State-space models

Some notation:

- $x(t)$ = kinematic state
- $r(t)$ = neural response
- encoding model: $p[r(t)|x(t)]$
- kinematics model: $p[x(t)|x(t-1)]$
- optimal Bayesian decoder: $p[x(t)|r(1:t)]$

This approach is computationally efficient (because the decoder is computed recursively at each time step t), provides measures of posterior uncertainty, and is currently the most flexible / powerful framework available.

Progress in several directions

- Fast estimation of nonlinear, nonstationary encoding models (Pnevmatikakis et al JCGS '13, Pnevmatikakis+P, AISTATS '12)
- Robust subspace identification for dimensionality reduction and estimation of latent dynamics (Pnevmatikakis, Pfau, P, COSYNE '13)
- Fast robust methods for nonparametric estimation of nonlinear encoding models (Rahnama+P, COSYNE '13)
- "Expected loglikelihood" methods for fast encoding model estimation (Ramirez+P, under review)
- Hierarchical estimation of encoding models: sharing information across all available neurons (Merel, Pnevmatikakis et al, COSYNE '13)

- Exact inference in "low-rank" state-space models (Smith et al, AISTATS '12)
- Robust sequential Markov chain Monte Carlo "particle filter" methods (P et al, CISS '12)
- Fast, robust MCMC methods for sampling from constrained distributions (Pakman +P, under review)
- Quantifying information loss due to spike sorting errors and loss of spiking temporal resolution (Smith+P, under review)

- See http://www.stat.columbia.edu/~liam/research/pubs for (p)reprints

Fast high-dimensional adaptive encoding model estimation methods

Goal: update encoding model parameters adaptively.

Standard fully-Bayesian adaptive estimation methods scale like $O(d^3)$ per timestep (or worse).

New method: $O(d)$. Allows for much richer nonstationary models than previously possible. Main idea: low-rank approximation of posterior state covariance.

Can handle non-smooth priors, likelihoods.

Introduced in P (2011); generalized in Pnevmatikakis+P (2012);

Pnevmatikakis et al (2012) proved convergence, rigorous error bounds.

Exploiting expected loglikelihoods

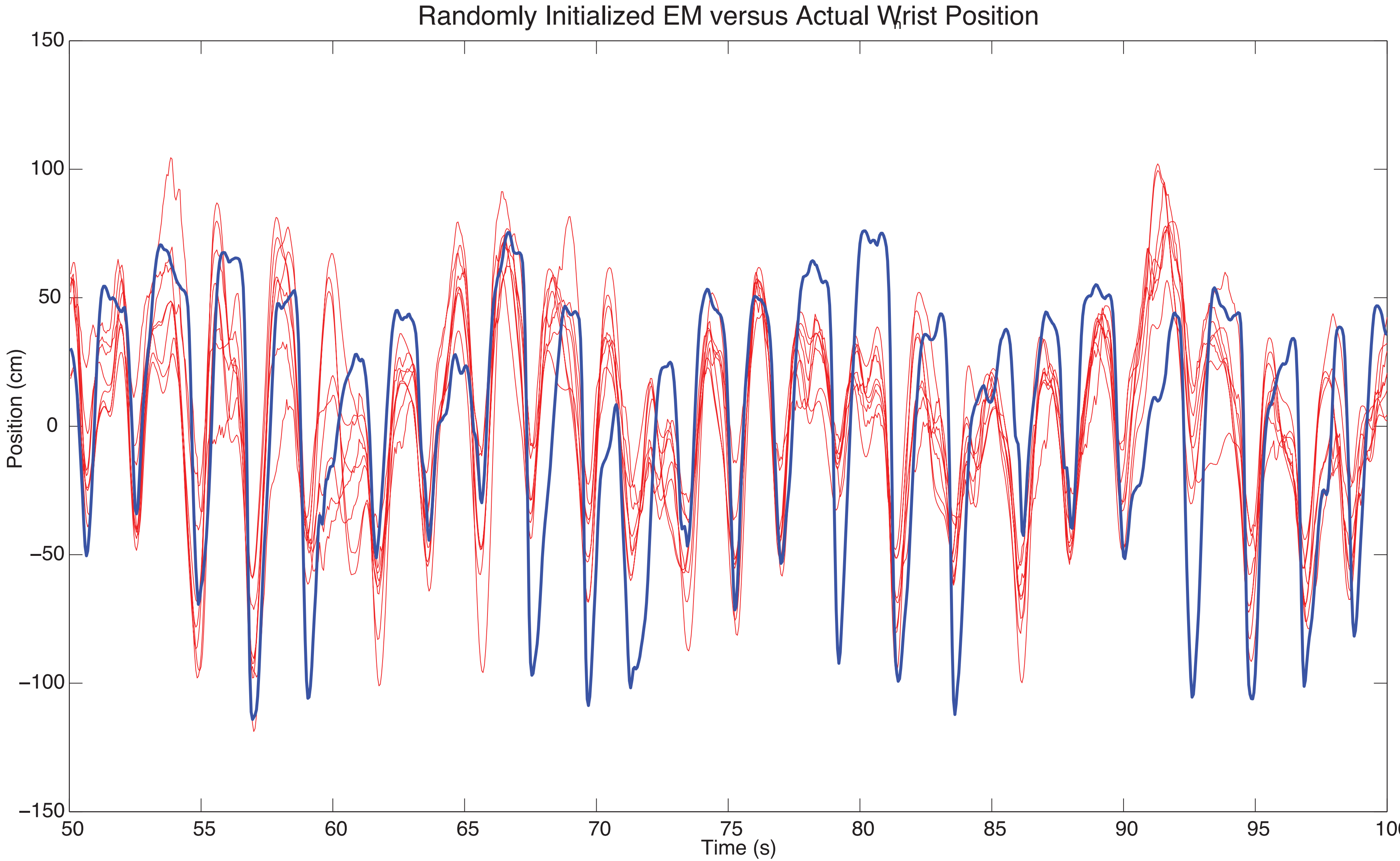
$$\begin{aligned} L(\theta) &= \sum_{n=1}^{N} \Big((x_n^T\theta) r_n - G(x_n^T\theta) \Big) + \text{const}(\theta) \\ &\approx \Big(\sum_{n=1}^{N} x_n^T r_n \Big) \theta - N\mathbf{E}\Big[G(x^T\theta) \Big] \equiv \tilde{L}(\theta) \end{aligned}$$

r: responses; x: kinematic variables; θ: parameter to be estimated

Expected LL (ELL) can be computed and optimized orders of magnitude faster than LL.
ELL estimates are often more accurate than MLE. Full details: Ramirez and Paninski (2012).

Convex methods for state-space identification

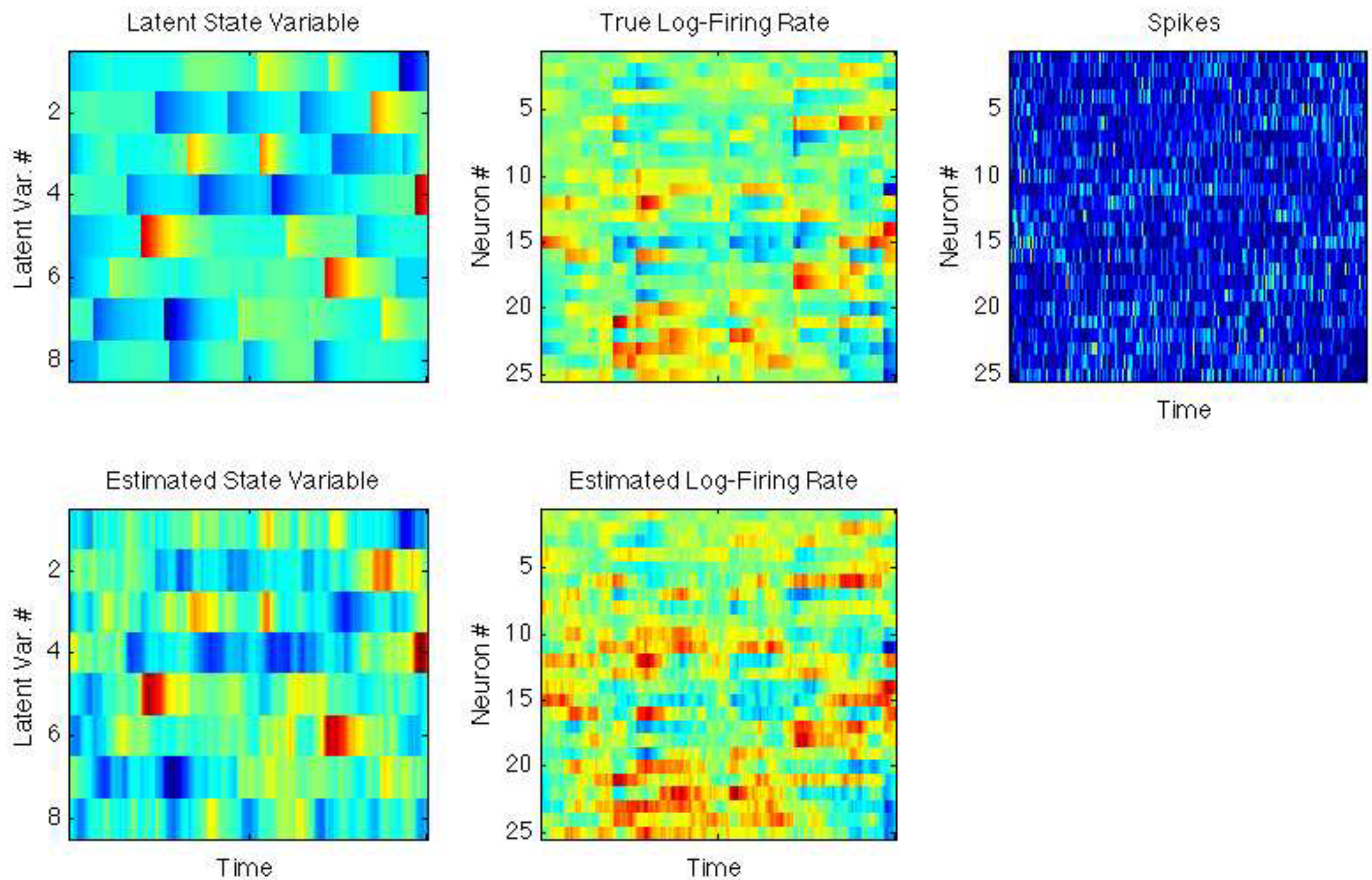
- Basic idea: observed responses depend on latent (unobserved) variables in addition to (observed) kinematics
- Modeling these latent effects explicitly improves decoding performance (as shown by various groups)
- Usual approach for estimating latent factors: expectation maximization (EM)
- But EM is slow and non-robust: non-convex



Randomly Initialized EM versus Actual Wrist Position
150
100
50
0
−50
−100
−150
Position (cm)
50
55
60
65
70
75
80
85
90
95
100
Time (s)

Dimensionality reduction; inferring hidden dynamics

Dynamic generalized factor analysis model: q_t evolves according to a simple linear dynamical system, with "kicks." Log-firing rates modeled as linear functions of q_t. Convex rank-penalized optimization methods to infer q_t given spike train.



Open challenge: richer nonlinear models. E. Pnevmatikakis and D. Pfau, ongoing

Exact inference methods in nonstandard state spaces

How do we perform exact inference for time series on manifolds, or more general state spaces (e.g., space of all reachable joint configurations)? Standard methods assume vector state spaces; exact methods previously available only in Gaussian (Kalman) setting.

Main result: broad class of priors that enable exact inference:

$$p(X) \propto \prod_{t=1}^{T-1} \sum_{z_t=1}^{R_t} f_{t,z_t}(x_t) g_{t,z_t}(x_{t+1})$$

See Smith et al (2012) for full details.

Sequential MCMC methods for robust particle filtering

Recall basic recursion:

$$p(q_t, Y_{1:t}) = p(y_t|q_t) \int_{q_{t-1}} p(q_t|q_{t-1}) p(q_{t-1}, Y_{1:t-1}) dq_{t-1}.$$

Particle filter: importance sampling to approximate integral.

Can be highly effective (Doucet et al., 2001).

However, importance sampling is known to be very non-robust in many important cases: if we put the particles in the wrong part of the space, the variance of the importance weights becomes too large and the filter fails.

Robust particle filtering via sequential MCMC

Rewrite basic recursion:

$$p(q_t, Y_{1:t}) = \int_{q_{t-1}} p(y_t|q_t)p(q_t|q_{t-1})p(q_{t-1}, Y_{1:t-1})dq_{t-1}. \quad (1)$$

$$= \int_{q_{t-1}} p(q_t, q_{t-1}, Y_{1:t})dq_{t-1}. \quad (2)$$

Basic idea: use MCMC to sample directly from $p(q_t, q_{t-1}|Y_{1:t})$ — bypass nonrobust particle filter step entirely.

See Paninski et al (2012) for full details.