Optimal and efficient deconvolution for non-negative linearly filtered signals, with applications to neural fluorescence data

Anonymous Author(s)
Affiliation
Address
email

Abstract

1 Introduction

Non-negative deconvolution [**?**] and matrix factorization [**?**, **?**] arise in a number of different scenarios, including from audio signals processing [**?**] and image processing. We are interested in dealing with a subset of these problems. In particular, we assume the existence of a signal of interest, $\boldsymbol{z} = z_0, \ldots, z_T$, where each $z_t \in \mathcal{R}_+$ is a scalar non-negative number, which is then filtered by a series of linear ordinary differential equations, of the form:

$$\boldsymbol{x}_{t+1} = \boldsymbol{D}(\boldsymbol{x}_t + \boldsymbol{x}_b) + \mathbf{1}z_t \tag{1}$$

where $\boldsymbol{x}_t \in \mathcal{R}^{N\times 1}$ is an N dimensional column vector, $\boldsymbol{D} \in \mathcal{R}^{N\times N}$ is a differential operator matrix that updates $\boldsymbol{x}_t$, $\boldsymbol{x}_b$ is the baseline resting value for $\boldsymbol{x}_t$, and $\mathbf{1}$ is a N dimensional column vector of ones. Observations of this system, $\boldsymbol{y}_t \in \mathcal{R}^{M\times 1}$, are linear-Gaussian functions of $\boldsymbol{x}_t$:

$$\boldsymbol{y}_t = \boldsymbol{A}\boldsymbol{x}_t + \boldsymbol{b} + \boldsymbol{\varepsilon}_t, \qquad \boldsymbol{\varepsilon}_t \sim \mathcal{N}(0, \boldsymbol{I}) \tag{2}$$

where $\boldsymbol{A} \in \mathcal{R}^{M\times N}$ is a scaling matrix, $\boldsymbol{b} \in \mathcal{R}^{M\times 1}$ is an offset vector, and $\varepsilon_t \in \mathcal{R}^{M\times 1}$ is a standard multivariate normal random variable. Models characterized by (1) and (2) arise in a number of contexts, including several in neuroscience [**?**].

It is often of interest, given such a model, to find either the trajectory $\boldsymbol{z}$ that makes the observations most likely, i.e., maximum likelihood estimate, $\widehat{\boldsymbol{z}}_{MLE}$, or the most likely trajectory $\boldsymbol{z}$, given the observations, i.e., the maximum *a posteriori* estimate, $\widehat{\boldsymbol{z}}_{MAP}$:

$$\widehat{\boldsymbol{z}}_{MLE} = \underset{z_t \geq 0 \forall t}{\operatorname{argmax}}\, p(\boldsymbol{y}|\boldsymbol{z}) \tag{3}$$

$$\widehat{\boldsymbol{z}}_{MAP} = \underset{z_t \geq 0 \forall t}{\operatorname{argmax}}\, p(\boldsymbol{z}|\boldsymbol{y}) = \underset{\boldsymbol{x}_t \geq 0 \forall t}{\operatorname{argmax}}\, \frac{1}{p(\boldsymbol{y})} p(\boldsymbol{y}|\boldsymbol{z})p(\boldsymbol{z}) = \underset{z_t \geq 0 \forall t}{\operatorname{argmax}}\, p(\boldsymbol{y}|\boldsymbol{z})p(\boldsymbol{z}) \tag{4}$$

where $p(\boldsymbol{z})$ is a prior distribution on $\boldsymbol{z}$, the second equality in (4) follows from a couple applications of Bayes rule, and the third equality follows from the fact that $p(\boldsymbol{y})$ simply scales the result, and

may thus be dropped. Unfortunately, (4) is typically not a concave problem, so solving it directly may be very time consuming.

In Section 2 we described a highly efficient and optimal non-negative filter for solving (4), by making use of two techniques. We use a barrier technique for dealing with the non-negativity constraint, in which we iteratively solve a series of related log-concave problems. When the likelihood to be maximized is twice differentiable, we can use an efficient algorithm to invert the Hessian, based on the assumptions implicit in (1). In Section **??**, we apply this approach to an application in neuroscience, specifically, inferring spike trains from noisy calcium-sensitive fluorescence observations. Finally, in Section 6, we compare the results of this filter with others, and discuss various possibilities for future work.

2 Methods

2.1 Optimal Non-negative filter

Although the approach developed here is sufficiently general to be applicable to any system characterized by (1) and (2), we are particularly interested in dealing with the problem of inferring sparse spike trains from calcium sensitive fluorescence measurements. As such, we provide details for the method only in terms of our model of that situation. In particular, we assume the following model relating spikes, n, intracellular calcium concentration, $[\text{Ca}^{2+}]$, and fluorescence measurements, F:

$$F_t = [\text{Ca}^{2+}]_t + \varepsilon_t, \qquad \varepsilon_t \sim \mathcal{N}(0,1) \tag{5}$$

$$\tau \frac{[\text{Ca}^{2+}]_t - [\text{Ca}^{2+}]_{t-1}}{\Delta} = -[\text{Ca}^{2+}]_{t-1} + [\text{Ca}^{2+}]_b + \omega n_t \tag{6}$$

$$n_t \sim Poisson(\lambda_t \Delta) \tag{7}$$

where ε_t is a standard normal random variable, τ is the time constant for calcium decay, Δ is the time step size (usually taken to be the duration of an image, i.e., $\Delta = 1/(\text{frame rate})$), $[\text{Ca}^{2+}]_0$ is the baseline calcium concentration, and spikes are Poisson random variables with rate $\lambda_t \Delta$ (the Δ ensures that the rate does not change with temporal discretization). Fig. 1 depicts an exemplary simulation. It should be clear to the reader that (5) – (7) adheres to the constraints imposed by (1) and (2) by noting the following substitutions:

$$\boldsymbol{D} = 1 - \Delta/\tau, \quad \boldsymbol{A} = 1, \quad \boldsymbol{b} = 0, \quad \boldsymbol{x}_b = \frac{[\text{Ca}^{2+}]_b/\omega}{\boldsymbol{D}}, \quad z_t = n_t, \quad \boldsymbol{x}_t = \frac{\tau}{\omega\Delta}[\text{Ca}^{2+}]_t, \quad \boldsymbol{y}_t = F_t$$

Given such a model, our goal is to find the most likely spike train, $\boldsymbol{n}$, given the fluorescence measurements, $\boldsymbol{F}$:

$$\widehat{\boldsymbol{n}}_{MAP} = \underset{n_t \geq 0 \forall t}{\operatorname{argmax}} P(\boldsymbol{n}|\boldsymbol{F}) = \underset{n_t \geq 0 \forall t}{\operatorname{argmax}} P(\boldsymbol{F}|\boldsymbol{n}) P_{\boldsymbol{\theta}}(\boldsymbol{n}) \tag{8a}$$

$$= \underset{n_t \geq 0 \forall t}{\operatorname{argmax}} \prod_{t=1}^{T} P_{\boldsymbol{\theta}}(F_t|n_t) P(n_t) = \underset{n_t \geq 0 \forall t}{\operatorname{argmax}} \sum_{t=1}^{T} \log P(F_t|n_t) + \log P(n_t) \tag{8b}$$

$$= \underset{n_t \geq 0 \forall t}{\operatorname{argmax}} \sum_{t=1}^{T} \log \mathcal{N}(F_t; [\text{Ca}^{2+}]_t, 1) + \log Poisson(n_t; \lambda_t \Delta) \tag{8c}$$

Solving (8) exactly is problematic because it requires searching over the 2^T possible sequences of $\boldsymbol{n}$. We therefore approximate the Poisson distribution with an exponential distribution, which is an excellent approximation assuming that the rate is relatively low. While this approximation makes things easier, the "sharp" threshold imposed by the non-negativity constraint makes the argument not log-concave. We therefore take a "barrier" approach, in which we drop the sharp threshold, and

add a barrier term, $f(n_t)$, which must approach $-\infty$ as n_t approaches zero (e.g., $f(n_t) = -\log n_t$). By iteratively reducing the weight of the barrier term, η, we are guaranteed to converge to the correct solution [**?**]. Thus, our goal is to efficiently solve:

$$\widehat{\boldsymbol{n}}_\eta = \underset{n_t \forall t}{\operatorname{argmin}} \sum_{t=1}^{T} (F_t - [\mathrm{Ca}^{2+}]_t)^2 + \lambda_t \Delta n_t - \eta \log n_t \tag{9}$$

To efficiently solve (11), we first note that $\boldsymbol{z}$ may be written as a linear function of $\boldsymbol{x}$:

$$\boldsymbol{z} = (\mathbf{1}'\mathbf{1})^{-1}\mathbf{1}'(\boldsymbol{x}_{t-1} - \boldsymbol{D}\boldsymbol{x}_t) = \begin{bmatrix} -\boldsymbol{D} & 0 & 0 & \cdots \\ 1 & -\boldsymbol{D} & 0 & \cdots \\ \vdots & & & \\ 0 & \cdots & 1 & -\boldsymbol{D} \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_1 \\ \boldsymbol{x}_2 \\ \vdots \\ \boldsymbol{x}_T \end{bmatrix} = \boldsymbol{M}\boldsymbol{x} \tag{10}$$

where $\boldsymbol{M} \in \mathcal{R}^{T \times T}$ is a bidiagonal matrix. Now, we could solve (4) in terms of $\boldsymbol{x}$ instead of $\boldsymbol{z}$, or for our model, we could solve (11) in terms of $[\mathrm{Ca}^{2+}]_t$ instead of n_t:

$$\widehat{\boldsymbol{C}}_\eta = \underset{[\mathrm{Ca}^{2+}]_t \geq 0 \forall t}{\operatorname{argmin}} \sum_{t=1}^{T} (F_t - [\mathrm{Ca}^{2+}]_t)^2 + \lambda_t \Delta n_t - \eta \log n_t \tag{11}$$

where we have let $\boldsymbol{C} = [\mathrm{Ca}^{2+}]_1, \ldots, [\mathrm{Ca}^{2+}]_T$, which is a typical log-concave optimization problem. This may be apparent by rewriting (11) in vector notation:

$$\boldsymbol{C}_\eta = \underset{[\mathrm{Ca}^{2+}]_t \geq 0 \forall t}{\operatorname{argmin}} \|\boldsymbol{F} - \boldsymbol{C}\|_2^2 + \Delta\boldsymbol{\lambda}'\boldsymbol{n} - \eta \log(\mathbf{1}'\boldsymbol{n}) \tag{12}$$

As (12) is concave, we may use any gradient descent technique to solve it. We elect to use the Newton-Raphson technique, which requires the analytic computation of both the gradient $\boldsymbol{g}$ and Hessian $\boldsymbol{H}$ of the likelihood term in (12) [**?**]. Letting $\boldsymbol{M}\boldsymbol{C} = \boldsymbol{n}$, we have:

$$\boldsymbol{g} = -2(\boldsymbol{F} - \boldsymbol{C}) + \Delta\boldsymbol{\lambda}'\boldsymbol{M} - \eta\boldsymbol{M}'(\boldsymbol{M}\boldsymbol{C})^{-1} \tag{13}$$

$$\boldsymbol{H} = 2\boldsymbol{I} + 2\eta\boldsymbol{M}'(\boldsymbol{M}\boldsymbol{C})^{-2}\boldsymbol{M} \tag{14}$$

Having $\boldsymbol{g}$ and $\boldsymbol{H}$, we may plug them into the Newton-Raphson update step:

$$\boldsymbol{C}_\eta \leftarrow \boldsymbol{C}_\eta + s\boldsymbol{H}^{-1}\boldsymbol{g} \tag{15}$$

where s is the step size, and $\boldsymbol{H}^{-1}$ is the inverse Hessian. While typically, inverting the Hessian takes $O(T^3)$ time, the filtering of $\boldsymbol{n}$ by a set of linear ordinary differential equations implies that the Hessian is actually tridiagonal (or, for the more general case, block tridiagonal). Therefore, we use the fact that the inverse of block-tridiagonal matrices may be computed in $O(T)$ time, which greatly expedites each Newton-Raphson step.

Having computed the step direction, $\boldsymbol{H}^{-1}\boldsymbol{g}$, we must now compute the step size, s. The step size is chosen to satisfy two constraints. First, upon stepping, the likelihood must *increase*. Second, the step must be small enough such that $\boldsymbol{n}$ remains non-negative:

$$\boldsymbol{M}(\boldsymbol{C} + s\boldsymbol{H}^{-1}\boldsymbol{g}) \geq 0 \Rightarrow s \geq -\boldsymbol{M}\boldsymbol{C}\big(\boldsymbol{M}(\boldsymbol{H}^{-1}\boldsymbol{g})\big)^{-1} \tag{16}$$

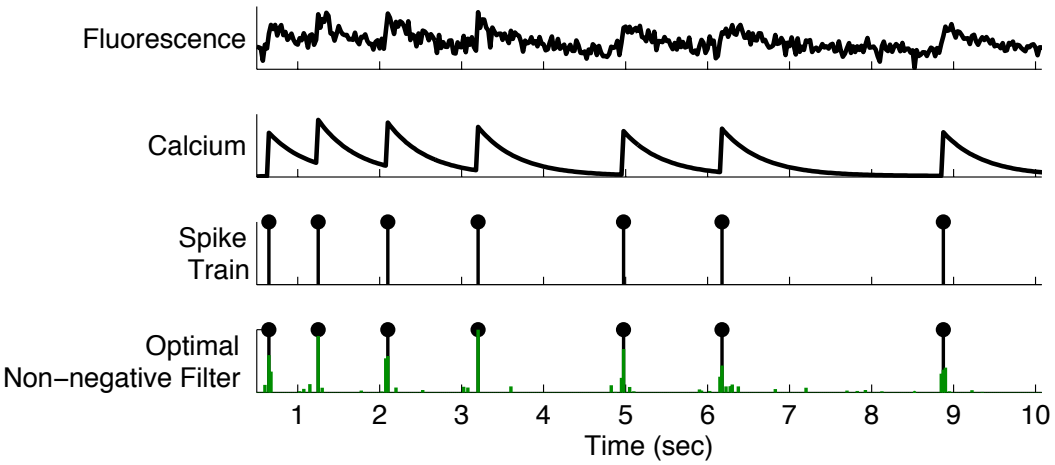

Figure 1: Schematic demonstrating our neuron model and inference. Top panel: Fluorescence measurements. Second panel: Intracellular calcium concentration. Third panel: Spike train. Fourth panel: Output of optimal non-negative filter. Parameters: $\lambda = 10$, $\Delta = 0.005$ msec, $A = 1$, $\tau = 0.5$ sec, $[\text{Ca}^{2+}]_0 = 0.1$, $\sigma = 0.8$.

Thus, s is initialized at 1, and if either of these two constraints are not satisfied, then s is reduced until they both are. Substituting (13) and (14) into (15), and ensuring that s satisfies its constraints, one can solve for $\widehat{\boldsymbol{C}}_\eta$. Iteratively repeating this process while reducing η yields $\widehat{\boldsymbol{C}}_{MAP}$.

Pseudocode for an efficient implementation of this approach is provided in Algorithm (1). Fig. 1 shows how this method performs on an example fluorescence signal simulated according to (5)–(7).

Algorithm 1 Pseudocode for implementing the optimal non-negative filter for an exponential prior

while $\eta > \epsilon$ **do**
  Initialize $\boldsymbol{z}$
  $\mathcal{L}_i = \sum_t (\boldsymbol{y}_t - \boldsymbol{A}\boldsymbol{x}_t - \boldsymbol{b})\boldsymbol{I}^{-1}(\boldsymbol{y}_t - \boldsymbol{A}\boldsymbol{x}_t - \boldsymbol{b}) + \Delta\lambda_t z_t - \eta \log z_t$
  while $\mathcal{L}_i < \mathcal{L}_{i-1}$ **do**
    $\boldsymbol{g} = -2(\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}) - \Delta\boldsymbol{\lambda}'\boldsymbol{M} - \eta\boldsymbol{M}'(\boldsymbol{z}^{-1})$
    $\boldsymbol{H} = 2\boldsymbol{I} + 2\eta\boldsymbol{M}'(\boldsymbol{z}^{-2})\boldsymbol{M}$
    Compute $\boldsymbol{d} = \boldsymbol{H}^{-1}\boldsymbol{g}$ efficiently
    Choose s such that
      $s \geq -\boldsymbol{z}(\boldsymbol{M}\boldsymbol{d})^{-1}$
      and $\mathcal{L}_i < \mathcal{L}_{i-1}$
    Let $\boldsymbol{x} \leftarrow \boldsymbol{x} + s\boldsymbol{d}$
    $i \leftarrow i + 1$
  end while
  reduce η
end while

3 Learning the parameters

All the above approaches assume the parameters governing our model are known. In general, however, these parameters may be estimated from the data. To find the maximum likelihood estimator for the parameters, $\widehat{\boldsymbol{\theta}}$, we must integrate over the unknown variable, $\boldsymbol{z}$:

$$\widehat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta}}{\text{argmax}} \int p(\boldsymbol{y}|\boldsymbol{z}, \boldsymbol{\theta}) p(\boldsymbol{z}|\boldsymbol{\theta}) d\boldsymbol{z} \tag{17}$$

However, integrating over all possible $\boldsymbol{z}$ is not generally possible. Thus, one often approximates the inside integral in (18) with the most likelihood estimate of $\boldsymbol{z}$ [**?**], which we obtain using Algorithm 1:

$$\widehat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta}}{\operatorname{argmax}}\, p(\boldsymbol{y}|\widehat{\boldsymbol{z}}, \boldsymbol{\theta}) p(\widehat{\boldsymbol{z}}|\boldsymbol{\theta}) \tag{18}$$

This approximation is good whenever the likelihood is very peaky, meaning that most of the mass is around the most likely sequence, and it is flat nearly everywhere else. For models characterized by (1) and (2), $\boldsymbol{\theta} = \{\boldsymbol{D}, \boldsymbol{A}, \boldsymbol{b}\}$, and those parameters governing $p(\boldsymbol{z})$. For our spiking, calcium, fluorescence model, (5)–(7), we have $\boldsymbol{D} = 1 - \Delta/\tau$, $\boldsymbol{A} = 1$, $\boldsymbol{b} = \Delta[\mathrm{Ca}^{2+}]_0/\tau$, and $p(n_t)$ is parameterized by $\lambda_t \Delta$. Thus, for our model we have:

$$\widehat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta}}{\operatorname{argmax}}\, p(\boldsymbol{F}|\widehat{\boldsymbol{n}}, \boldsymbol{D}, \boldsymbol{A}, \boldsymbol{b}) p(\widehat{\boldsymbol{n}}|\boldsymbol{\lambda}) \tag{19}$$

which is separable into two log-concave problems: one for the parameters governing the likelihood term, $\{\boldsymbol{D}, \boldsymbol{A}, \boldsymbol{b}\}$, and the other for the parameter governing the prior term, $\{\boldsymbol{\lambda}\}$. The likelihood parameters may be estimated using standard constrained quadratic programming tools:

$$\{\widehat{\boldsymbol{D}}, \widehat{\boldsymbol{A}}, \widehat{\boldsymbol{b}}\} = \underset{\boldsymbol{D},\boldsymbol{A},\boldsymbol{b}\geq 0}{\operatorname{argmax}} \sum_t \big(F_t - \boldsymbol{A}(\boldsymbol{D}[\mathrm{Ca}^{2+}]_{t-1} + n_t + \boldsymbol{b})\big)' \boldsymbol{I}^{-1} \big(F_t - \boldsymbol{A}(\boldsymbol{D}[\mathrm{Ca}^{2+}]_{t-1} + n_t + \boldsymbol{b})\big) \tag{20a}$$

$$= \underset{\boldsymbol{D},\boldsymbol{A},\boldsymbol{b}\geq 0}{\operatorname{argmax}} (\boldsymbol{F} + \boldsymbol{G}\boldsymbol{w})' \boldsymbol{I}^{-1} (\boldsymbol{F} + \boldsymbol{G}\boldsymbol{w}) \tag{20b}$$

$$= \underset{\boldsymbol{D},\boldsymbol{A},\boldsymbol{b}\geq 0}{\operatorname{argmax}} \big(\boldsymbol{w}'\boldsymbol{G}'\boldsymbol{G}\boldsymbol{w} + 2\boldsymbol{G}'\boldsymbol{F}\boldsymbol{w}\big) \tag{20c}$$

where

$$\boldsymbol{G} = \begin{bmatrix} [\mathrm{Ca}^{2+}]_1 & n_2 & 1 \\ [\mathrm{Ca}^{2+}]_2 & n_3 & 1 \\ \vdots & \vdots & \vdots \\ [\mathrm{Ca}^{2+}]_{T-1} & n_T & 1 \end{bmatrix} \qquad \boldsymbol{w} = \begin{bmatrix} -\boldsymbol{A}\boldsymbol{D} \\ -\boldsymbol{A} \\ -\boldsymbol{A}\boldsymbol{b} \end{bmatrix} \tag{21}$$

And the parameter for the likelihood term is given simply by

$$\widehat{\lambda}_t = \frac{1}{\widehat{n}_t \Delta} \tag{22}$$

4 Comparison to other methods

Perhaps the most closely related filter to our optimal non-negative filter is the Wiener filter [**?**]. The Wiener filter differs in construction from our filter in a few ways. First, it imposes no constraint on $\boldsymbol{z}$. Second, it is optimal upon assuming that the prior distribution of $\boldsymbol{z}$ is Gaussian (as opposed to exponential in our non-negative filter). In our model, we assumed that spiking was Poisson, which is well approximated by an exponential distribution when spiking is sparse. However, when spike rates are fast — e.g. on average, several spikes per image frame — a Poisson distribution is well approximated by a Gaussian distribution. Furthermore, when spiking is fast, the mean of the Gaussian would be relatively high, and the variance proportional, so the probability of sampling a negative number would be relatively small, obviating the need for the non-negativity constraint. Fig. 2 shows two example fluorescent traces. One the left, a neuron was simulated with a rate of

1 Hz; on the right, 10 Hz. The top and second panels show the simulated fluorescence and spike train, respectively. The third panels show the performance of the optimal linear (i.e., Wiener) filter; whereas the fourth panels show the performance of the optimal non-negative filter. Both filters perform very well in the scenario when there are on average several spikes per frame.

The bottom panels show the results of a different algorithm, projection pursuit regression (PPR) [**?**]. PPR is different from both the optimal linear and non-negative filter, in a couple ways. First, it solves a related, but slightly different problem. First, instead of finding the MAP estimate of the spike train, PPR finds the MLE estimate. Second, PPR constrains the solution to have only integers. Therefore, $\boldsymbol{n}_{PPR}$ is the solution to the following optimization problem:

$$\boldsymbol{n}_{PPR} = \underset{n_t \in \mathcal{Z}}{\operatorname{argmax}} P(\boldsymbol{F}|\boldsymbol{n}) \tag{23}$$

More specifically, for this problem, PPR assumes that the signal of interest, $\boldsymbol{F}$, is a sum of exponentials times Heavyside step functions and Gaussian noise:

$$\boldsymbol{F} = \sum_{t_i} \omega e^{(t_i - t)/\tau} H(t_i - t) + \boldsymbol{\varepsilon} \tag{24}$$

where $H(x) = 1$ when $x \geq 0$ and zero otherwise, and the goal is to find the spike times, $\{t_i\}$. PPR proceeds in an iterative fashion, adding a spike with each iteration, as long as doing so reduces the residual square error. One obtains the time of the next spike to add by finding the maximum of the convolution of the residual square error from the previous iteration with the calcium kernel, $\omega e^{-t/\tau}$. This process is repeated until adding another spike increases the residual square error. In general, this procedure takes $O(T \log T)$ per iteration, but because the kernel here is exponential, the convolution requires only $O(T)$ time. So, like the optimal non-negative filter, PPR can be faster than the optimal Wiener filter. However, unlike the other two filters, the likelihood function is not concave, so we are not guaranteed to converge to the optimal solution. Nonetheless, this algorithm performs very well for simulated data, as depicted by the bottom panels of Fig. 2. Learning the parameters for this model is very similar to the optimal non-negative filter.

5 Application to in vitro data

While in simulations, all the above algorithms perform reasonably well, the true test is how well they perform given real data. Briefly, somatosensory thalamocortical slices were 450 μm thick were prepared from C57BL/6 mice as described in [**?**]. Neurons were filled with 50 μM Fura 2 pentapotassium salt through a recording pipette containing 130 K-methylsulfate, 2 MgCl2, 0.6 EGTA, 10 HEPES, 4 ATP-Mg, and 0.3 GTP-Tris, pH 7.2 (295 mOsm). After cells were fully loaded with dye imaging was done by a modified BX50-WI upright confocal microscope (Olympus, Melville, NY). Image acquisition was performed with or the C9100-12 CCD camera from Hamamatsu Photonics (Shizuoka, Japan) with arclamp illumination at 385nm and 510/60nm collection filters (Chroma, Fuerstenfeldbruck, Germany). Images were saved as tiffs and were analyzed using custom software written in Matlab (Mathworks, Natick, MA). All recordings were made using the Multiclamp 700B amplifier (Molecular Probes, Sunnyvale, CA), digitized with National Instruments 6259 multichannel cards and recod BX50-WI upright confocal microscope (Olympus, Melville, NY). All recordings were made using the Multiclamp 700B amplifier (Molecular Probes, Sunnyvale, CA), digitized with National Instruments 6259 multichannel cards and recorded using custom software written using the LabView platform (National Instruments, Austin, TX) .

In this example, although the neuron spikes with a relatively low rate, the calcium accumulates leading to fluorescence saturation. As all the abovedescribed methods are inherently linear methods, their assumptions do not adequately capture these nonlinear dynamics. While both the optimal linear and non-negative filter are still able to infer the correct spike times, the non-negative filter does so faster and with less noise. PPR, which has a sharp threshold for including an additional spike, however, performs relatively poorly, demonstrating the dependency of this method on a good model fit.

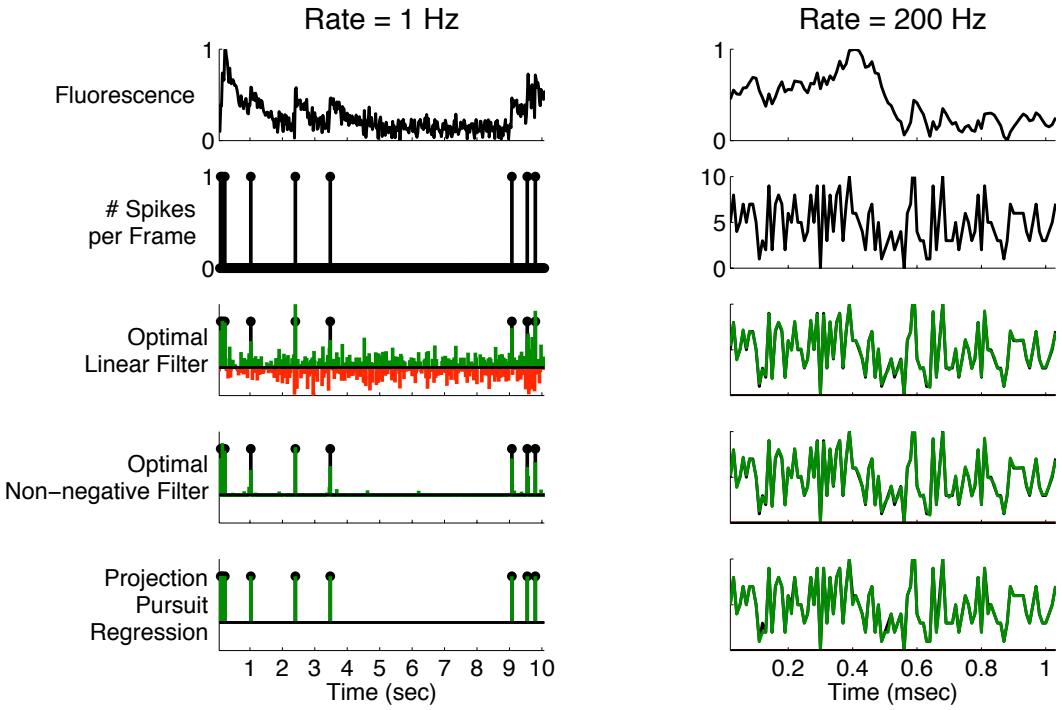

Figure 2: Comparison of various filters for a simulated Poisson neuron spiking with a rate of 1 Hz (left panels) and 200 Hz (right panels). Top panels: Fluorescence measurements. Second panels: Number of spikes per frame. Third panels: Optimal linear (i.e., Wiener) filter output given fluorescence the above fluorescence signal. Fourth panels: Same as third panels, but for the optimal non-negative filter. Fifth panels: Same as third panels, but using the Projection Pursuit Regression. Parameters: $\lambda = 10$, $\Delta = 0.005$ msec, $A = 1$, $\tau = 0.5$ sec, $[\mathrm{Ca}^{2+}]_0 = 0.1$, $\sigma = 0.8$.

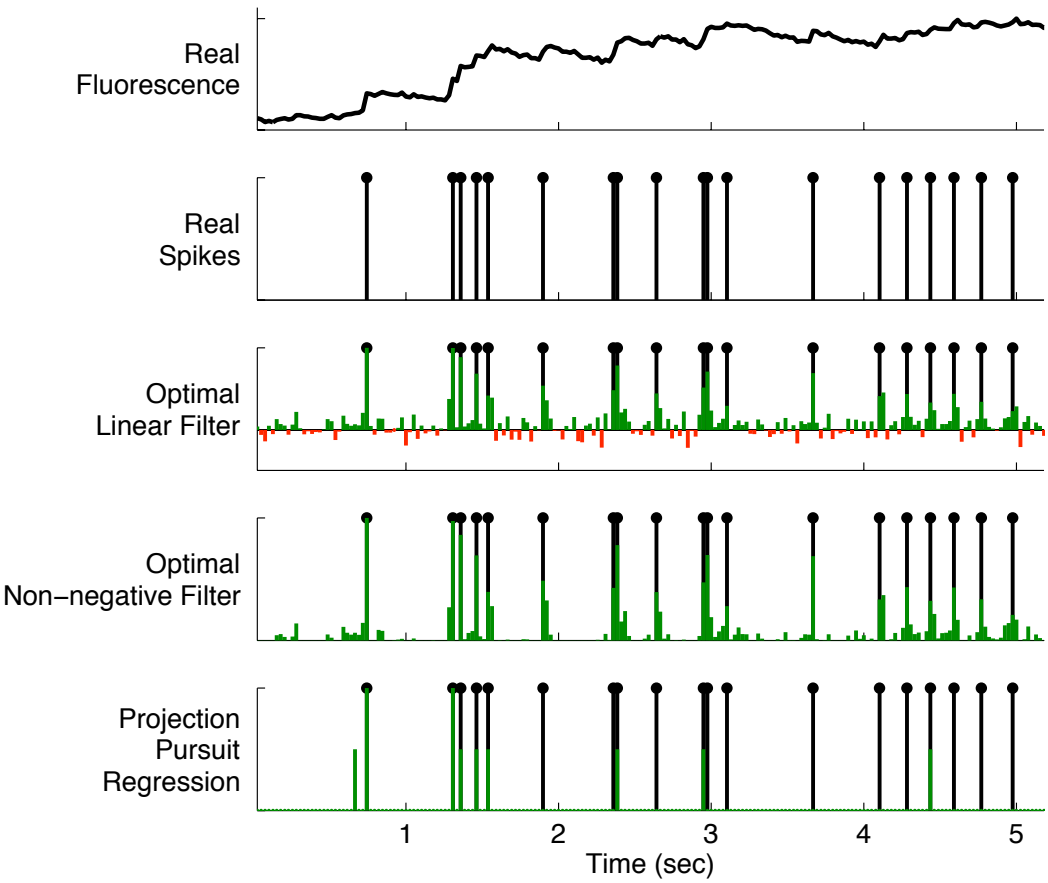

Figure 3: Inferring a spike train given fluorescence measurements. Top panel: Fluorescence measurements. Second panel: Spike train. Third panel: The output of the optimal linear filter. Fourth panel: The output of our non-negative sparse filter. Fifth panel: The output of our PPR.

6 Discussion

Limitations: saturation, errorbars, generalizations

Next steps: spatiotemporal filtering, populations

Applications: real time display

Acknowledgments

Support for JV was provided by NIDCD DC00109. LP is supported by an NSF CAREER award, by an Alfred P. Sloan Research Fellowship, and the McNight Scholar Award. BOW was supported by NDS grant F30 NS051964.

References