F}$, respectively. Given this normalization, α is set to one. Because spiking is assumed to be sparse, $\boldsymbol{F}$ tends to be around baseline, so β is initialized to be the median of $\boldsymbol{F}$, and σ is initialized as the median absolute deviation of $\boldsymbol{F}$, i.e., $\sigma = \operatorname{median}_t(|F_t - \operatorname{median}_s(F_s)|)/K$, where $\operatorname{median}_i(X_i)$ indicates the median of X with respect to index i, and $K = 1.4785$ is the correction factor when using median absolute deviation as a robust estimator of the standard deviation. Because in these data, the posterior tends to be relatively flat along the γ dimension, i.e., large changes in γ result in relatively small changes in the posterior, estimating γ is difficult. Further, previous work has shown that results are somewhat robust to minor variations in time constant [37]; therefore γ is initialized at $1 - \Delta/(1\text{sec})$, which is fairly typical [26]. Finally, λ is initialized at 1 Hz, which is between typical baseline and evoked spike rate for these data.

2.4.2 Estimating the parameters given $\widehat{\mathbf{n}}$

Ideally, one could integrate out the hidden variables, to find the most likely parameters:

$$\widehat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta}}{\operatorname{argmax}} \int P[\boldsymbol{F}, \boldsymbol{C}|\boldsymbol{\theta}]d\boldsymbol{C} = \underset{\boldsymbol{\theta}}{\operatorname{argmax}} \int P[\boldsymbol{F}|\boldsymbol{C};\boldsymbol{\theta}]P[\boldsymbol{C}|\boldsymbol{\theta}]d\boldsymbol{C}. \tag{17}$$

However, evaluating those integrals is not particularly tractable. Therefore, Eq. (17) is approximated by simply maximizing the parameters given the MAP estimate of the hidden variables:

$$\widehat{\boldsymbol{\theta}} \approx \underset{\boldsymbol{\theta}}{\operatorname{argmax}} P[\boldsymbol{F}, \widehat{\boldsymbol{C}}|\boldsymbol{\theta}] = \underset{\boldsymbol{\theta}}{\operatorname{argmax}} P[\boldsymbol{F}|\widehat{\boldsymbol{C}};\boldsymbol{\theta}]P[\widehat{\boldsymbol{n}}|\boldsymbol{\theta}] = \underset{\boldsymbol{\theta}}{\operatorname{argmax}}\{\log P[\boldsymbol{F}|\widehat{\boldsymbol{C}};\boldsymbol{\theta}] + \log P[\widehat{\boldsymbol{n}}|\boldsymbol{\theta}]\}, \tag{18}$$

where $\widehat{\boldsymbol{C}}$ and $\widehat{\boldsymbol{n}}$ are determined using the above described inference algorithm. The approximation in Eq. (18) is good whenever most of the mass in the integral in Eq. (18) is around the MAP sequence, $\widehat{\boldsymbol{C}}$.[1] The argument from the right-hand-side of Eq. (18) may be expanded:

$$\log P[\boldsymbol{F}|\widehat{\boldsymbol{C}};\boldsymbol{\theta}] + \log P[\widehat{\boldsymbol{n}}|\boldsymbol{\theta}] = \sum_{t=1}^{T} \log P[F_t|\widehat{C}_t;\alpha,\beta,\sigma] + \sum_{t=1}^{T} \log P[\widehat{n}_t|\lambda]. \tag{19}$$

Note that the two terms in the right-hand-side of Eq. (19) may be optimized separately. The maximum likelihood estimate (MLE) for the observation parameters, $\{\alpha, \beta, \sigma\}$, is therefore given by:

$$\{\widehat{\alpha}, \widehat{\beta}, \widehat{\sigma}\} = \underset{\alpha,\beta,\sigma>0}{\operatorname{argmax}} \sum_{t=1}^{T} \log P[F_t|\widehat{C}_t;\beta,\sigma] = \underset{\beta,\sigma>0}{\operatorname{argmax}} -\frac{1}{2}(2\pi\sigma^2) - \frac{1}{2}\left(\frac{F_t - \alpha(\widehat{C}_t + \beta)}{\sigma}\right)^2. \tag{20}$$

Note that a rescaling of α may be offset by a complementary rescaling of $\boldsymbol{C}$ and β. Therefore, because the scale of $\boldsymbol{C}$ is arbitrary, α can be set to one without loss of generality. Plugging $\alpha = 1$ into Eq. (20), and maximizing with respect to β yields:

$$\widehat{\beta} = \underset{\beta>0}{\operatorname{argmax}} \sum_{t=1}^{T} -(F_t - (\widehat{C}_t + \beta))^2. \tag{21}$$

Computing the gradient with respect to β, setting the answer to zero, and solving for $\widehat{\beta}$, yields $\widehat{\beta} = \frac{1}{T}\sum_t(F_t - \widehat{C}_t)$. Similarly, computing the gradient of Eq. (20) with respect to σ, setting it to zero, and solving for $\widehat{\sigma}$ yields:

$$\widehat{\sigma} = \sqrt{\frac{1}{T}\sum_t (F_t - \widehat{C}_t - \widehat{\beta})^2}, \tag{22}$$

which is simply the root-mean-square of the residual error. Finally, the MLE of $\widehat{\lambda}$ is given by solving:

$$\widehat{\lambda} = \underset{\lambda>0}{\operatorname{argmax}} \sum_t (\log(\lambda\Delta) - \widehat{n}_t\lambda\Delta), \tag{23}$$

which, again, computing the gradient with respect to λ, setting it to zero, and solving for $\widehat{\lambda}$, yields $\widehat{\lambda} = T/(\Delta\sum_t \widehat{n}_t)$, which is the inverse of the inferred average firing rate.

   Iterations stop whenever (i) the iteration number exceeds some upper bound, or (ii) the relative change in likelihood does not exceed some lower bound. In practice, parameters tend to converge after several iterations, given the above initializations.

2.5 Spatial filtering

In the above, we assumed that the raw movie of fluorescence measurements collected by the experimenter had undergone two stages of preprocessing before filtering. First, the movie was segmented, to determine regions-of-interest

---

[1]Eq. (18) may be considered a first-order Laplace approximation [13].

(ROIs), yielding a vector, $\vec{F}_t = (F_{1,t}, \ldots, F_{N_p,t})$, which corresponded to the fluorescence intensity at time t for each of the N_p pixels in the ROI. Second, at each time t, that vector was projected into a scalar, yielding F_t, the assumed input to the filter. In this section, the optimal projection is determined by considering a more general model:

$$F_{x,t} = \alpha_x(C_t + \beta) + \sigma\varepsilon_{x,t}, \qquad\qquad \varepsilon_{x,t} \overset{iid}{\sim} \mathcal{N}(0,1) \qquad\qquad (24)$$

where α_x scales each pixel, from which some number of photons are contributed due to calcium fluctuations, C_t, and others due to baseline fluorescence, β. Further, the noise is assumed to be both spatially and temporally white, with standard deviation, σ, in each pixel (this assumption can easily be relaxed, by modifying the covariance matrix of $\varepsilon_{x,t}$). Performing inference in this more general model proceeds in a nearly identical manner as before. In particular, the maximization, gradient, and Hessian become:

$$\widehat{\boldsymbol{C}}_z = \underset{\boldsymbol{MC} \geq \boldsymbol{0}}{\operatorname{argmax}} -\frac{1}{2\sigma^2}\left\|\vec{\boldsymbol{F}} - \vec{\alpha}(\boldsymbol{C}^{\mathsf{T}} + \boldsymbol{\beta}^{\mathsf{T}})\right\|_F^2 - (\boldsymbol{MC})^{\mathsf{T}}\boldsymbol{\lambda} + z\log(\boldsymbol{MC})^{\mathsf{T}}\boldsymbol{1}, \qquad (25)$$

$$\boldsymbol{g} = \frac{\vec{\alpha}}{\sigma^2}(\vec{\boldsymbol{F}} - \vec{\alpha}(\boldsymbol{C}^{\mathsf{T}} + \boldsymbol{\beta})) - \boldsymbol{M}^{\mathsf{T}}\boldsymbol{\lambda} + z\boldsymbol{M}^{\mathsf{T}}(\boldsymbol{MC})^{-1} \qquad (26)$$

$$\boldsymbol{H} = -\frac{\vec{\alpha}^{\mathsf{T}}\vec{\alpha}}{\sigma^2}\boldsymbol{I} - z\boldsymbol{M}^{\mathsf{T}}(\boldsymbol{MC})^{-2}\boldsymbol{M} \qquad (27)$$

where $\vec{\boldsymbol{F}}$ is an $N_p \times T$ element matrix, $\vec{\alpha}$ is a column vector of length N_p, $\boldsymbol{\beta} = \beta\boldsymbol{1}_T$, where $\boldsymbol{1}_T$ is a column vector of ones with length T, $\boldsymbol{I}$ is an $N_p \times N_p$ identity matrix, and $\|x\|_F$ indicates the Frobenius norm, i.e. $\|x\|_F^2 = \sum_{i,j} x_{i,j}^2$. Note that to speed up computation, one can first project the $N_c \times T$ dimensional movie onto the spatial filter, $\vec{\alpha}$, yielding a one-dimensional time series, F, reducing the problem to evaluating a $T \times 1$ vector norm, as in Eq. (15).

Typically, the parameters $\vec{\alpha}$ and β are unknown, and therefore must be estimated from the data. Following the strategy developed in the last section, we first initialize the parameters. Let the initial spatial filter be the median image frame, i.e., $\widehat{\alpha}_x = \operatorname{median}_t(F_{x,t})$, and the initial offset be the total movie median, $\widehat{\beta} = \operatorname{median}_{x,t}(F_{x,t})$. Given these robust initializations, the maximum likelihood estimator for each α_x is given by:

$$\widehat{\alpha}_x = \underset{\alpha_x}{\operatorname{argmax}} P[\boldsymbol{F}_x | \widehat{\boldsymbol{C}}] = \underset{\alpha_x}{\operatorname{argmax}} \sum_t \log P[F_{x,t} | \widehat{C}_t] \qquad (28\mathrm{a})$$

$$= \underset{\alpha_x}{\operatorname{argmax}} \sum_t \left\{ -\frac{1}{2}(2\pi\sigma^2) - \frac{1}{2\sigma^2}\left(F_{x,t} - \alpha_x(\widehat{C}_t + \widehat{\beta})\right)^2 \right\} = \underset{\alpha_x}{\operatorname{argmax}} \sum_t -(F_{x,t} - \alpha_x(\widehat{C}_t - \widehat{\beta}))^2, \quad (28\mathrm{b})$$

which is solved by regressing $(\boldsymbol{C} + \widehat{\boldsymbol{\beta}})$ onto $\boldsymbol{F}_x$. In other words, by computing the gradient of Eq. (28b) with respect to α_x and setting to zero, one obtains (using Matlab notation): $\widehat{\alpha}_x = (\boldsymbol{C} + \widehat{\boldsymbol{\beta}})\backslash\boldsymbol{F}_x$. Computing the gradient of Eq. (28) with respect to $\widehat{\beta}$, setting the result to zero, and solving for $\widehat{\beta}$, yields:

$$\widehat{\beta} = \frac{1}{TN_p}\sum_{t=1}^{T}\sum_{x=1}^{N_p} \frac{F_{x,t} + \widehat{\alpha}_x\widehat{C}_t}{\widehat{\alpha}_x^2}. \qquad (29)$$

Iterating these two steps results in a coordinate ascent approach to estimate $\vec{\alpha}$ and β [3]. As in the scalar F_t case, we iterate estimating the parameters of this model, $\boldsymbol{\theta} = \{\vec{\alpha}, \beta, \sigma, \gamma, \lambda\}$, and the spike train, $\boldsymbol{n}$. Because of the free scale term discussed in section 2.4, the absolute magnitude of $\vec{\alpha}$ is not identifiable. Thus, convergence is defined here by the "shape" of the spike train converging, i.e., the norm of the difference between the inferred spike trains from subsequent iterations, both normalized such that $\max(\widehat{n}_t) = 1$. In practice, this procedure converged after several iterations.

2.6 Overlapping spatial filters

It is not always possible to segment the movie into pixels containing only fluorescence from a single neuron. Therefore, the above model can be generalized to incorporate multiple neurons within an ROI. Specifically, letting the superscript i index the N_c neurons in this ROI yields:

$$\vec{F}_t = \sum_{i=1}^{N_c} \vec{\alpha}^i(C_t^i + \beta^i) + \sigma\vec{\varepsilon}_t, \qquad\qquad \vec{\varepsilon}_t \overset{iid}{\sim} \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}) \qquad (30)$$

$$C_t^i = \gamma^i C_{t-1}^i + n_t^i, \qquad\qquad n_t^i \overset{iid}{\sim} \operatorname{Poisson}(n_t^i; \lambda_i\Delta) \qquad (31)$$

where each neuron is implicitly assumed to be independent, and each pixel is conditionally independent and identically distributed with standard deviation σ, given the underlying calcium signals. To perform inference in this more general model, let $\mathbf{1}$ and $\mathbf{0}$ correspond to an $N_c \times 1$ row vector of ones, and zeros, respectively, and $\boldsymbol{n}_t = [n_t^1, \ldots, n_t^{N_c}]$, $\boldsymbol{C}_t = [C_t^1, \ldots, C_t^{N_c}]$, and $\boldsymbol{\Gamma} = [-\gamma^1, \ldots, -\gamma^{N_c}]^\mathsf{T}$, yielding:

$$\boldsymbol{MC} = \begin{bmatrix} -\boldsymbol{\Gamma} & \mathbf{1} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & -\boldsymbol{\Gamma} & \mathbf{1} & \mathbf{0} & \cdots & \mathbf{0} \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \mathbf{0} & \cdots & \mathbf{0} & -\boldsymbol{\Gamma} & \mathbf{1} & \mathbf{0} \\ \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} & -\boldsymbol{\Gamma} & \mathbf{1} \end{bmatrix} \begin{bmatrix} \boldsymbol{C}_1 \\ \boldsymbol{C}_2 \\ \vdots \\ \boldsymbol{C}_{T-1} \\ \boldsymbol{C}_T \end{bmatrix} = \begin{bmatrix} \boldsymbol{n}_1 \\ \boldsymbol{n}_2 \\ \vdots \\ \boldsymbol{n}_{T-1} \end{bmatrix}, \tag{32}$$

and proceed as above. Note that Eq. (32) is very similar to Eq. (14), except that $\boldsymbol{M}$ is no longer tridiagonal, but rather, block tridiagonal (and $\boldsymbol{C}_t$ and $\boldsymbol{n}_t$ are vectors instead of scalars). Importantly, the Thomas algorithm, which is a simplified form of Gaussian elimination, finds the solution to linear equations with block tridiagonal matrices in linear time, so the efficiency gained from utilizing the tridiagonal structure above is maintained for this block tridiagonal structure [28].

If the parameters are unknown, they must be estimated. Define $\boldsymbol{\alpha}_x = [\alpha_x^1, \ldots, \alpha_x^{N_c}]^\mathsf{T}$ and $\boldsymbol{\beta} = [\beta^1, \ldots, \beta^{N_c}]^\mathsf{T}$. To initialize, let $\boldsymbol{\beta} = \text{median}_{x,t}(F_{x,t})\mathbf{1}_{N_p}$, where $\mathbf{1}_{N_p}$ corresponds to a column vector of length N_p. To initialize $\vec{\alpha} = [\vec{\alpha}_1, \ldots, \vec{\alpha}_{N_c}]$, the goal is to be able to represent the matrix, $\vec{\boldsymbol{F}}$, as the sum of only N_c time-varying components, also known as a *low-rank* approximation. Singular value decomposition is a tool known to find the low-rank approximation to a matrix, with the smallest mean-square-error of all possible low-rank approximations [10]. Because the "singular values" are equivalent to the "principal components" of the covariance of the movie, a natural initial estimate for the N_c $\vec{\alpha}$ vectors are the N_c first principal components. While other methods to initialize the spatial filters (such as "independent component analysis" [22]) could also work, because fast algorithms for computing the first few principal components are readily available [29], PCA was both sufficiently effective and efficient. Given these initializations, estimating $\vec{\alpha}$ and $\boldsymbol{\beta}$ follows very similarly as in Eqs. (28) and (29):

$$\widehat{\boldsymbol{\alpha}}_x = \operatorname*{argmin}_{\boldsymbol{\alpha}_x} \sum_{t=1}^{T} \left(F_{x,t} \sum_{i=1}^{N_c} \alpha_x^i (\widehat{C}_t^i + \widehat{\beta}^i) \right)^2 \tag{33}$$

$$\widehat{\beta}^i = \frac{1}{TN_p} \sum_{t=1}^{T} \sum_{x=1}^{N_p} \frac{F_{x,t} + \widehat{\alpha}_x^i \widehat{C}_t^i}{\widehat{\alpha}_x^2}, \tag{34}$$

where Eq. (33) is solved efficiently in Matlab using $\boldsymbol{\alpha}_x = (\widehat{\boldsymbol{C}} + \widetilde{\boldsymbol{\beta}}) \backslash \boldsymbol{F}_x$, where $\widetilde{\boldsymbol{\beta}}$ is the set of estimated baseline parameters, $\widehat{\boldsymbol{\beta}}$, concatenated T times. Convergence of parameters and spike trains in this model behaved similarly to the scenario described in section 2.5, assuming the spikes were sufficiently uncorrelated and observations had a sufficiently high SNR.

2.7 Experimental Methods

2.7.1 Slice Preparation and Imaging

All animal handling and experimentation was done according to the National Institutes of Health and local Institutional Animal Care and Use Committee guidelines. Somatosensory thalamocortical or coronal slices 350-400 μm thick were prepared from C57BL/6 mice at age P14 as described [17]. Neurons were filled with 50 μM Oregon Green Bapta 1 hexapotassium salt (OGB-1; Invitrogen, Carlsbad, CA) through the recording pipette or bulk loaded with Fura-2 AM (Invitrogen, Carlsbad, CA). Pipette solution contained 130 mM K-methylsulfate, 2 mM MgCl$_2$, 0.6 mM EGTA, 10 mM HEPES, 4 mM ATP-Mg, and 0.3 mM GTP-Tris, pH 7.2 (295 mOsm). After cells were fully loaded with dye, imaging was done by using a modified BX50-WI upright microscope (Olympus, Melville, NY). Image acquisition was performed with the C9100-12 CCD camera from Hamamatsu Photonics (Shizuoka, Japan) with arclamp illumination with excitation and emission bandpass filters at 480-500 nm and 510-550 nm, respectively (Chroma, Rockingham, VT) for Oregon Green. Imaging of Fura-2 loaded slices was performed with a confocal spinning disk (Solamere Technology Group, Salt Lake City, UT) and an Orca CCD camera from Hamamatsu Photonics (Shizuoka, Japan). Images were saved and analyzed using custom software written in Matlab (Mathworks, Natick, MA).

2.7.2   Electrophysiology

All recordings were made using the Multiclamp 700B amplifier (Molecular Devices, Sunnyvale, CA), digitized with National Instruments 6259 multichannel cards and recorded using custom software written using the LabView platform (National Instruments, Austin, TX) . Square pulses of sufficient amplitude to yield the desired number of action potentials were given as current commands to the amplifier using the LabView and National Instruments system.

2.7.3   Fluorescence preprocessing

Traces were extracted using custom Matlab scripts to (i) segment the mean image into ROIs, and then (ii) average all the pixels within the ROI. The Fura-2 fluorescence traces were inverted. As some slow drift was sometimes present in the traces, each trace was Fourier transformed, and all frequencies below 0.5 Hz were set to zero (0.5 Hz was chosen by eye), and the resulting fluorescence trace was then normalized to be between zero and one.

3   Results

3.1   Main Result

The main result of this paper is that the fast filter can find the approximately most likely spike train, $\widehat{n}$, very efficiently, and that this approach yields more accurate spike train estimates than optimal linear deconvolution. Fig. 2 depicts a simulation showing this result. Clearly, the fast filter's inferred "spike train" (third panel) more closely resembles the true spike train (second panel) than the optimal linear deconvolution's inferred spike train (bottom panel; Wiener filter). Note that neither filter results in an integer sequence, but rather, each infers a real number at each time.

The Wiener filter implicitly approximates the Poisson spike rate with a Gaussian spike rate (see Appendix B for details). A Poisson spike rate indicates that in each frame, the number of possible spikes is an integer, 0, 1, 2, ... The Gaussian approximation, however, allows for any real number of spikes in each frame, including both partial spikes, e.g., 1.4, and *negative* spikes, e.g., -0.8. While a Gaussian well approximates a Poisson distribution when rates are about 10 spikes per frame, this example is very far from that regime, so the Gaussian approximation performs relatively poorly. More specifically, the Wiener filter exhibits a "ringing" effect. Whenever fluorescence drops rapidly, the most likely underlying signal is a proportional drop. Because the Wiener filter does not impose a non-negative constraint on the underlying signal (which, in this case, is a spike train), it infers such a drop. After such a drop has been inferred, since no corresponding drop occurred in the true underlying signal here, a complementary jump is often then inferred, to "re-align" the inferred signal with the observations. This oscillatory behavior results in poor inference quality. The non-negative constraint imposed by the fast filter prevents this because the underlying signal never drops below zero, so the complementary jump never occurs either.

The inferred "spikes", however, are still not binary events when using the fast filter. This is a by-product of approximating the Poisson distribution on spikes with an exponential (cf. Eq. (11a)), because the exponential is a *continuous* distribution, versus the Poisson, which is discrete. The height of each spike is therefore proportional to the inferred calcium jump size, and can be thought of as a proxy for the confidence with which the algorithm believes a spike occurred. Importantly, by utilizing the Gaussian elimination and interior-point methods, as described in the Methods section, the computational complexity of fast filter is the same as an efficient implementation of the Wiener filter.

Figure 2: The fast filter's inferred spike train is significantly more accurate than the output of the optimal linear deconvolution (Wiener filter) on typical simulated data. Note that neither filter constrains the inference to be a sequence of integers; rather, the fast filter relaxes the constraint to allow all non-negative numbers, and the Wiener filter allows for all real numbers. The restriction of the fast filter to exclude negative numbers eliminates the ringing effect seen in the Wiener filter output, resulting in a much cleaner inference. Note that the magnitude of the inferred spikes in the fast filter output is proportional to the inferred calcium jump size. Top panel: fluorescence trace. Second panel: spike train. Third panel: fast filter inference. Bottom panel: Wiener filter inference. Note that the gray bars in the bottom panel indicate *negative* spikes. Black '+'s in bottom two panels indicate true spike times. Simulation details: $T \approx 3000$ time steps, $\Delta = 5$ msec, $\alpha = 1$, $\beta = 0$, $\sigma = 0.3$, $\tau = 1$ sec, $\lambda = 1$ Hz. Conventions for other figures as above, unless otherwise indicated.

Although in Figure 2 the model parameters were provided, in the general case, the parameters are unknown, and must therefore be estimated from the observations (as described in section 2.4). Importantly, this algorithm does not require "training" data, i.e., there is no need for joint imaging and electrophyiological experiments to estimate the parameters governing the relationship between the two. Figure 3 shows another simulated example; in this example, however, the parameters are estimated from the observed fluorescence trace alone. Again, it is clear that the fast filter far outperforms the Wiener filter.

Figure 3: The fast filter significantly outperforms the Wiener filter, even when the parameters unknown. For both filters, the appropriate parameters were estimated using only the data shown above, unlike Figure 2, in which the true parameters were provided to the filters. Simulation details as in Figure 2.

Given the above two results, the fast filter was applied to real data. More specifically, by jointly recording electrophysiologically and imaging, the true spike times are known, and the accuracy of the two filters can be compared. Figure 4 shows a result typical of the 12 joint electrophysiological and imaging experiments conducted. Although it is difficult to see in this figure, the first four "events" are actually pairs of spikes, which is reflected by the width and height of the corresponding inferred spikes when using the fast filter. This suggests that although the scale of n is arbitrary, the fast filter can correctly ascertain the number of spikes within spike events.

Figure 4: The fast filter significantly outperforms the Wiener filter on typical *in vitro* data, using OGB-1. Note that all the parameters for both filters were estimated only from the fluorescence data in the top panel (ie, not considering the voltage data at all). Again, '+'s denote true spike times extracted from the patch data, not inferred spike times from F.

Figure 5 further evaluates this claim. While recording and imaging, the cell was forced to spike once, twice, or thrice, for each spiking event. Naïvely, this would suggest that an algorithm based on a purely linear model would struggle to resolve spike fidelity in this high frequency spiking regime. However, the fast filter infers the correct number of spikes in each event. On the contrary, there is no obvious way to count the number of spikes within each event when using the Wiener filter.

Figure 5: The fast filter can often resolve the correct number of spikes within each spiking event, while imaging using OGB-1, given sufficiently high SNR. It is difficult, if not impossible, to count the number of spikes given the Wiener filter output. Recording and fitting parameters as in Figure 4. Note that the parameters were estimated using a 60 sec long recording, of which only a fraction is shown here, to more clearly depict the number of spikes per event.

3.2   Online analysis of spike trains using the fast filter

A central aim for this work was the development of an algorithm that infers spikes fast enough to use online while imaging a large population of neurons (eg, ≈ 100). Figure 6 shows a segment of the results of running the fast filter on 136 neurons, recorded simultaneously, as described in section 2.7. Note that the filtered fluorescence signals show fluctuations in spiking much more clearly than the unfiltered fluorescence trace. These spike trains were inferred in less than imaging time, meaning that one could infer spike trains for the past experiment while conducting the subsequent experiment. More specifically, a movie with 5,000 frames of 100 neurons can be analyzed in about ten seconds on a standard desktop computer. Thus, if that movie was recorded at 50 Hz, while collecting the data required 100 seconds, inferring spikes only required ten seconds, a ten-fold improvement over real-time.

3.3   Extensions

Section 2.1 describes a simple principled first-order model relating the spike trains to the fluorescence trace. A number of the simplifying assumptions can be straightforwardly relaxed, as described below.

Figure 6: The fast filter infers spike trains from a large population of neurons imaged simultaneously *in vitro*, using Fura-2, faster than real-time. Specifically, inferring the spike trains from this 400 sec long movie including 136 neurons requires only about 40 sec on a standard laptop computer. The inferred spike trains much more clearly convey neural activity than the raw fluorescence traces. Although no intracellular "ground truth" is available on this population data, the noise seems to be reduced, consistent with the other examples with ground truth. Left panel: Mean image field, segmented into ROIs each containing a single neuron. Middle panel: example fluorescence traces. Right panel: fast filter output corresponding to each associated trace. Note that neuron identity is indicated by color across the three panels.

3.3.1 Replacing Gaussian observations with Poisson

In the above, observations were assumed to have a Gaussian distribution. The statistics of photon emission and counting, however, suggest that a Poisson distribution would be more natural, especially for two-photon data [32], yielding:

$$F_t \overset{iid}{\sim} \text{Poisson}(\alpha C_t + \beta). \tag{35}$$

One additional advantage to this model over the Gaussian model, is that the variance parameter, σ^2, no longer exists, which might make learning the parameters simpler. Importantly, the log-posterior is still concave in C, as the prior remains unchanged, and the new log-likelihood term is a sum of terms concave in C:

$$\log P[\boldsymbol{F}|\boldsymbol{C}] = \sum_{t=1}^{T} \log P[F_t|C_t] = \sum_{t=1}^{T} \{F_t \log(\alpha C_t + \beta) - (\alpha C_t + \beta) - \log(F_t!)\}. \tag{36}$$

The gradient and Hessian of the log-posterior can therefore be computed analytically by substituting the above likelihood terms for those implied by Eq. (1). In practice, however, modifying the filter for this model extension did not seem to significantly improve inference results in any simulations or data (not shown).

3.3.2 Allowing for a time-varying prior

In Eq. (4), the rate of spiking is a constant. Often, additional knowledge about the experiment, including external stimuli, or other neurons spiking, can provide strong time-varying prior information [34]. A simple model modification can incorporate that feature:

$$n_t \overset{iid}{\sim} \text{Poisson}(\lambda_t \Delta), \tag{37}$$

where λ_t is now a function of time. Approximating this time-varying Poisson with a time-varying exponential with the same time-varying mean (similar to Eq. (11a)), and letting $\boldsymbol{\lambda} = [\lambda_1, \ldots, \lambda_T]^\mathsf{T} \Delta$, yields an objective function identical to Eq. (15), so log-concavity is maintained, and the same techniques may be applied. However, as above, this model extension did not yield any significantly improved filtering results (not shown).

3.3.3 Saturating fluorescence

Although all the above models assumed a *linear* relationship between F_t and C_t, the relationship between fluorescence and calcium is typically better approximated by the nonlinear Hill equation [26]. Modifying Eq. (1) to reflect this change yields:

$$F_t = \alpha \frac{C_t}{C_t + k_d} + \beta + \sigma \varepsilon_t, \qquad \varepsilon_t \overset{iid}{\sim} \mathcal{N}(0, 1). \tag{38}$$

Importantly, log-concavity of the posterior is no longer guaranteed in this nonlinear model, meaning that converging to the global maximum is no longer guaranteed. Assuming a good initialization can be found, however, if this model is more accurate, then ascending the gradient for this model might yield improved inference results. In practice, initializing with the inference from the fast filter assuming a linear model (eg, Eq. (30)) often resulted in nearly equally accurate inference, but inference assuming the above nonlinearity was far less robust than the inference assuming the linear model (not shown).

3.3.4 Using the fast filter to initialize the sequential Monte Carlo filter

A sequential Monte Carlo (SMC) method to infer spike trains can incorporate this saturating nonlinearity, as well as the other model extensions discussed above [34] . However, this SMC filter is not nearly as computationally efficient as the fast filter proposed here. Like the fast filter, the SMC filter estimates the model parameters in a completely unsupervised fashion, i.e., from the fluorescence observations, using an expectation-maximization algorithm (which requires iterating between computing the expected value of the hidden variables — C and n — and updating the paramters). In [34], parameters for the SMC filter were initialized based on other data. While effective, this initialization was often far from the final estimates, and therefore, required a relatively large number of iterations (eg, 20–25) before converging. Thus, it seemed that the fast filter could be used to obtain an improvement to the initial parameter estimates, given an appropriate rescaling to account for the nonlinearity, thereby reducing the required number of iterations to convergence. Indeed, Figure 7 shows how the SMC filter outperforms the fast filter on biological data, and only required 3–5 iterations to converge on this data, given the initialization from the fast filter (which was typical). Note that the first few events of the spike train are individual spikes, resulting in relatively small fluorescence fluctuations, whereas the next events are actually spike doublets or triplets, causing a much larger fluorescence fluctuation. Only the SMC filter picks up the individual spikes in this trace, a result typical when the effective signal-to-noise ratio (SNR) is poor. Thus, these two inference algorithms are complementary: the fast filter can be used for rapid, online inference, and for initializing the SMC filter, which can then be used to further refine the spike train estimate. Importantly, although the SMC filter often outperforms the fast filter, the fast filter is more robust, meaning that it more often works "out-of-the-box."

Figure 7: The fast filter effectively initializes the parameters for the SMC filter, significantly reducing the number of expectation-maximization iterations to convergence, on typical *in vitro* data, using OGB-1. Note that while the fast filter clearly infers the spiking events in the end of the trace, those in the beginning of the trace are less clear. On the other hand, the SMC filter more clearly separates non-spiking activity from true spikes. Also note that the ordinate on the bottom panel corresponds to the inferred probability of a spike having occurred in each frame.

3.4 Spatial filter

In the above, the filters operated on one-dimensional fluorescence traces. Typically, the data are time-series of images which are first segmented into regions-of-interest (ROI), and then (usually) averaged to obtain F_t. In theory, one could improve the effective SNR of the fluorescence trace by scaling each pixel relative to one another. In particular, pixels not containing any information about calcium fluctuations can be ignored, and pixels that are partially anti-correlated with one another could have weights with opposing signs.

Figure 8 demonstrates the potential utility of this approach. The top row shows different depictions of an ROI containing a single neuron. On the far left panel is the true spatial filter for this neuron. This particular spatial filter was chosen based on experience analyzing both *in vitro* and *in vivo* movies; often, it seems that the pixels immediately around the soma are anti-correlated with those in the soma. This effect is possibly due to the influx of calcium from the extracellular space immediately around the soma. The simulated movie (not shown) is relatively noisy, as indicated by the second panel, which depicts an exemplary image frame. The standard approach, given such a noisy movie, would be to first segment the movie to find an ROI corresponding to the soma of this cell, and then spatially average all the pixels found to be within this ROI. The third panel shows this standard "boxcar spatial filter". The fourth panel shows the mean frame. Clearly, this mean frame is very similar to the true spatial filter.

The bottom panels of Figure 8 depict the effect of using the true spatial filter, versus the typical one. The left side shows the fluorescence trace and its associated spike inference obtained from using the typical spatial filter. The right side shows the same when using the true spatial filter. Clearly, the true spatial filter results in a much cleaner fluorescence trace and spike inference. When the true spatial filter is a single Gaussian, the boxcar spatial filter works about as well as the true spatial filter (not shown).

3.5 Overlapping spatial filters

The above shows that if a ROI contains only a single neuron, the effective SNR can be enhanced by spatially filtering. However, this analysis assumes that only a single neuron is in the ROI. Often, ROIs are overlapping, or nearly

Figure 8: A simulation demonstrating that using a better spatial filter can significantly enhance the effective SNR. The true spatial filter was a difference of Gaussians: a positively weighted Gaussian of small width, and a negatively weighted Gaussian with larger width (both with the same center). Top row far left: true spatial filter. Top row second from left: example movie frame. Top row second from right: typical spatial filter. Top row far right: mean frame. Middle row left: fluorescence trace using the boxcar spatial filter. Bottom row left: fast filter output using the boxcar spatial filter. Middle row right: fluorescence trace using true spatial filter. Bottom right: fast filter output using true spatial filter. Simulation details: $\vec{\alpha} = \mathcal{N}(\boldsymbol{0}, 2\boldsymbol{I}) - 1.1\mathcal{N}(\boldsymbol{0}, 2.5\boldsymbol{I})$ where $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ indicates a two-dimensional Gaussian with mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$, $\boldsymbol{\beta} = 1$, $\tau = 0.85$ sec, $\lambda = 5$ Hz.

overlapping, making the segmentation problem more difficult. Therefore, it is desirable to have an ability to crudely segment, yielding only a few neurons in each ROI, and then spatially filter within each ROI to pick out the spike trains from each neuron. This may be achieved in a principled manner by generalizing the model as described in section 2.6. Figure 9 shows how this approach can separate the two signals, assuming that the spatial filters of the two neurons are known.

Typically, the true spatial filters of the neurons in the ROI will be unknown, and thus, must be estimated from the data. This problem may be considered a special case of blind source separation [2, 22]. Figure 10 shows that multiple signals can be separated, with reasonable assumptions on correlations between the signals, and SNR. Note that separation occurs even though the signal is overlapping in several pixels (top left panel), leading to a "bleed-through" effect in the one-dimensional fluorescence projections (bottom left panel). The inferred filters (top middle and right panels) are not the true filters, but rather, their sum is linearly related to the sum of the true filters. Regardless, the inferred spike trains are well separated (bottom middle and right panels).

Figure 9: Simulation showing that even when two neurons' spatial filters are overlapping, one can separate the two spike trains by spatial filtering. Top left panel: mean frame from the movie. Bottom left: optimal one-dimensional fluorescence projections for the neuron 1 (black line) and neuron 2 (gray line), and their respective spike trains (black and gray '+' symbols, respectively). Top middle panel: the true spatial filter for neuron 1. Bottom middle panel: inferred (black line) and true (black '+' symbols) spike trains. Top right panel: the true spatial filter for neuron 2. Bottom right panel: inferred (gray line) and true (gray '+' symbols) spike trains. Simulation details: $\vec{\alpha}^1 = \mathcal{N}([-1.8, 1.8]^\mathsf{T}, 2\boldsymbol{I})$ $\vec{\alpha}^2 = \mathcal{N}([1.8, -1.8]^\mathsf{T}, 5\boldsymbol{I})$, $\boldsymbol{\beta} = [1, 1]^\mathsf{T}$, $\tau = [0.5, 0.5]^\mathsf{T}$ sec, $\lambda = [1.5, 1.5]^\mathsf{T}$ Hz.

Figure 10: Simulation showing that even when two neurons' spatial filters are largely overlapping, spatial filters that together are linearly related to the true spatial filters can be inferred, to separate the two signals. Simulation details as above. Note that the spatial fields are sufficiently overlapping to cause significant "bleed-through" between the two signals. In particular, this is clear from the rise in the black line in the first few seconds of the bottom left panel, which should be fluorescence due to neuron 1, but is in fact due to spiking from neuron 2. Regardless, the spike trains are accurately inferred here. Simulation parameters as in Figure 9.

4 Discussion

This work describes an algorithm that approximates the *maximum a posteriori* (MAP) spike train, given a calcium fluorescence movie. The approximation is required because finding the actual MAP estimate is not currently computationally tractable. Replacing the assumed Poisson distribution on spikes with an exponential distribution yields a log-concave optimization problem, which can be solved using standard gradient ascent techniques (such as Newton-Raphson). This exponential distribution has an advantage over a Gaussian distribution by restricting spikes to be positive, which improves inference quality (cf. Figure 2), and is a better approximation to a Poisson distribution with low rate. Furthermore, by utilizing the special banded structure of the Hessian matrix of the log-posterior, this approximate MAP spike train can be inferred fast enough on standard computers to use it for online analyses. Finally, all the parameters can be estimated from only the fluorescence observations, obviating the need for joint electrophysiology and imaging (cf. Figure 3). This approach is robust, in that it works "out-of-the-box" on all the *in vivo* and *in vitro* data analyzed (cf. Figure 4).

Ideally, one could compute the full joint posterior of entire spike trains, conditioned on the fluorescence data. This distribution is analytically intractable, due to the Poisson assumption on spike trains. A Bayesian approach could use Markov Chain Monte Carlo methods to recursively sample spikes until a whole sample spike train is obtained [1, 36]. Because a central aim here was computational expediency, a "greedy" approach is natural: i.e., recursively sample the most likely spike, update the posterior, and repeat until the posterior stops increasing. Template matching, projection pursuit regression [5], and matching pursuit [18] are examples of such a greedy approach (Greenberg et al's algorithm [8] could also be considered a special case of such a greedy approach). Both the greedy methods, and the one developed here, aim to optimize a similar objective function. While greedy methods reduce the computational burden by restricting the search space of spike trains, here analytic approximations are made. The advantage of the greedy approaches relative to this one is that they result in a spike train (ie, a binary sequence). However, because of the numerical approximations and restrictions, one can never be sure whether the algorithm finds the *most* likely possible spike train. On the other hand, the approach developed herein is guaranteed to quickly find the most likely spike "train", but now the inferred spike train allows for partial spikes. One interesting future direction might be to explore whether greedy methods could be improved by initializing with a thresholded version of the fast filter output.

Further, the fast filter is based on a biophysical model capturing key features of the data, and may therefore be straightforwardly generalized in several ways to improve accuracy. Unfortunately, some of these generalizations do not improve inference accuracy, perhaps because of the exponential approximation. Instead, the fast filter output can be used to initialize the more general SMC filter [34], to further improve inference quality (cf. Figure 7). Another model generalization allows incorporation of spatial filtering of the raw movie into this approach (cf. Figure 8).

A number of extensions follow from this work. First, pairing this filter with a crude but automatic segmentation tool to obtain ROIs would create a completely automatic algorithm that converts raw movies of populations of neurons into populations of spike trains. Second, combining this algorithm with recently developed connectivity inference algorithms on this kind of data [36], could yield very efficient connectivity inference.

A   Pseudocode

B   Wiener Filter

The Poisson distribution in Eq. (4) can be replaced with a Gaussian instead of a Poisson distribution, ie, $n_t \overset{iid}{\sim} \mathcal{N}(\lambda\Delta, \lambda\Delta)$, which, when plugged into Eq. (7) yields:

$$\widehat{\boldsymbol{n}} = \underset{n_t}{\operatorname{argmax}} \sum_{t=1}^{T} \left(\frac{1}{2\sigma^2}(F_t - \alpha(C_t + \beta))^2 + \frac{1}{2\lambda\Delta}(n_t - \lambda\Delta)^2 \right). \tag{39}$$

Note that since fluorescence integrates over Δ, it makes sense that the mean scales with Δ. Further, since the Gaussian here is approximating a Poisson with high rate [32], the variance should scale with the mean. Using the same tridiagonal trick as above, Eq. (11b) can be solved using Newton-Raphson once (because this expression is quadratic in $\boldsymbol{n}$). Writing the above in matrix notation and substituting $C_t - \gamma C_{t-1}$ for n_t, yields:

$$\widehat{\boldsymbol{C}} = \underset{\boldsymbol{C}}{\operatorname{argmax}} -\frac{1}{2\sigma^2} \|\boldsymbol{F} - \boldsymbol{C}\|^2 - \frac{1}{2\lambda\Delta} \|\boldsymbol{M}\boldsymbol{C} - \lambda\Delta\boldsymbol{1}\|^2, \tag{40}$$

which is quadratic in $\boldsymbol{C}$. The gradient and Hessian are given by:

$$\boldsymbol{g} = -\frac{1}{\sigma^2}(\boldsymbol{C} - \boldsymbol{F}) - \frac{1}{\lambda\Delta}((\boldsymbol{M}\widehat{\boldsymbol{C}})^\mathsf{T}\boldsymbol{M} + \lambda\Delta\boldsymbol{M}^\mathsf{T}\boldsymbol{1}), \tag{41}$$

$$\boldsymbol{H} = \frac{1}{\sigma^2}\boldsymbol{I} + \frac{1}{\lambda\Delta}\boldsymbol{M}^\mathsf{T}\boldsymbol{M}. \tag{42}$$

Algorithm 1 Pseudocode for inferring the approximately most likely spike train, given fluorescence data. Note that $\xi_i \ll 1$ for $i \in \{1, 2\}$; the algorithm is robust to small variations in each. The equations listed below refer to the most general equations in the text (simpler equations could be substituted when appropriate). Curly brackets, $\{\cdot\}$, indicate comments.

1: initialize parameters, $\boldsymbol{\theta}$ (section 2.4.1)
2: **while** convergence criteria not met **do**
3:      **for** $z = 1, 0.1, 0.01, \ldots, \xi_1$ **do** {interior point method to find $\widehat{\boldsymbol{C}}$}
4:          Initialize $n_t = \xi_2$ for all $t = 1, \ldots, T$, $C_1 = 0$ and $C_t = \gamma C_{t-1} + n_t$ for all $t = 2, \ldots, T$.
5:          let $\boldsymbol{C}_z$ be the initialized calcium, and $\mathcal{P}_z$, be the posterior given this initialization
6:          **while** $\mathcal{P}_{z'} < \mathcal{P}_z$ **do** {Newton-Raphson with backtracking line searches}
7:              compute $\boldsymbol{g}$ using Eq. (26)
8:              compute $\boldsymbol{H}$ using Eq. (27)
9:              compute $\boldsymbol{d}$ using $\boldsymbol{H} \backslash \boldsymbol{g}$ {(block-) tridiagonal Gaussian elimination}
10:             let $\boldsymbol{C}_{z'} = \boldsymbol{C}_z + s\boldsymbol{d}$, where s is between 0 and 1, and $\mathcal{P}_{z'} > \mathcal{P}_z$ {backtracking line search}
11:         **end while**
12:     **end for**
13:     check convergence criteria
14:     update $\vec{\alpha}$ using Eq. (33) {only if spatial filtering}
15:     update $\vec{\beta}$ using Eq. (34)
16:     let σ be the root-mean square of the residual
17:     let $\lambda = \frac{1}{T} \sum_t \widehat{n}_t$
18: **end while**

Note that this solution is the optimal linear solution, under the assumption that spikes follow a Gaussian distribution, and is often referred to as the Wiener filter, regression with a smoothing prior, or ridge regression [3]. Estimating the parameters for this model follows similarly as described in section 2.4.

References

[1] C. Andrieu, É. Barat, and A. Doucet. Bayesian deconvolution of noisy filtered point processes. *IEEE Transactions on Signal Processing*, 49(1):134–146, 2001.

[2] A. J. Bell and T. J. Sejnowski. An information-maximisation approach to blind separation and blind deconvolution. *Neural Computation*, 7(6):1004, 1995.

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

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

[5] J. H. Friedman and W. Stuetzle. Projection Pursuit Regression. *J. AM. STAT. ASSOC.*, 76(376):817–823, 1981.

[6] O. Garaschuk, O. Griesbeck, and A. 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.

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

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

[9] T. F. Holekamp, D. Turaga, and T. 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.

[10] R. Horn and C. Johnson. *Matrix analysis*. Cambridge Univ Pr, 1990.

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

[12] Y. Ikegaya, G. Aaron, R. Cossart, D. Aronov, I. Lampl, D. Ferster, and R. Yuste. Synfire chains and cortical songs: temporal modules of cortical activity. *Science*, 304(5670):559–564, Apr 2004.

[13] R. Kass and A. Raftery. Bayes Factors. *Journal of the American Statistical Association*, 90(430):773–795, 1995.

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

[15] Y. Lin, D. D. Lee, and L. K. Saul. Nonnegative deconvolution for time of arrival estimation. *International Conference on Acoustics, Speech, and Signal Processing*, 2004.

[16] L. Luo, E. M. Callaway, and K. Svoboda. Genetic dissection of neural circuits. *Neuron*, 57(5):634–660, Mar 2008.

[17] J. N. MacLean, B. O. Watson, G. B. Aaron, and R. Yuste. Internal dynamics determine the cortical response to thalamic stimulation. *Neuron*, 48(5):811–823, Dec 2005.

[18] S. Mallat and Z. Zhang. Matching pursuit with time-frequency dictionaries: IEEE Trans. *Signal Processing*, 41:3397–3415, 1993.

[19] M. Mank, A. F. Santos, S. Direnberger, T. D. Mrsic-Flogel, S. B. Hofer, V. Stein, T. Hendel, D. F. Reiff, C. Levelt, A. Borst, T. Bonhoeffer, M. Hbener, and O. Griesbeck. A genetically encoded calcium indicator for chronic in vivo two-photon imaging. *Nat Methods*, 5(9):805–811, Sep 2008.

[20] B. Mao, F. Hamzei-Sichani, D. Aronov, R. Froemke, and R. Yuste. Dynamics of spontaneous activity in neocortical slices. *Neuron*, 32(5):883–98, 2001.

[21] J. Markham and J.-A. Conchello. Parametric blind deconvolution: a robust method for the simultaneous estimation of image and blur. *Journal of The Optical Society Of America A. Optics, Image Science, and Vision*, 16(10):2377–2391, Oct 1999.

[22] E. A. Mukamel, A. Nimmerjahn, and M. J. Schnitzer. Automated analysis of cellular signals from large-scale calcium imaging data. *Neuron*, 63(6):747–760, Sep 2009.

[23] S. Nagayama, S. Zeng, W. Xiong, M. L. Fletcher, A. V. Masurkar, D. J. Davis, V. A. Pieribone, and W. R. Chen. In vivo simultaneous tracing and Ca^{2+} imaging of local neuronal circuits. *Neuron*, 53(6):789–803, Mar 2007.

[24] O'Grady, Paul D. and Pearlmutter, Barak 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.

[25] L. Paninski, Y. Ahmadian, D. Ferreira, S. Koyama, K. R. Rad, M. Vidne, J. Vogelstein, and W. Wu. A new look at state-space models for neural data. *J Comput Neurosci*, Aug 2009.

[26] T. A. Pologruto, R. Yasuda, and K. Svoboda. Monitoring neural activity and $[Ca^{2+}]$ with genetically encoded Ca^{2+} indicators. *J Neurosci*, 24(43):9572–9579, Oct 2004.

[27] 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.

[28] W. Press, S. Teukolsky, W. Vetterling, and B. Flannery. *Numerical recipes in C*. Cambridge University Press, 1992.

[29] V. Rokhlin, A. Szlam, and M. Tygert. A randomized algorithm for principal component analysis. *SIAM J. Matrix Anal. Appl*, 31:1100–1124, 2009.

[30] T. Sasaki, N. Takahashi, N. Matsuki, and Y. Ikegaya. Fast and accurate detection of action potentials from somatic calcium fluctuations. *Journal of Neurophysiology*, 100(3):1668, Jul 2008.

[31] T. Schwartz, D. Rabinowitz, V. K. Unni, V. S. Kumar, D. K. Smetters, A. Tsiola, and R. Yuste. Networks of coactive neurons in developing layer 1. *Neuron*, 20:1271–1283, 1998.

[32] L. Sjulson and G. Miesenböck. Optical recording of action potentials and other discrete physiological events: a perspective from signal detection theory. *Physiology (Bethesda)*, 22:47–55, Feb 2007.

[33] D. Smetters, A. Majewska, and R. Yuste. Detecting action potentials in neuronal populations with calcium imaging. *Methods*, 18(2):215–221, Jun 1999.

[34] J. T. Vogelstein, B. O. Watson, A. M. Packer, R. Yuste, B. Jedynak, and L. Paninski. Spike inference from calcium imaging using sequential monte carlo methods. *Biophys J*, 97(2):636–655, Jul 2009.

[35] D. J. Wallace, S. M. zum Alten Borgloh, S. Astori, Y. Yang, M. Bausen, S. Kgler, A. E. Palmer, R. Y. Tsien, R. Sprengel, J. N. D. Kerr, W. Denk, and M. T. Hasan. Single-spike detection in vitro and in vivo with a genetic Ca2+ sensor. *Nat Methods*, 5(9):797–804, Sep 2008.

[36] M. Y, V. JT, and P. L. A Bayesian approach for inferring neuronal connectivity from calcium fluorescent imaging data. *Annals of Applied Statistics*, in press, 2009.

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

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