A new look at state-space models for neural data

Liam Paninski

Department of Statistics and Center for Theoretical Neuroscience
Columbia University
http://www.stat.columbia.edu/∼liam
liam@stat.columbia.edu

June 27, 2008

Support: NIH CRCNS, Sloan Fellowship, NSF CAREER, McKnight Scholar award.

State-space models

Unobserved state q_t with Markov dynamics $p(q_{t+1}|q_t)$

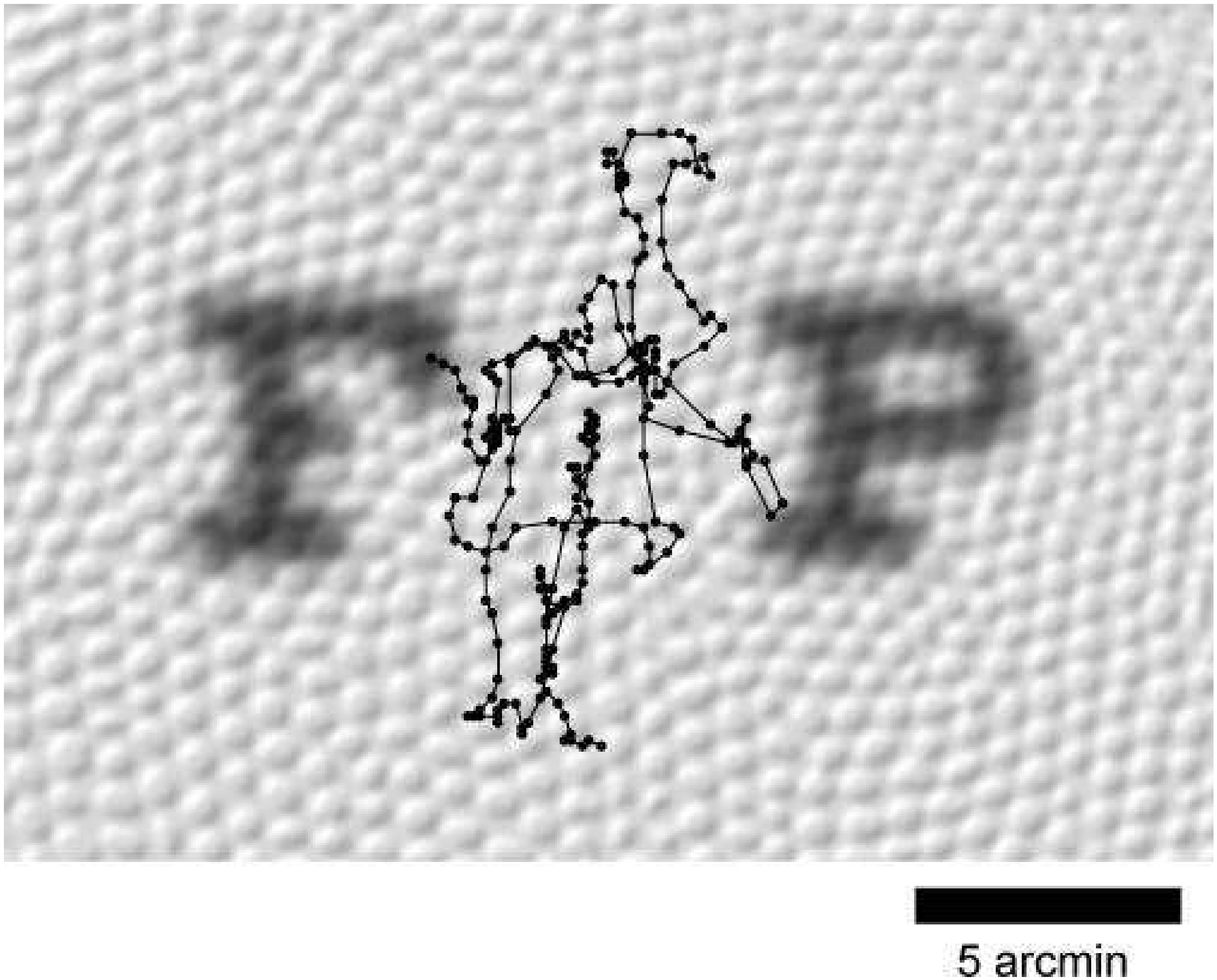
Observed y_t: $p(y_t|q_t)$

Goal: infer $p(q_t|Y_{0:T})$

Exact solutions: finite state-space HMM, Kalman filter (KF): forward-backward algorithm (recursive; $O(T)$ time)

Approximate solutions: extended KF, particle filter, etc.... basic idea: recursively update an approximation to "forward" distribution $p(q_t|Y_{0:t})$

Example: image stabilization



From (Pitkow et al., 2007): neighboring letters on the 20/20 line of the Snellen eye chart. Trace shows 500 ms of eye movement.

A state-space method for image stabilization

Assume image $I(\vec{x})$ is fixed; $\vec{q}_t$ = the (unknown) eye position.

Simple random-walk dynamics for q_t: $q_{t+1} = q_t + e$, e i.i.d.

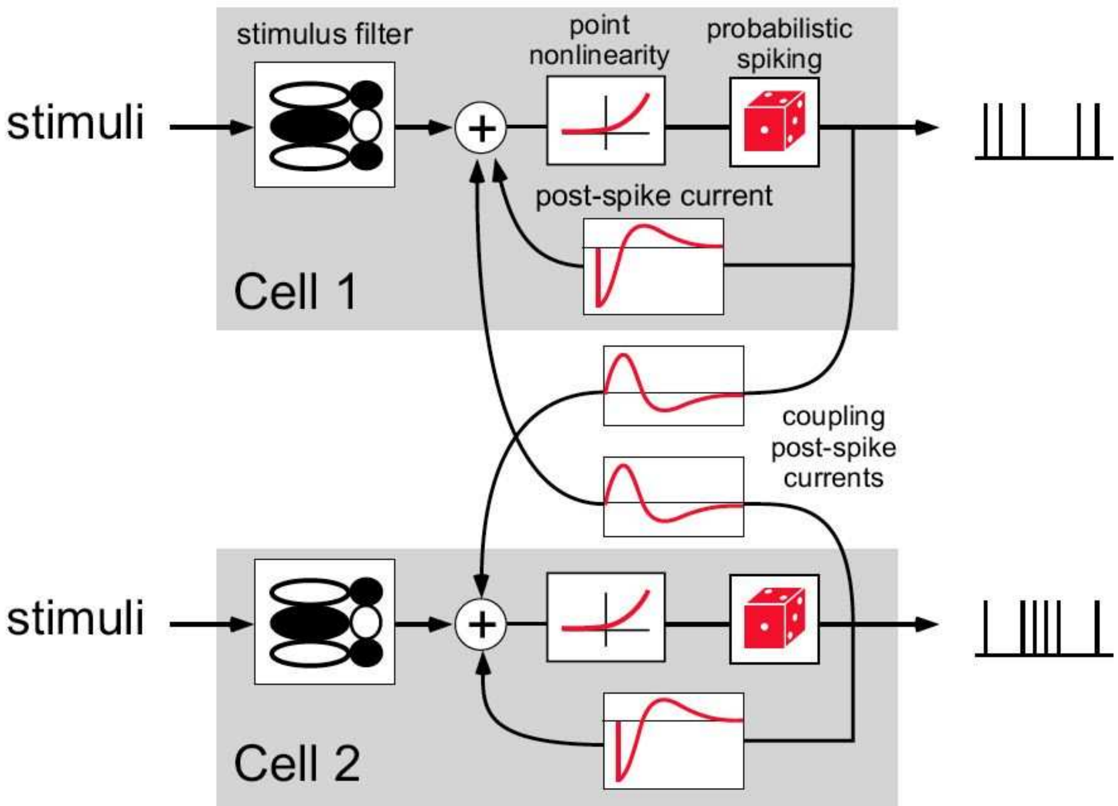
Image falling on retina at point $\vec{x}$: $I_t(\vec{x}) = I(\vec{x} - q_t)$.

Goal: infer $p(I|Y_{0:T})$. Initialize: prior $p(I)$. Now recurse:

- dynamics step:
  $p(I_t|Y_{0:t}) \to p(I_{t+1}|Y_{0:t}) = \int S_e \left[p(I_t|Y_{0:t})\right] p(e)de$ (mixture)
- observation step: $p(I_{t+1}|Y_{0:t+1}) = p(I_{t+1}|Y_{0:t})p(y_{t+1}|I_{t+1})$
- do a greedy merge to make sure number of mixture components stays bounded

Now we just need a model for $p(y_{t+1}|I_{t+1})$...

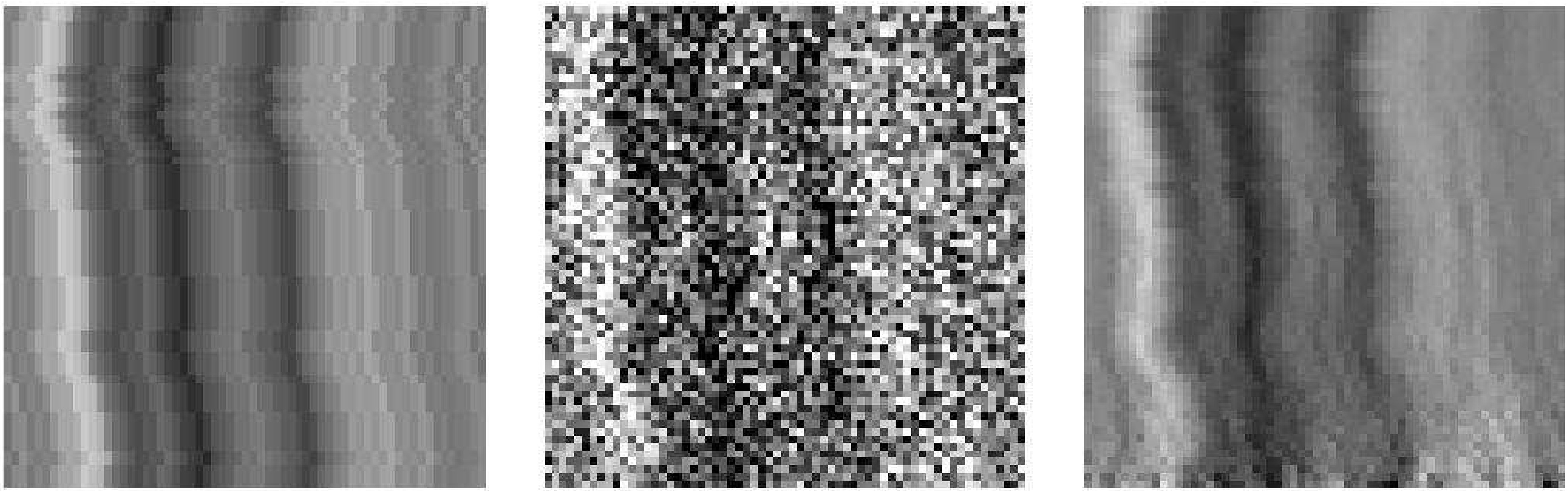
Multineuronal generalized linear model



$$\lambda_i(t) = f\left(b_i + \vec{k}_i \cdot I_t + \sum_{j,\tau} h_{i,j} n_j(t-\tau)\right); \;\; \theta = (b_i, \vec{k}_i, h_{ij})$$

— $\log p(Y|I,\theta)$ is concave in both θ and I (Pillow et al., 2008).

Simulated example: image stabilization



true image w/ translations; observed noisy retinal responses; estimated image.

Questions: how much high-frequency information can we recover? What is effect of nonlinear spiking response (Rucci et al., 2007)?

Computing the MAP path

We often want to compute the MAP estimate

$$\hat{Q} = \arg\max_{Q} p(Q|Y).$$

In standard Kalman setting, forward-backward gives MAP (because $E(Q|Y)$ and $\hat{Q}$ coincide in Gaussian case).

More generally, extended Kalman-based methods give approximate MAP, but are non-robust: forward distribution $p(q_t|Y_{0:t})$ may be highly non-Gaussian even if full joint distribution $p(Q|Y)$ is nice and log-concave.

Write out the posterior:

$$\begin{aligned}\log p(Q|Y) &= \log p(Q) + \log p(Y|Q) \\ &= \sum_t \log p(q_{t+1}|q_t) + \sum_t \log p(y_t|q_t)\end{aligned}$$

Two basic observations:

- If $\log p(q_{t+1}|q_t)$ and $\log p(y_t|q_t)$ are concave, then so is $\log p(Q|Y)$.
- Hessian H of $\log p(Q|Y)$ is block-tridiagonal: $p(y_t|q_t)$ contributes a block-diag term, and $\log p(q_{t+1}|q_t)$ contributes a block-tridiag term.

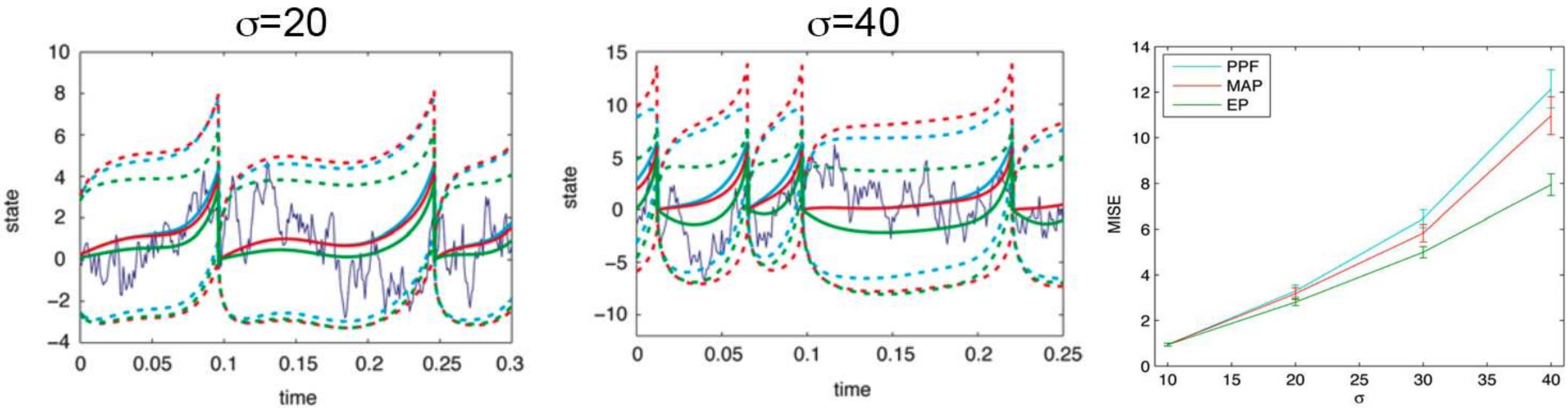
Now recall Newton's method: iteratively solve $HQ_{dir} = \nabla$. Solving tridiagonal systems requires $O(T)$ time.

— computing MAP by Newton's method requires $O(T)$ time, even in highly non-Gaussian cases.

(Newton here acts as an iteratively reweighted Kalman smoother (Davis and Rodriguez-Yam, 2005; Jungbacker and Koopman, 2007); all suff. stats may be obtained in $O(T)$ time. Similar results also well-known for expectation propagation (Ypma and Heskes, 2003; Yu and Sahani, 2007).)

Comparison on simulated soft-threshold leaky integrate-and-fire data

Model: $dV_t = -(V_t/\tau)dt + \sigma dB_t;\ \lambda(t) = f(V_t)$.



— extended Kalman-based methods are best in high-information (low-noise) limit, where Gaussian approximation is most accurate (Koyama et al., 2008).

Parameter estimation

Standard method: Expectation-Maximization (EM). Iterate between computing $E(Q|Y)$ (or $\hat{Q}$) and maximizing w.r.t. parameters θ.

Can be seen as coordinate ascent (slow) on first two terms of Laplace approximation:

$$\begin{aligned}
\log p(Y|\theta) &= \log \int p(Q|\theta)p(Y|\theta,Q)dQ \\
&\approx \log p(\hat{Q}_\theta|\theta) + \log p(Y|\hat{Q}_\theta,\theta) - \frac{1}{2}\log |H_{\hat{Q}_\theta}| \\
\hat{Q}_\theta &= \arg\max_Q \{\log p(Q|\theta) + \log p(Y|Q,\theta)\}
\end{aligned}$$

Better approach: simultaneous joint optimization. Main case of interest:

$$\begin{aligned}
\lambda_i(t) &= f\left[b + \vec{k}_i \cdot \vec{x}(t) + \sum_{i',j} h_{i',j} n_{i'}(t-j) + q_i(t)\right] \\
&= f\left[X_t\theta + q_i(t)\right] \\
\vec{q}_{t+dt} &= \vec{q}_t + A\vec{q}_t dt + \sigma\sqrt{dt}\vec{\epsilon}_t
\end{aligned}$$

More generally, assume q_t has an AR(p) prior and the observations y_t are members of a canonical exponential family with parameter $X_t\theta + q_t$.

We want to optimize

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

w.r.t. θ. If we drop the last term, we have a simple jointly concave optimization:

$$\begin{aligned}\hat{\theta} &= \arg\max_\theta \left\{\log p(\hat{Q}_\theta|\theta) + \log p(Y|\hat{Q}_\theta, \theta)\right\} \\ &= \arg\max_\theta \max_Q \left\{\log p(\hat{Q}|\theta) + \log p(Y|\hat{Q}, \theta)\right\}.\end{aligned}$$

Write the joint Hessian in (Q,θ) as $\begin{pmatrix} H_{\theta\theta} & H_{\theta Q}^T \\ H_{\theta Q} & H_{QQ} \end{pmatrix}$, with H_{QQ} block-tridiag.

Now use the Schur complement to efficiently compute the Newton step.

Computing $\nabla_\theta \log|H_{\hat{Q}_\theta}|$ also turns out to be easy ($O(T)$ time) here.

Constrained optimization

In many cases we need to impose (e.g., nonnegativity) constraints on q_t. Easy to incorporate here, via interior-point (barrier) methods:

$$\begin{aligned} \arg\max_{Q\in C} \log p(Q|Y) &= \lim_{\epsilon \searrow 0} \arg\max_{Q} \left\{ \log p(Q|Y) + \epsilon \sum_t f(q_t) \right\} \\ &= \lim_{\epsilon \searrow 0} \arg\max_{Q} \left\{ \sum_t \log p(q_{t+1}|q_t) + \log p(y_t|q_t) + \epsilon f(q_t) \right\}; \end{aligned}$$

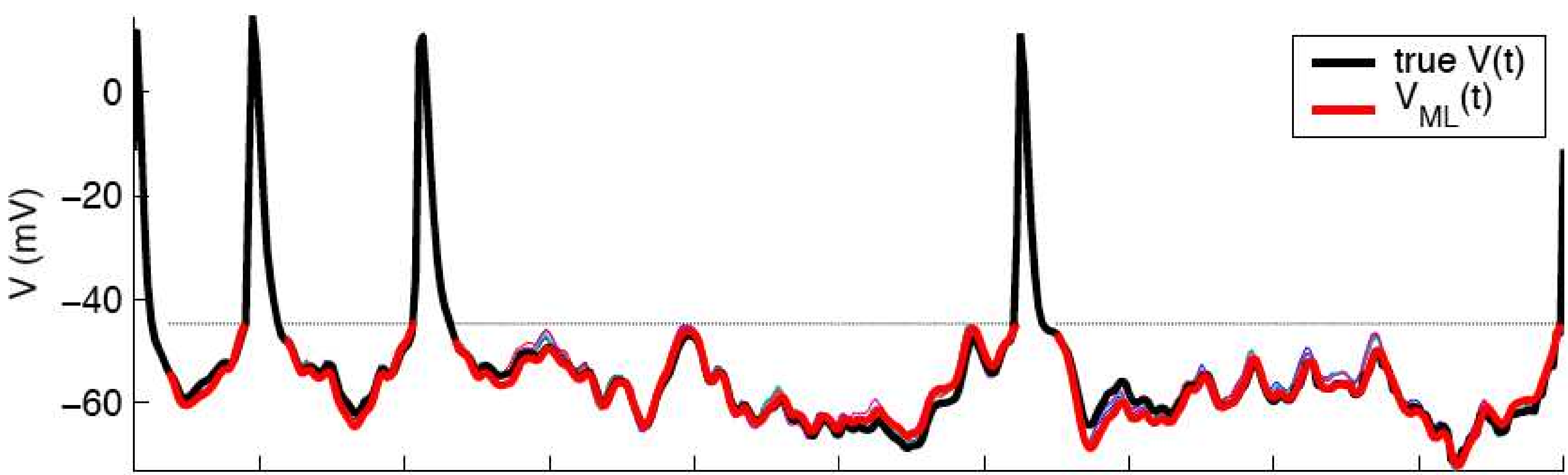
$f(.)$ is concave and approaching $-\infty$ near boundary of constraint set C. The Hessian remains block-tridiagonal and negative semidefinite for all $\epsilon > 0$, so optimization still requires just $O(T)$ time.

Example: computing the MAP subthreshold voltage given superthreshold spikes

Leaky, noisy integrate-and-fire model:

$$V(t+dt) = V(t) - dtV(t)/\tau + \sigma\sqrt{dt}\epsilon_t,\ \epsilon_t \sim \mathcal{N}(0,1)$$

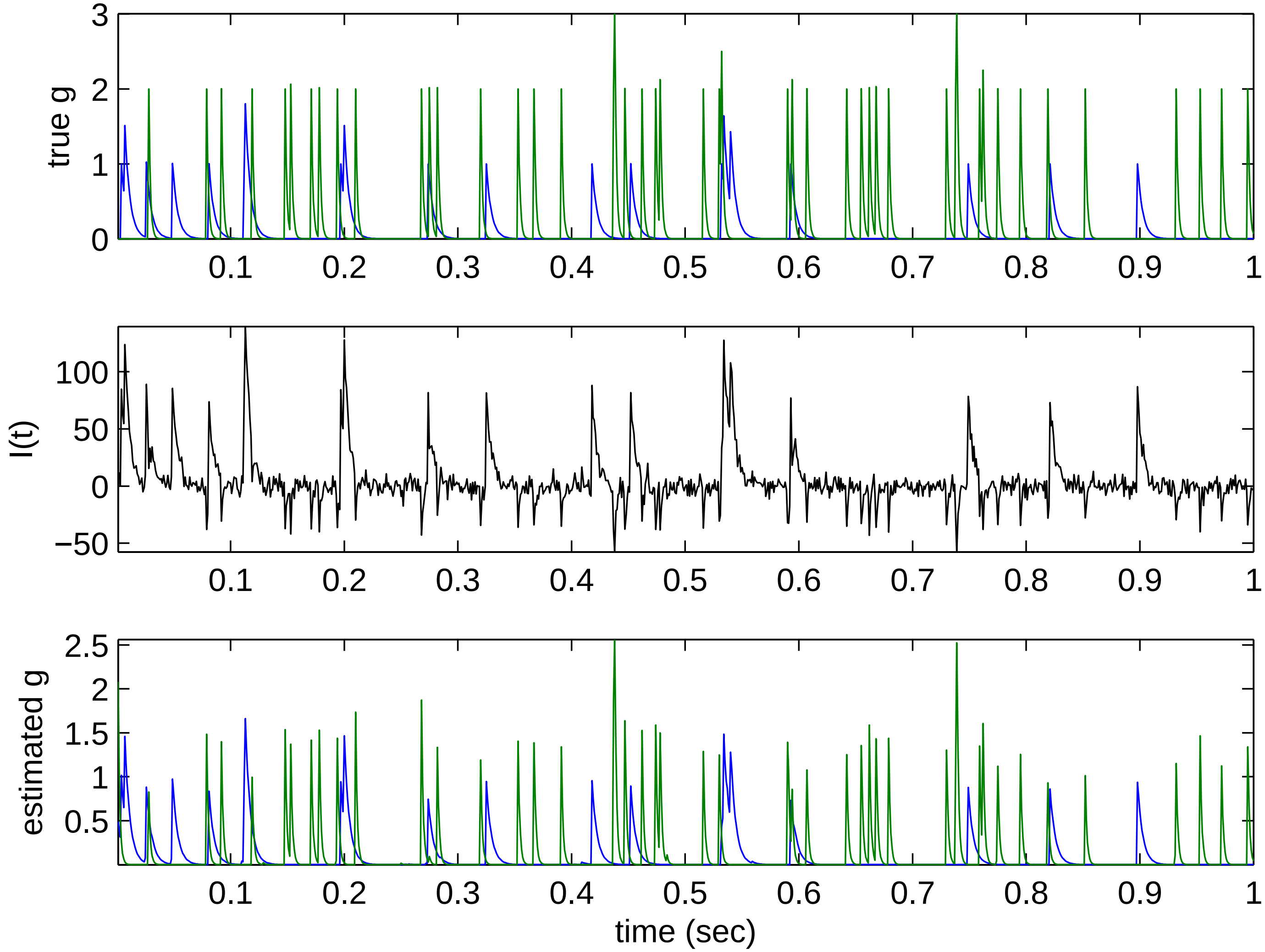
Observations: $y_t = 0$ (no spike) if $V_t < V_{th}$; $y_t = 1$ if $V_t = V_{th}$



(Paninski, 2006)

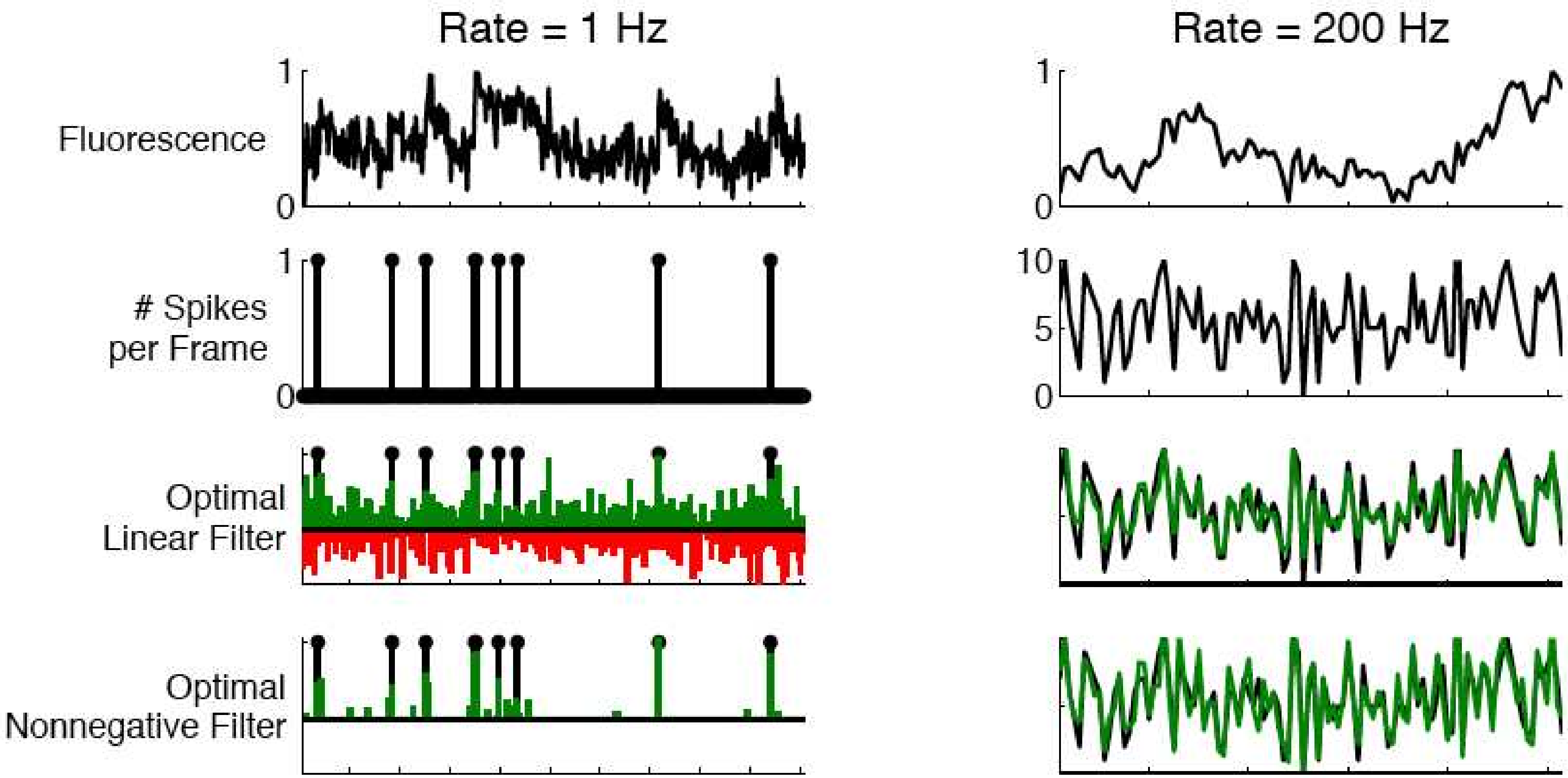
Example: inferring presynaptic input

$$
\begin{aligned}
I_t &= \sum_j g_j(t)(V_j - V_t) \\
g_j(t+dt) &= g_j(t) - dt g_j(t)/\tau_j + N_j(t), \ N_j(t) > 0
\end{aligned}
$$



Example: inferring spike times from slow, noisy calcium data

$$C(t+dt) = C(t) - dtC(t)/\tau + N_t;\;\; N_t > 0;\;\; y_t = C_t + \epsilon_t$$



— nonnegative deconvolution is a recurring problem in signal processing (e.g., spike sorting); many applications of these fast methods (Vogelstein et al., 2008).

Further generalizations: GLM spike train decoding

We've emphasized tridiagonal structure so far, but similar results hold for any problem with a banded Hessian.

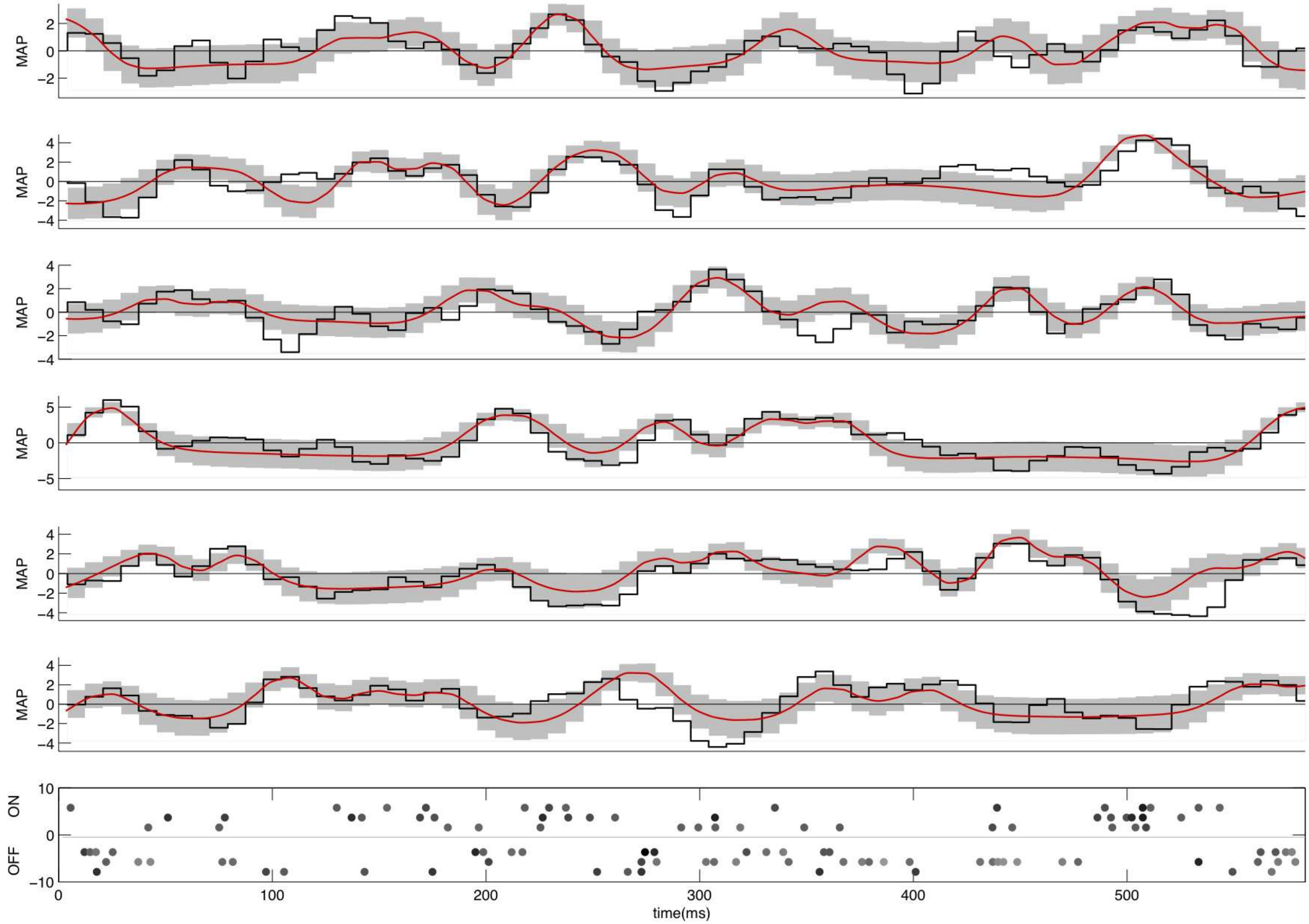
For example, look at point-process GLM again:

$$\lambda_i(t) = f\left[b + \vec{k}_i \cdot \vec{x}(t) + \sum_{i',j} h_{i',j} n_{i'}(t-j)\right]$$

If the spatiotemporal filter $\vec{k}_i$ has a finite impulse response, then Hessian (w.r.t. $\vec{x}(t)$) is banded and optimal decoding of stimulus $\vec{x}(t)$ requires $O(T)$ time.

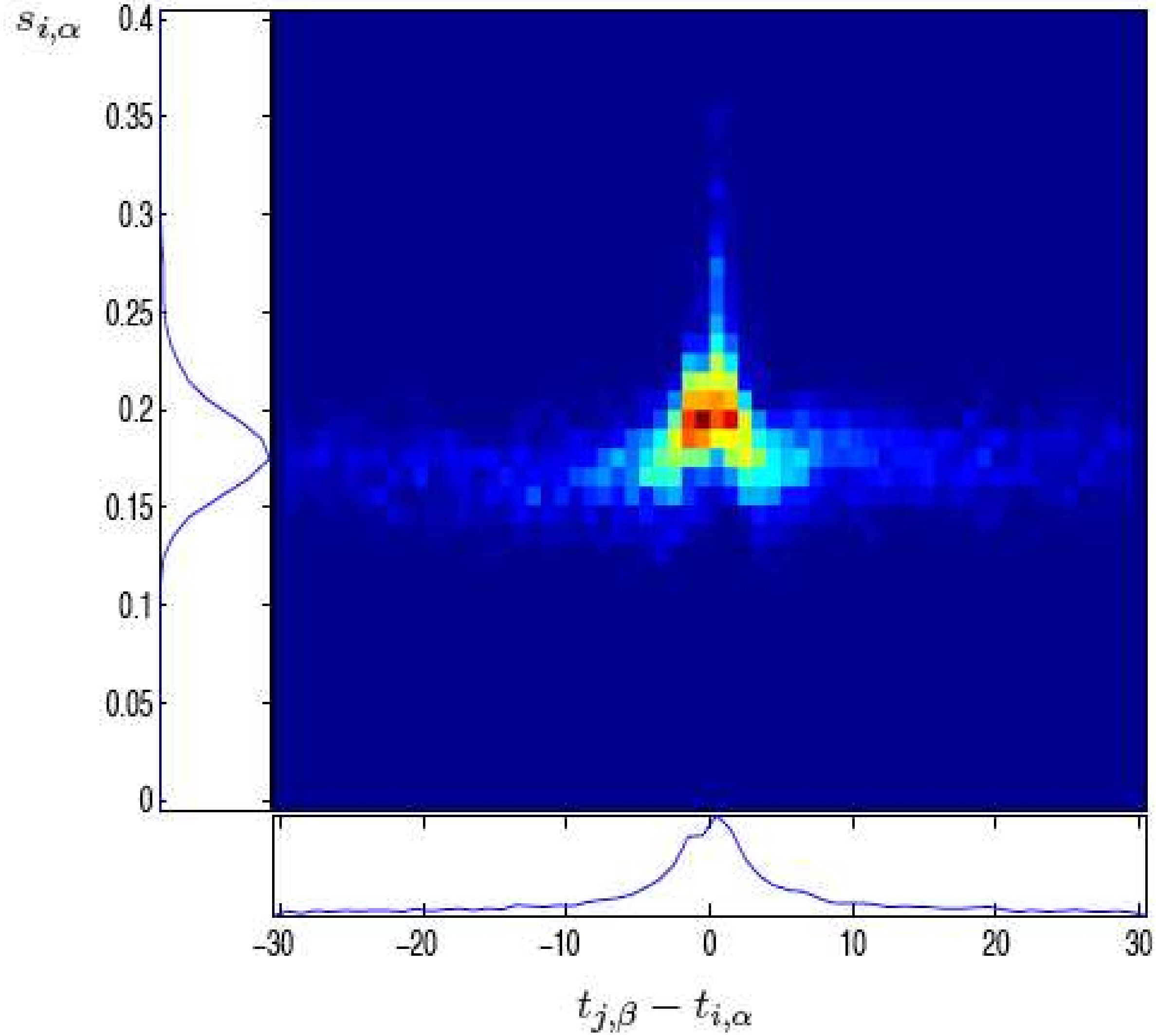
Similar speedups for MCMC methods (Ahmadian et al., 2008).

How important is timing?

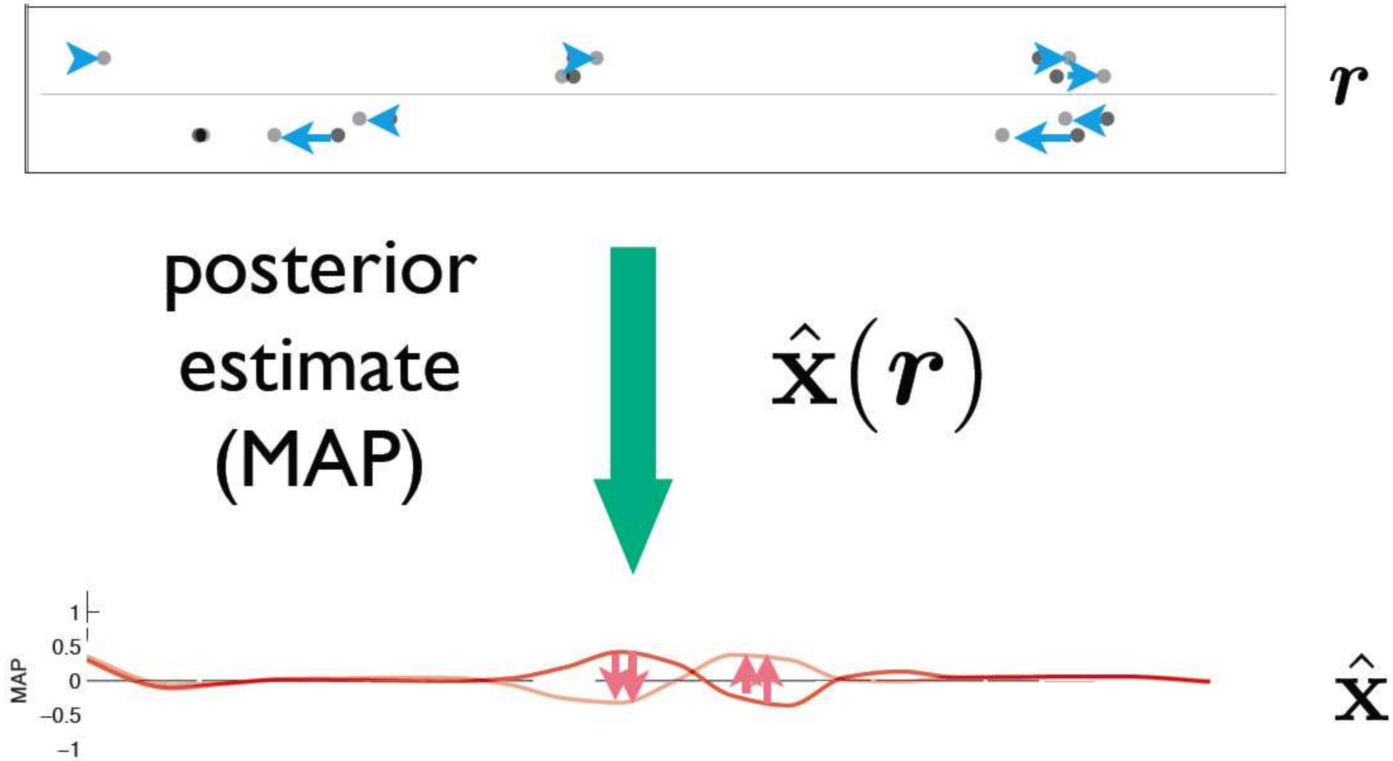


(Ahmadian et al., 2008)

Coincident spike are more "important"



Constructing a metric between spike trains



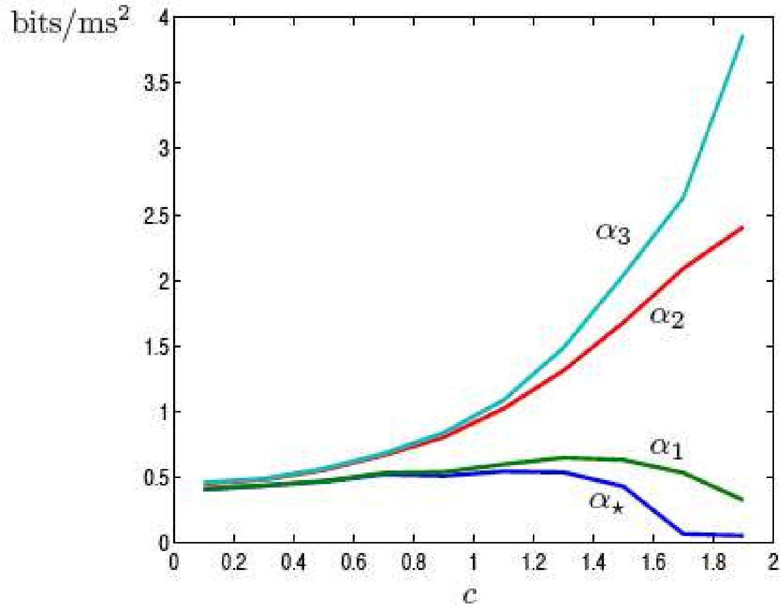
$$d(r_1, r_2) \equiv d_x(x_1, x_2)$$

Locally, $d(r, r + \delta r) = \delta r^T G_r \delta r$: interesting information in G_r.

Effects of jitter on spike trains

Look at degradations as we add Gaussian noise with covariance:

- α_*: $C \propto G^{-1}$ (optimal: minimizes error under constraint on $|C|$)
- α_1: $C \propto diag(G)^{-1}$ (perturb less important spikes more)
- α_2: $C \propto blkdiag(G)^{-1}$ (perturb spikes from different cells independently)
- α_3: $C \propto I$ (simplest)



— Non-correlated perturbations are more costly.

Can also add/remove spikes: cost of spike addition $\approx$ cost of jittering by 10 ms.

One last extension: two-d smoothing

Estimation of two-d firing rate surfaces comes up in a number of examples:

- place fields / grid cells
- post-fitting in spike-triggered covariance analysis
- tracking of non-stationary (time-varying) tuning curves
- "inhomogeneous Markov interval" models for spike-history dependence

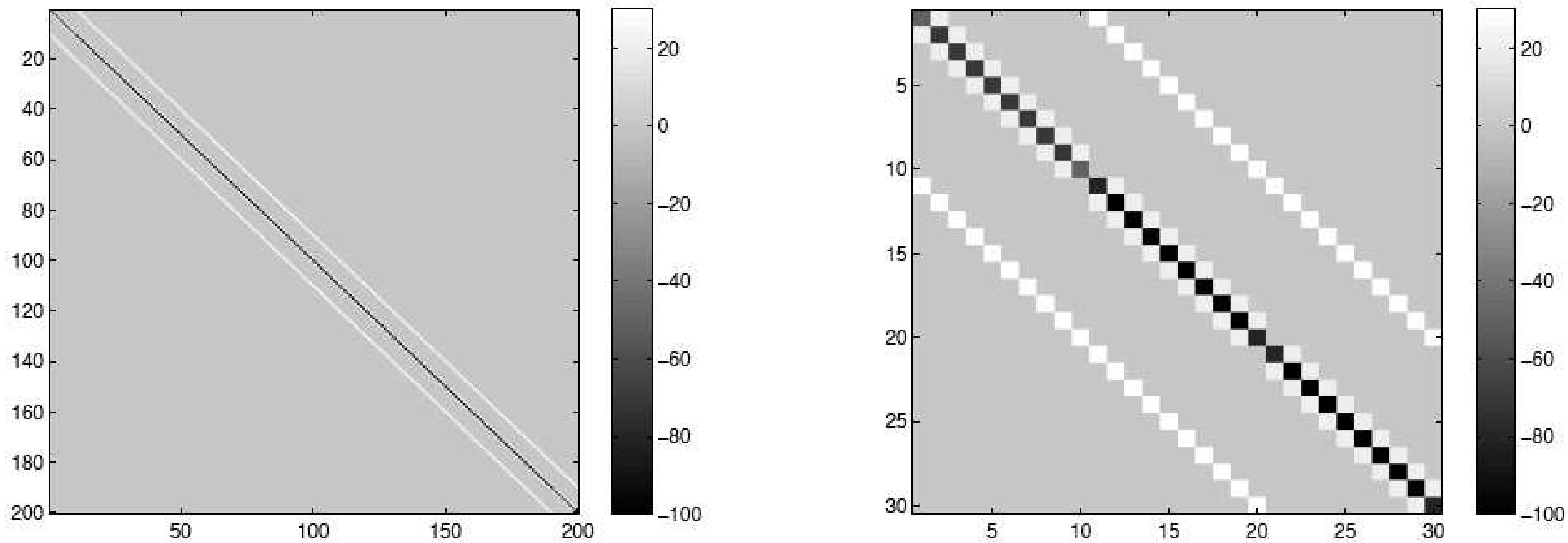
How to generalize fast 1-d state-space methods to 2-d case? Idea: use Gaussian process priors which are carefully selected to give banded Hessians.

Model: hidden variable Q is a random surface with a Gaussian prior:
$Q \sim \mathcal{N}(\mu, C)$;

Spikes are generated by a point process whose rate is a function of Q:
$\lambda(\vec{x}) = f[Q(\vec{x})]$ (easy to incorporate additional effects here, e.g. spike history)

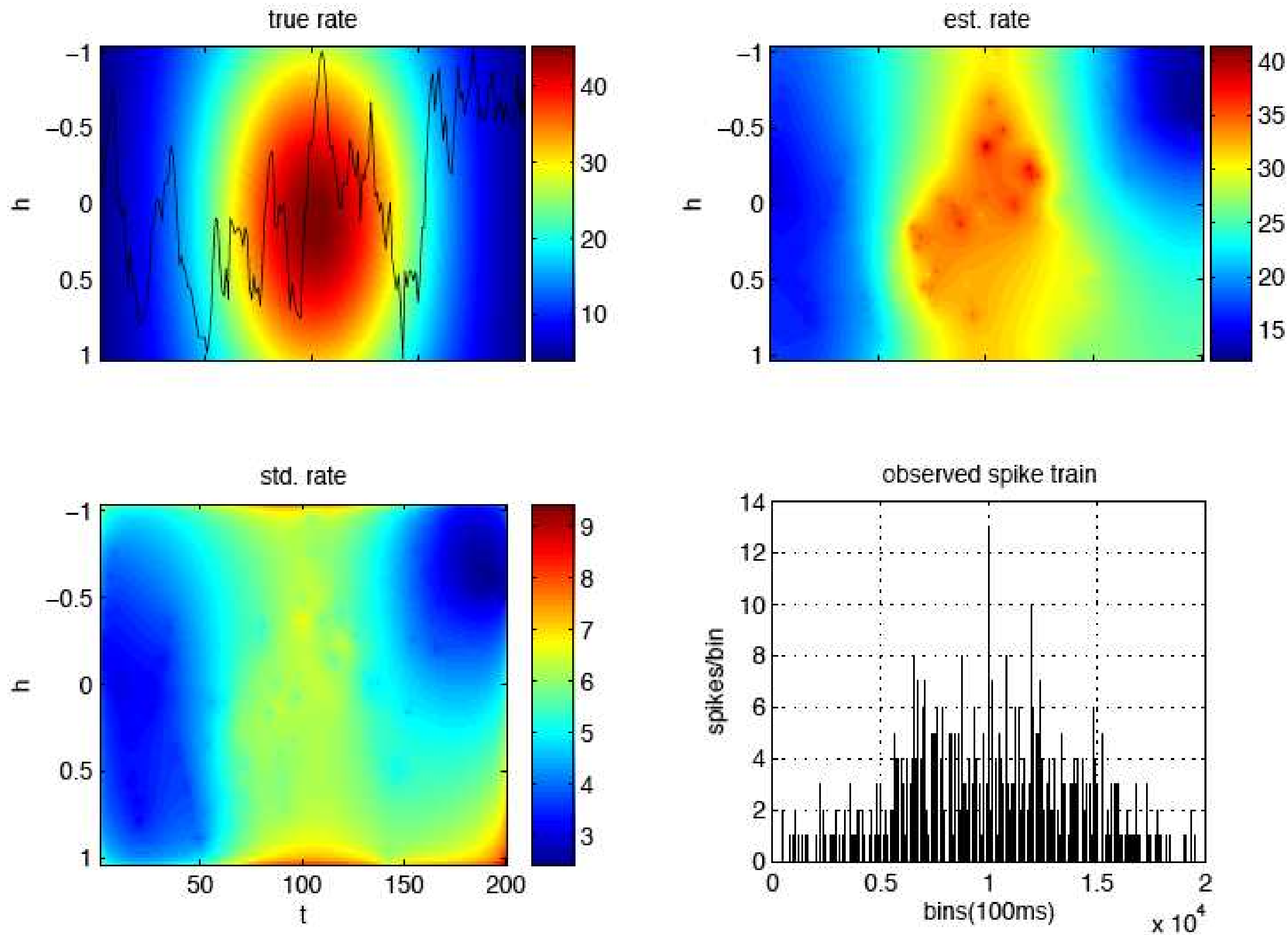
Now the Hessian of the log-posterior of Q is $C^{-1} + D$, where D is diagonal (Cunningham et al., 2007). For Newton, we need to solve $(C^{-1} + D)Q_{dir} = \nabla$.

Example: nearest-neighbor smoothing prior

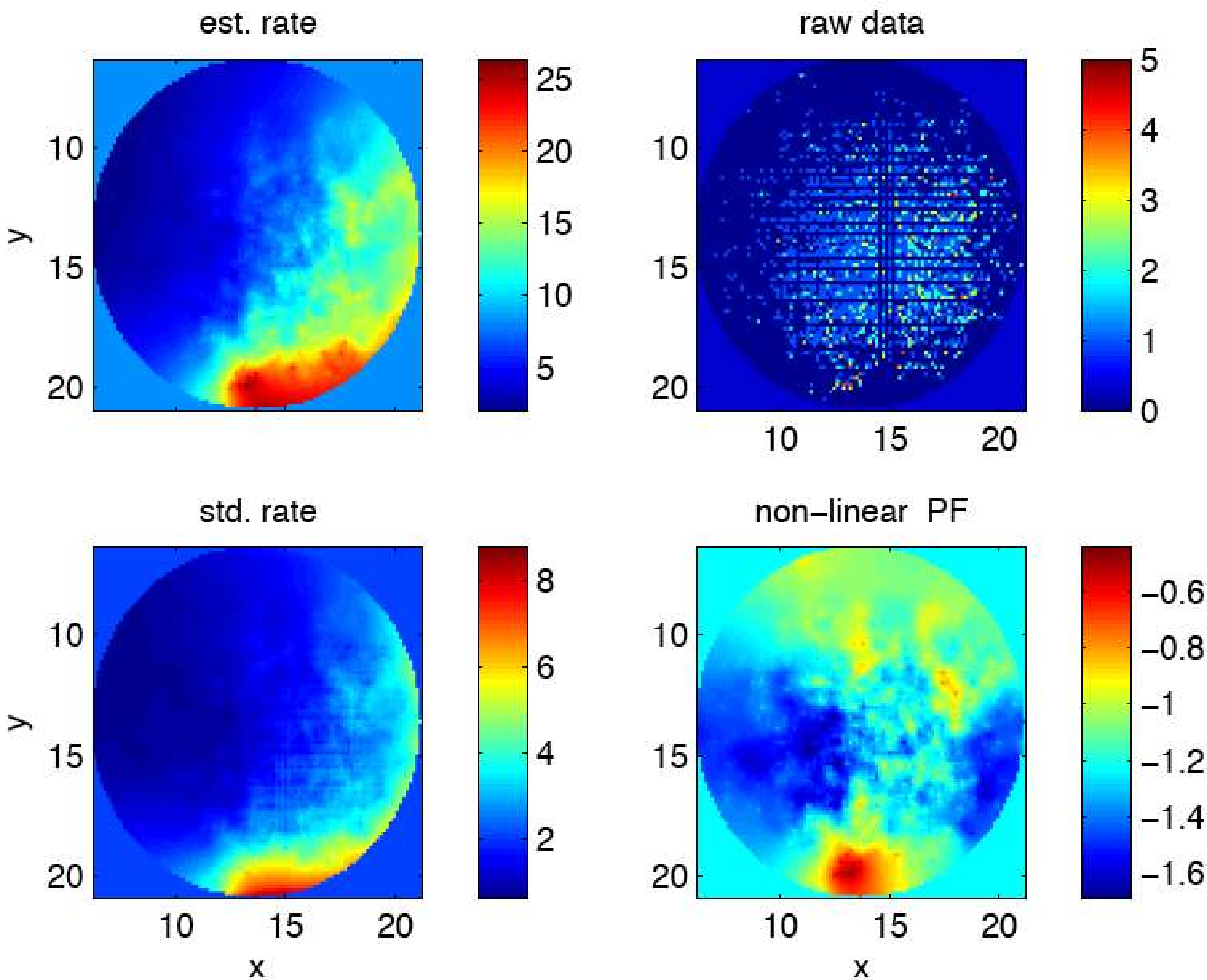


For prior covariance C such that C^{-1} contains only neighbor potentials, we can solve $(C^{-1} + D)Q_{dir} = \nabla$ in $O(\dim(Q)^{1.5})$ time using direct methods ("approximate minimum degree" algorithm — built-in to Matlab sparse $A\backslash b$ code) and potentially in $O(\dim(Q))$ time using multigrid (iterative) methods (Rahnama Rad and Paninski, 2008).

Estimating a time-varying tuning curve given a limited sample path



Estimating a two-d place field



Collaborators

Theory and numerical methods

- Y. Ahmadian, S. Escola, G. Fudenberg, Q. Huys, J. Kulkarni, M. Nikitchenko, X. Pitkow, K. Rahnama, G. Szirtes, T. Toyoizumi, Columbia
- E. Doi, E. Simoncelli, NYU
- E. Lalor, NKI
- A. Haith, C. Williams, Edinburgh
- M. Ahrens, J. Pillow, M. Sahani, Gatsby
- S. Koyama, R. Kass, CMU
- J. Lewi, Georgia Tech
- J. Vogelstein, Johns Hopkins
- W. Wu, FSU

Retinal physiology

- E.J. Chichilnisky, J. Shlens, V. Uzzell, Salk

References

Ahmadian, Y., Pillow, J., and Paninski, L. (2008). Efficient Markov Chain Monte Carlo methods for decoding population spike trains. *Under review, Neural Computation*.

Cunningham, J., Yu, B., Shenoy, K., and Sahani, M. (2007). Inferring neural firing rates from spike trains using Gaussian processes. *NIPS*.

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

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

Koyama, S., Kass, R., and Paninski, L. (2008). Efficient computation of the most likely path in integrate-andfire and more general state-space models. *COSYNE*.

Paninski, L. (2006). The most likely voltage path and large deviations approximations for integrate-and-fire neurons. *Journal of Computational Neuroscience*, 21:71–87.

Paninski, L. (2007). Inferring synaptic inputs given a noisy voltage trace via sequential Monte Carlo methods. *Journal of Computational Neuroscience*, Under review.

Pillow, J., Shlens, J., Paninski, L., , Sher, A., Litke, A., Chichilnisky, E., and Simoncelli, E. (2008). Spatiotemporal correlations and visual signaling in a complete neuronal population. *Nature*, In press.

Pitkow, X., Sompolinsky, H., and Meister, M. (2007). A neural computation for visual acuity in the presence of eye movements. *PLOS Biology*, 5:e331.

Rahnama Rad, K. and Paninski, L. (2008). Efficient estimation of two-dimensional firing rate surfaces via Gaussian process methods. *COSYNE*.

Rucci, M., Iovin, R., Poletti, M., and Santini, F. (2007). Miniature eye movements enhance fine spatial detail. *Nature*, 447:851–854.

Vogelstein, J., Babadi, B., Watson, B., Yuste, R., and Paninski, L. (2008). Fast nonnegative deconvolution via tridiagonal interior-point methods, applied to calcium fluorescence data. *Advances in Neural Information Processing*, Under review.