tapraid4/z9k-neurop/z9k-neurop/z9k00910/z9k0284d10z | xppws | S=1 | 8/4/10 | 12:14 | MS: J01073-9 | Ini: wsc

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

which is simply the root-mean-square of the residual error. Finally, the MLE of λ is given by solving

$$\hat{\lambda} = \underset{\lambda > 0}{\operatorname{argmax}} \sum_t [\ln(\lambda\Delta) - \hat{n}_t\lambda\Delta] \tag{23}$$

yields

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

Iterations stop whenever *1*) the iteration number exceeds some upper bound or *2*) the relative change in likelihood does not exceed some lower bound. In practice, parameter estimates tend to converge after several iterations, given the above-~~cited~~ initializations.

Spatial filtering

In the preceding text, 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 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 (note that we use the $\vec{X}$ throughout to indicate row vectors in space vs. X to indicate column vectors in time). 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

't' should be in subscript

BOLD X

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

where α_x corresponds to the number of photons that are contributed due to calcium fluctuations C_t, and β_x corresponds to the static photon emission at ~~each~~ pixel x. Further, the noise is assumed to be both spatially and temporally white, with ~~SD~~ σ, in each pixel (this assumption can always be approximately accurate by prewhitening; alternately, one could relax the spatial independence by representing joint noise over all pixels with a covariance matrix Σ_ε with arbitrary structure). Performing inference in this more general model proceeds in a nearly identical manner as before. In particular, the maximization, gradient, and Hessian become

standard deviation (SD)

$$\hat{C}_z = \underset{MC \geq_\odot 0}{\operatorname{argmax}} -\frac{1}{2\sigma^2}\|\vec{F} - C\vec{\alpha} - \mathbf{1}_T\vec{\beta}\|_F^2 - (MC)^T\lambda + z\ \ln_\odot(MC)^T 1 \tag{25}$$

BOLD!

$$g = (\vec{F} - C\vec{\alpha} - 1_T\vec{\beta})^T\frac{\vec{\alpha}^T}{\sigma^2} - M^T\lambda + zM^T(MC)_\odot^{-1} \tag{26}$$

$$H = -\frac{\vec{\alpha}\vec{\alpha}^T}{\sigma^2}I - zM^T(MC)_\odot^{-2}M \tag{27}$$

where $\vec{F}$ is an $N_p \times T$ element matrix, $\mathbf{1}_T$ is a column vector of ones with length T, I is an $N_p \times N_p$ identity matrix, and $\|x\|_F$ indicates the Frobenius norm, i.e., $\|x\|_2^F = \sum_{i,j} x_{i,j}^2$, and the exponents and log operator on the vector MC again indicate element-wise operations. Note that to speed up computation, one can first project the background subtracted ($N_c \times T$)-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*.

BOLD

The parameters $\vec{\alpha}$ and $\vec{\beta}$ tend to be unknown and thus must be estimated from the data. Following the strategy developed in the previous section, we first initialize the parameters. Because each voxel contains some number of fluorophores, which sets both the baseline fluorescence and the fluorescence due to calcium fluctuations, let both the initial spatial filter and initial background be the median image frame, i.e., $\hat{\alpha}_x = \hat{\beta}_x = \text{median}_t(F_{x,t})$. Given these robust initializations, the maximum likelihood estimator for each α_x and β_x is given by

$$\{\hat{\alpha}_x, \hat{\beta}_x\} = \underset{\alpha_x,\beta_x}{\operatorname{argmax}}\ P[F_x|\hat{C}] \tag{28a}$$

$$= \underset{\alpha_x,\beta_x}{\operatorname{argmax}} \sum_t \ln P[F_{x,t}|\hat{C}_t] \tag{28b}$$

$$= \underset{\alpha_x,\beta_x}{\operatorname{argmax}} \sum_t \left\{-\frac{1}{2}\ln(2\pi\sigma^2) - \frac{1}{2\sigma^2}(F_{x,t} - \alpha_x\hat{C}_t - \beta_x)^2\right\} \tag{28c}$$

$$= \underset{\alpha_x,\beta_x}{\operatorname{argmax}} - \sum_t (F_{x,t} - \alpha_x\hat{C}_t - \beta_x)^2 \tag{28d}$$

where the first equalities follow from *Eq. 1* and the last equality follows from dropping irrelevant constants. Because this is a standard linear regression problem, let $A = [\hat{C}, \mathbf{1}_T]^T$ be a $2 \times T$ element matrix and $Y_x = [\alpha_x, \beta_x]^T$ be a 2×1 element column vector. Substituting A and Y_x into *Eq. 28*d yields

$$\hat{Y}_x = \underset{Y_x}{\operatorname{argmax}} - \|F_x - A^TY_x\|_2^2 \tag{29}$$

which can be solved by computing the derivative of *Eq. 29* with respect to Y_x and setting to zero, or using Matlab notation: $\hat{Y}_x = A\backslash F_x$. Note that solving N_p two-dimensional quadratic problems is more efficient than solving a single ($2 \times N_p$)-dimensional quadratic problem. Also note that this approach does not regularize the parameters at all, by smoothing or sparsening, for instance. In the DISCUSSION we propose several avenues for further development, including the elastic net (Zou and Hastie 2005) and simple parametric models of the neuron. As in the scalar F_t case, we iterate estimating the parameters of this model $\theta = \{\vec{\alpha}, \vec{\beta}, \sigma, \gamma, \lambda\}$ and the spike train n. Because of the free scale term discussed earlier ~~in learning the parameters~~, 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 $\max(\hat{n}_t) = 1$. In practice, this procedure converged after several iterations.

Overlapping spatial filters

It is not always possible to segment the movie into pixels containing only fluorescence from a single neuron. Therefore the above-~~cited~~ 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

$$\vec{F}_t = \sum_{i=1}^{N_c} \vec{\alpha}^i C_t^i + \vec{\beta} + \vec{\varepsilon}_t \quad \vec{\varepsilon}_t \overset{iid}{\sim} \mathcal{N}(0, \sigma^2 I) \tag{30}$$

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

where each neuron is implicitly assumed to be independent and each pixel is conditionally independent and identically distributed with variance σ^2, given the underlying calcium signals. To perform inference in this more general model, let $n_t = [n_t^1, \ldots, n_t^{N_c}]$ and $C_t = [C_t^1, \ldots, C_t^{N_c}]$ be N_c-dimensional column vectors. Then, let Γ = diag $(\gamma^1, \ldots, \gamma^{N_c})$ be an $N_c \times N_c$ diagonal matrix and let I and $\mathbf{0}$ be an identity and zero matrix of the same size, respectively, yielding

BOLD