Efficient adaptive experimental design

Liam Paninski

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

April 1, 2010

— with J. Lewi, S. Woolley

Avoiding the curse of insufficient data

1: Estimate some functional $f(p)$ instead of full joint distribution $p(r, s)$

— information-theoretic functionals

2: Improved nonparametric estimators

— minimax theory for discrete distributions under KL loss

3: Select stimuli more efficiently

— optimal experimental design

(*4: Parametric approaches*)

Setup

Assume:

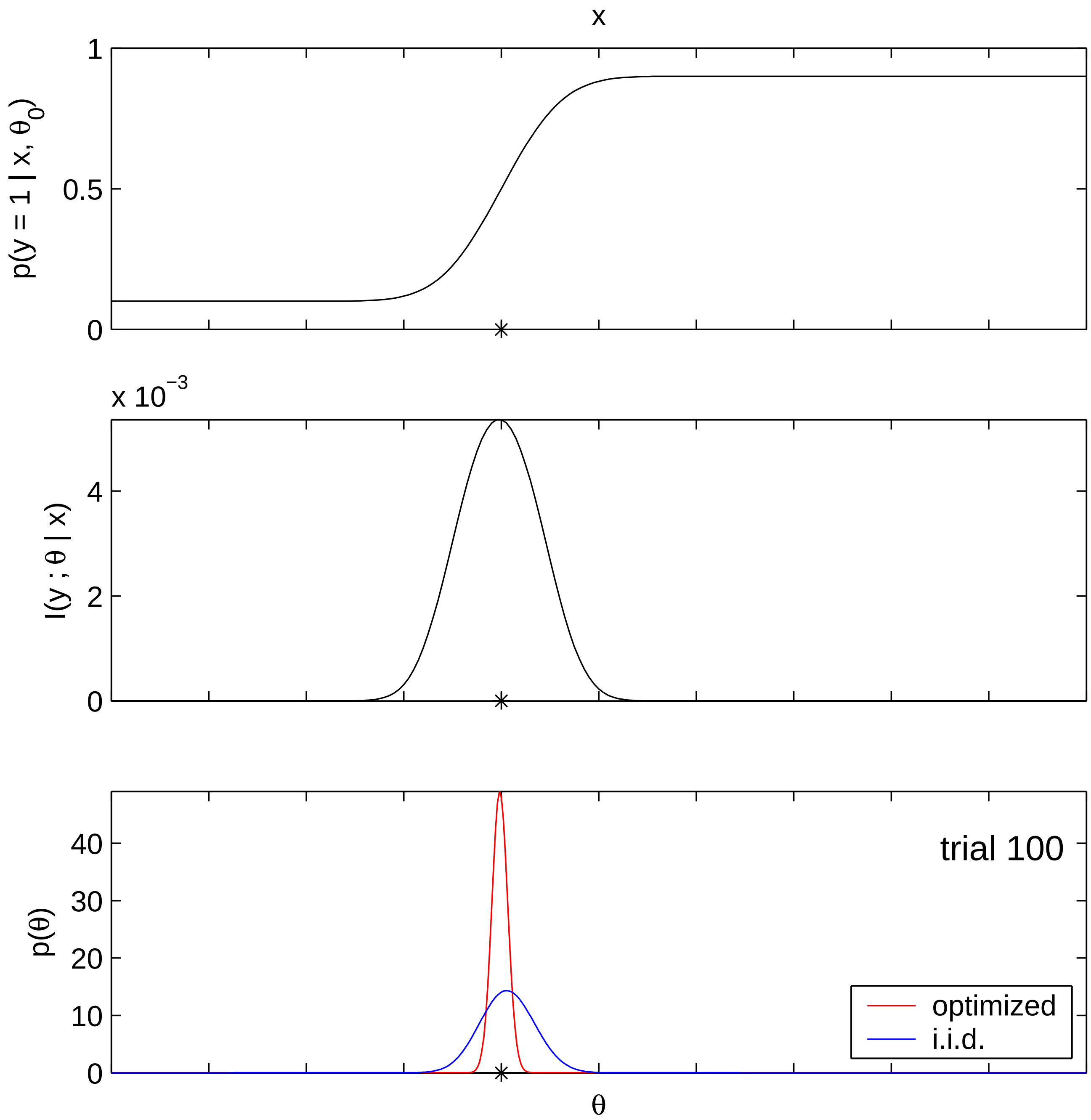
- parametric model $p_\theta(r|\vec{x})$ on responses r given inputs $\vec{x}$
- prior distribution $p(\theta)$ on finite-dimensional model space

Goal: estimate θ from experimental data

Usual approach: draw stimuli i.i.d. from fixed $p(\vec{x})$

Adaptive approach: choose $p(\vec{x})$ on each trial to maximize $E_{\vec{x}} I(\theta; r|\vec{x})$ (e.g. "staircase" methods).

Snapshot: one-dimensional simulation



Asymptotic result

Under regularity conditions, a posterior CLT holds (Paninski, 2005):

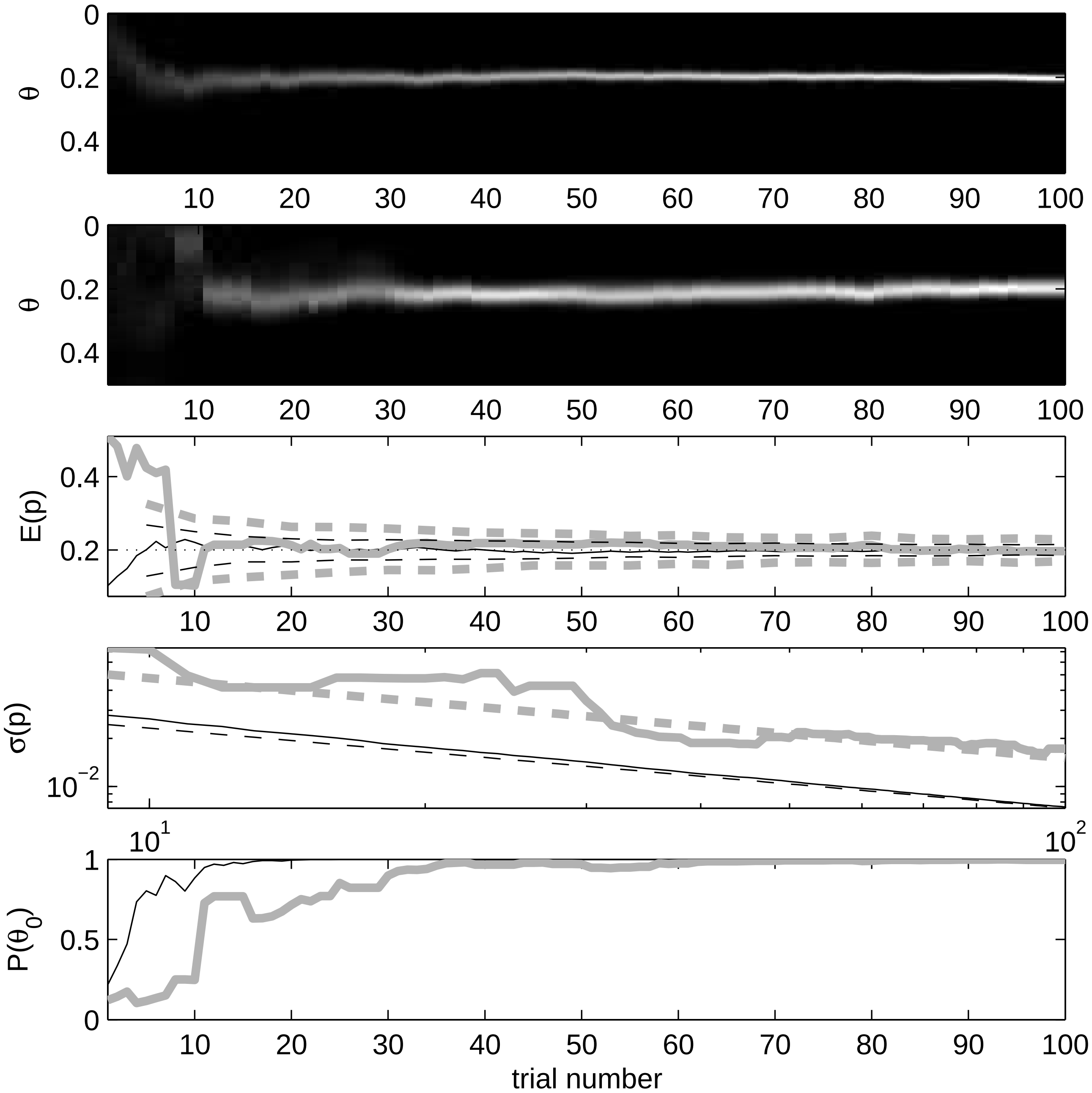
$$p_N\left(\sqrt{N}(\theta - \theta_0)\right) \to \mathcal{N}(\mu_N, \sigma^2); \quad \mu_N \sim \mathcal{N}(0, \sigma^2)$$

- $(\sigma^2_{iid})^{-1} = E_x(I_x(\theta_0))$
- $(\sigma^2_{info})^{-1} = \text{argmax}_{C \in co(I_x(\theta_0))} \log |C|$

$\implies \sigma^2_{iid} > \sigma^2_{info}$ unless $I_x(\theta_0)$ is constant in x

$co(I_x(\theta_0))$ = convex closure (over x) of Fisher information matrices $I_x(\theta_0)$. ($\log |C|$ strictly concave: maximum unique.)

Illustration of theorem

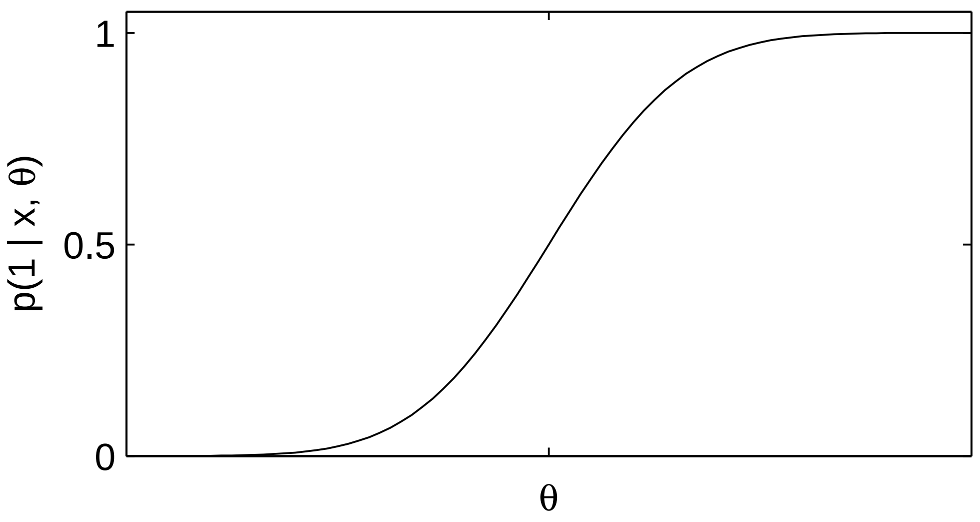


Psychometric example

- stimuli x one-dimensional: intensity
- responses r binary: detect/no detect

$$p(r = 1|x, \theta) = f((x - \theta)/a)$$

- scale parameter a (assumed known)
- want to learn threshold parameter θ as quickly as possible



Psychometric example: results

- variance-minimizing and info-theoretic methods asymptotically same

- just one unique function f^* for which $\sigma_{iid} = \sigma_{opt}$; for any other f, $\sigma_{iid} > \sigma_{opt}$

$$I_x(\theta) = \frac{(\dot{f}_{a,\theta})^2}{f_{a,\theta}(1 - f_{a,\theta})}$$

- f^* solves

$$\dot{f}_{a,\theta} = c\sqrt{f_{a,\theta}(1 - f_{a,\theta})}$$

$$f^*(t) = \frac{\sin(ct) + 1}{2}$$

- $\sigma_{iid}^2 / \sigma_{opt}^2 \sim 1/a$ for a small

Part 2: Computing the optimal stimulus

OK, now how do we actually do this in neural case?

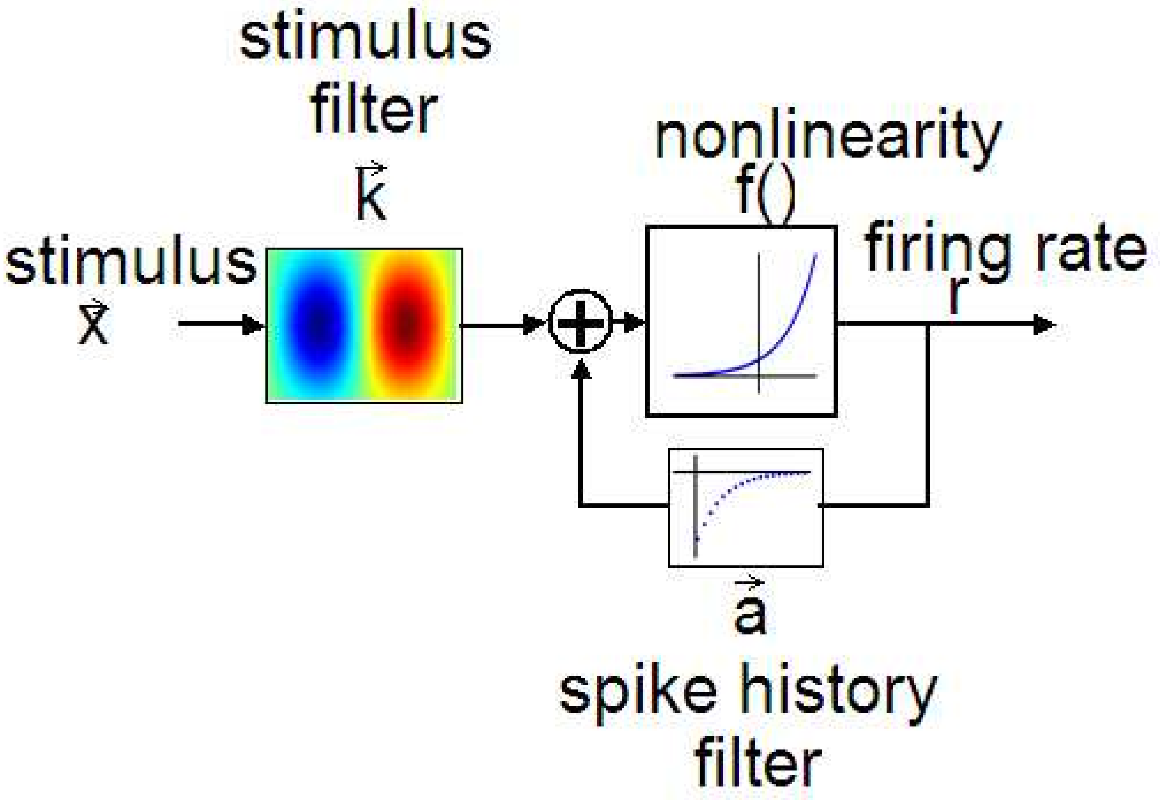
- Computing $I(\theta; r|\vec{x})$ requires an integration over θ
  — in general, exponentially hard in $\dim(\theta)$
- Maximizing $I(\theta; r|\vec{x})$ in $\vec{x}$ is doubly hard
  — in general, exponentially hard in $\dim(\vec{x})$

Doing all this in real time ($\sim$ 10 ms - 1 sec) is a major challenge!

Three key steps

1. Choose a tractable, flexible model of neural encoding
2. Choose a tractable, accurate approximation of the posterior $p(\vec{\theta}|\{\vec{x}_i, r_i\}_{i \leq N})$
3. Use approximations and some perturbation theory to reduce optimization problem to a simple 1-d linesearch

Step 1: focus on GLM case



$$r_i \sim Poiss(\lambda_i); \qquad \lambda_i | \vec{x}_i, \vec{\theta} = f(\vec{k} \cdot \vec{x}_i + \sum_j a_j r_{i-j}).$$

More generally, $\log p(r_i|\theta, \vec{x}_i) = k(r) f(\theta \cdot \vec{x}_i) + s(r) + g(\theta \cdot \vec{x}_i)$

Goal: learn $\vec{\theta} = \{\vec{k}, \vec{a}\}$ in as few trials as possible.

GLM likelihood

$$\lambda_i \sim Poiss(\lambda_i)$$

$$\lambda_i | \vec{x}_i, \vec{\theta} = f(\vec{k} \cdot \vec{x}_i + \sum_j a_j r_{i-j})$$

$$\log p(r_i | \vec{x}_i, \vec{\theta}) = -f(\vec{k} \cdot \vec{x}_i + \sum_j a_j r_{i-j}) + r_i \log f(\vec{k} \cdot \vec{x}_i + \sum_j a_j r_{i-j})$$

Two key points:

- Likelihood is "rank-1" — only depends on $\vec{\theta}$ along $\vec{z} = (\vec{x}, \vec{r})$.
- f convex and log-concave $\implies$ log-likelihood concave in $\vec{\theta}$

Step 2: representing the posterior

Idea: Laplace approximation

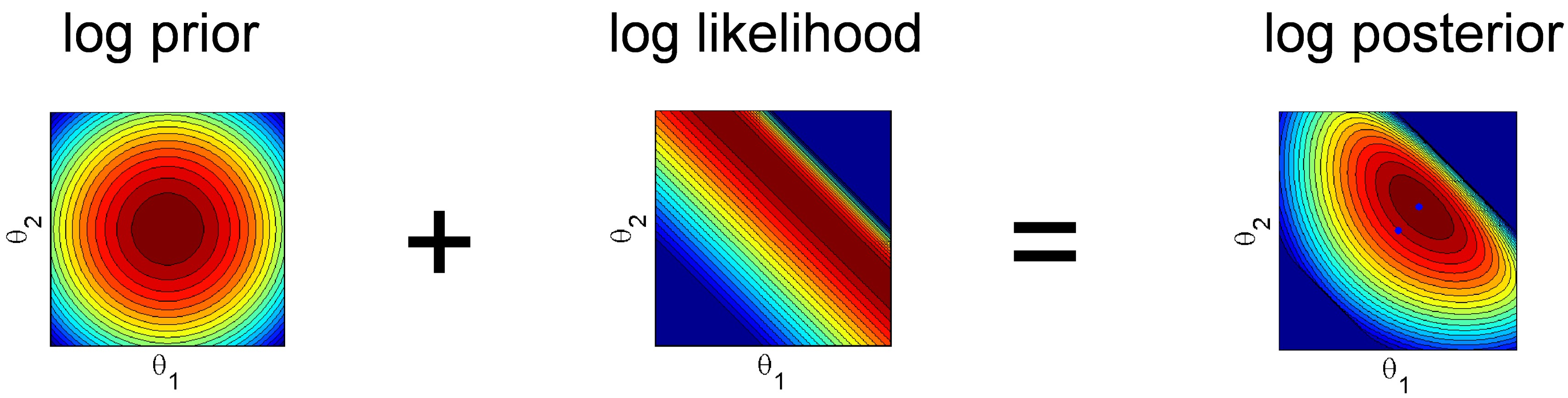
$$p(\vec{\theta}|\{\vec{x}_i, r_i\}_{i\leq N}) \approx \mathcal{N}(\mu_N, C_N)$$

Justification:

- posterior CLT
- likelihood is log-concave, so posterior is also log-concave:

$$\log p(\vec{\theta}|\{\vec{x}_i, r_i\}_{i\leq N}) \sim \log p(\vec{\theta}|\{\vec{x}_i, r_i\}_{i\leq N-1}) + \log p(r_N|x_N, \vec{\theta})$$

Efficient updating



Updating μ_N: one-d search

Updating C_N: rank-one update, $C_N = (C_{N-1}^{-1} + b\vec{z}^t\vec{z})^{-1}$ — use Woodbury lemma

Total time for update of posterior: $O(d^2)$

Step 3: Efficient stimulus optimization

Laplace approximation $\implies I(\theta; r|\vec{x}) \sim E_{r|\vec{x}} \log \frac{|C_{N-1}|}{|C_N|}$

— this is nonlinear and difficult, but we can simplify using perturbation theory: $\log|I + A| \approx \text{trace}(A)$.

Now we can take averages over $p(r|\vec{x}) = \int p(r|\theta, \vec{x}) p_N(\theta) d\theta$: standard Fisher info calculation given Poisson assumption on r.

Further assuming $f(.) = \exp(.)$ allows us to compute expectation exactly, using m.g.f. of Gaussian.

...finally, we want to maximize $F(\vec{x}) = g(\mu_N \cdot \vec{x}) h(\vec{x}^t C_N \vec{x})$.

Computing the optimal $\vec{x}$

$\max_{\vec{x}} g(\mu_N \cdot \vec{x}) h(\vec{x}^t C_N \vec{x})$ increases with $||\vec{x}||_2$: constraining $||\vec{x}||_2$ reduces problem to nonlinear eigenvalue problem.

Lagrange multiplier approach (Berkes and Wiskott, 2006) reduces problem to 1-d linesearch, once eigendecomposition is computed — much easier than full d-dimensional optimization!

Rank-one update of eigendecomposition may be performed in $O(d^2)$ time (Gu and Eisenstat, 1994).

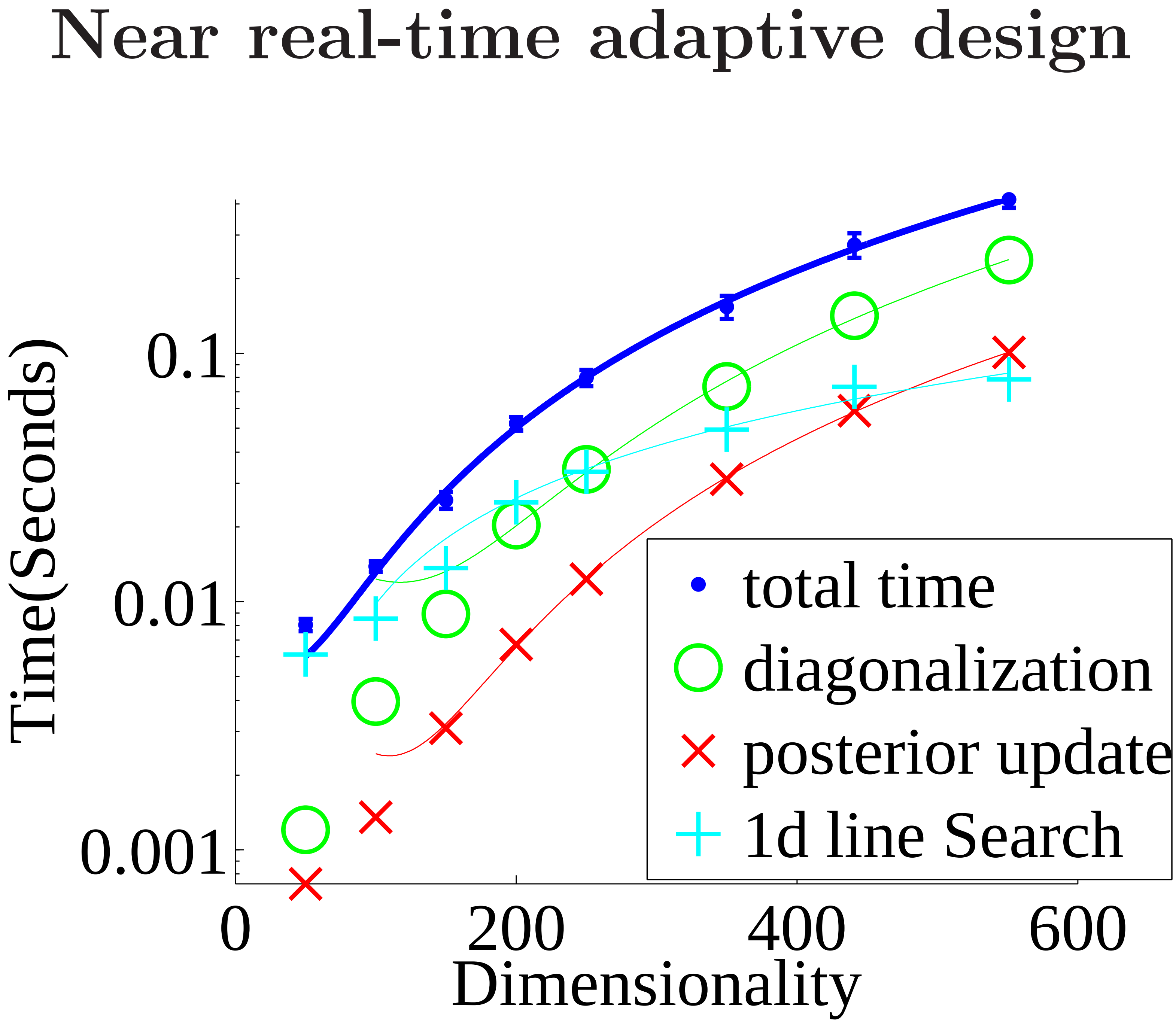
$\Longrightarrow$ Computing optimal stimulus takes $O(d^2)$ time.

Side note: linear-Gaussian case is easy

Linear Gaussian case:

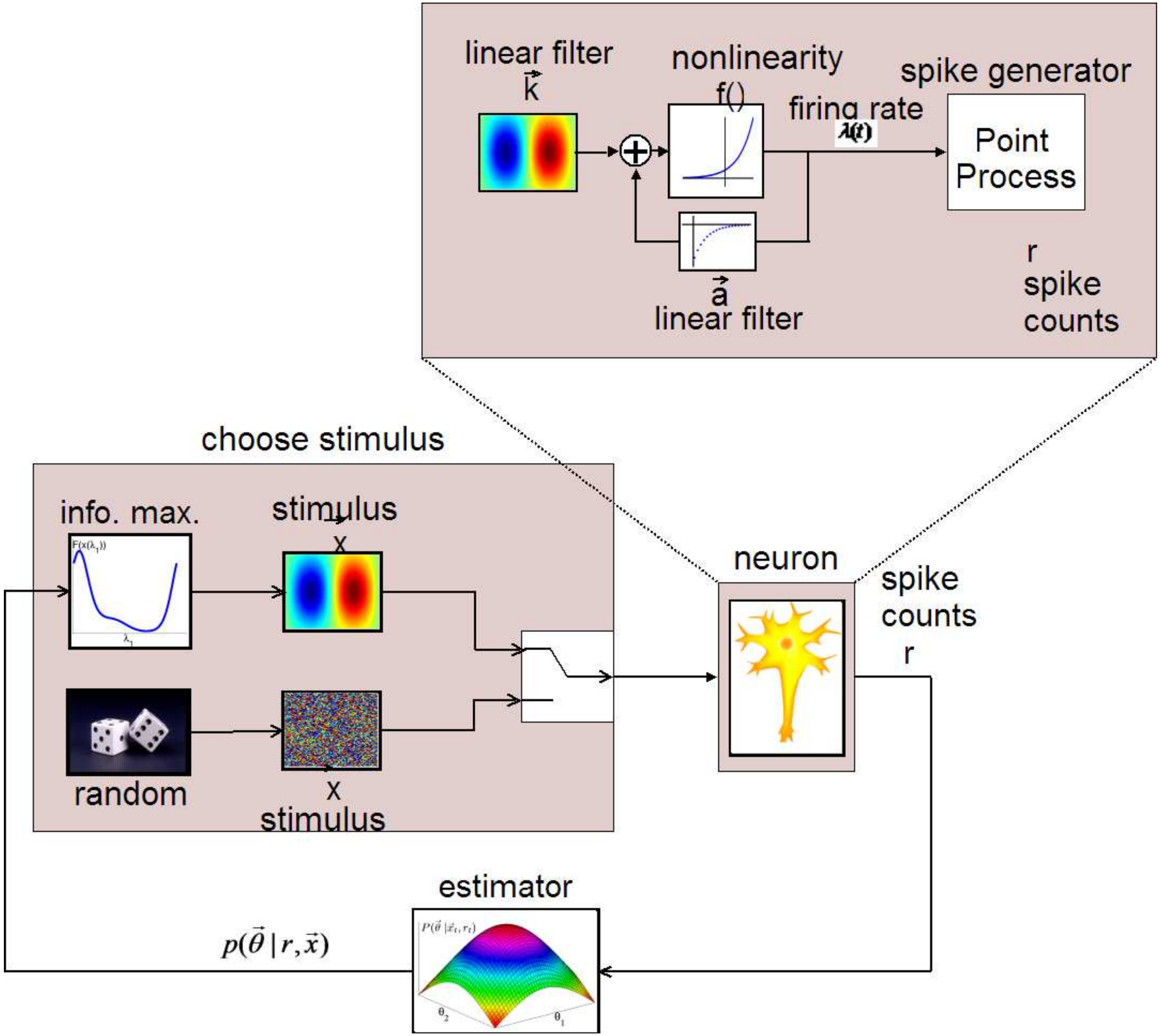
$$r_i = \theta \cdot \vec{x}_i + \epsilon_i, \;\; \epsilon_i \sim \mathcal{N}(0, \sigma^2)$$

- Previous approximations are exact; instead of nonlinear eigenvalue problem, we have standard eigenvalue problem. No dependence on μ_N, just C_N.
- Fisher information does not depend on observed r_i, so optimal sequence $\{\vec{x}_1, \vec{x}_2, \ldots\}$ can be precomputed, since observed r_i do not change optimal strategy.

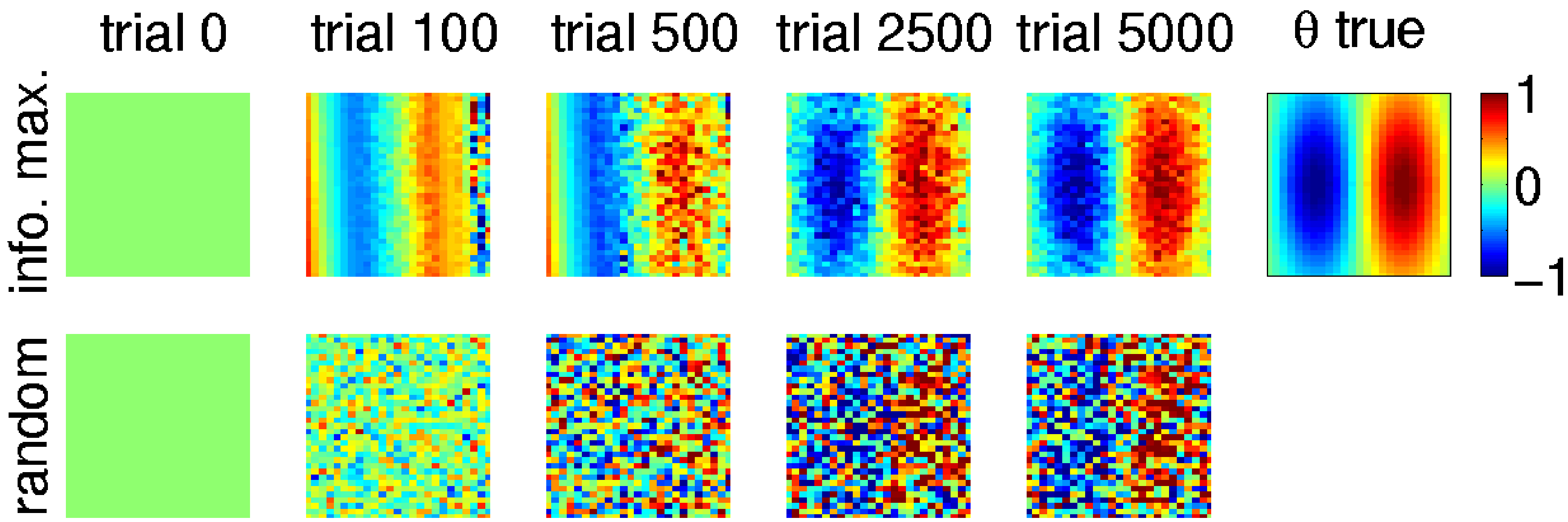


Near real-time adaptive design
Time(Seconds)
0.1
0.01
0.001
0
200
400
600
Dimensionality
total time
diagonalization
posterior update
1d line Search

Simulation overview

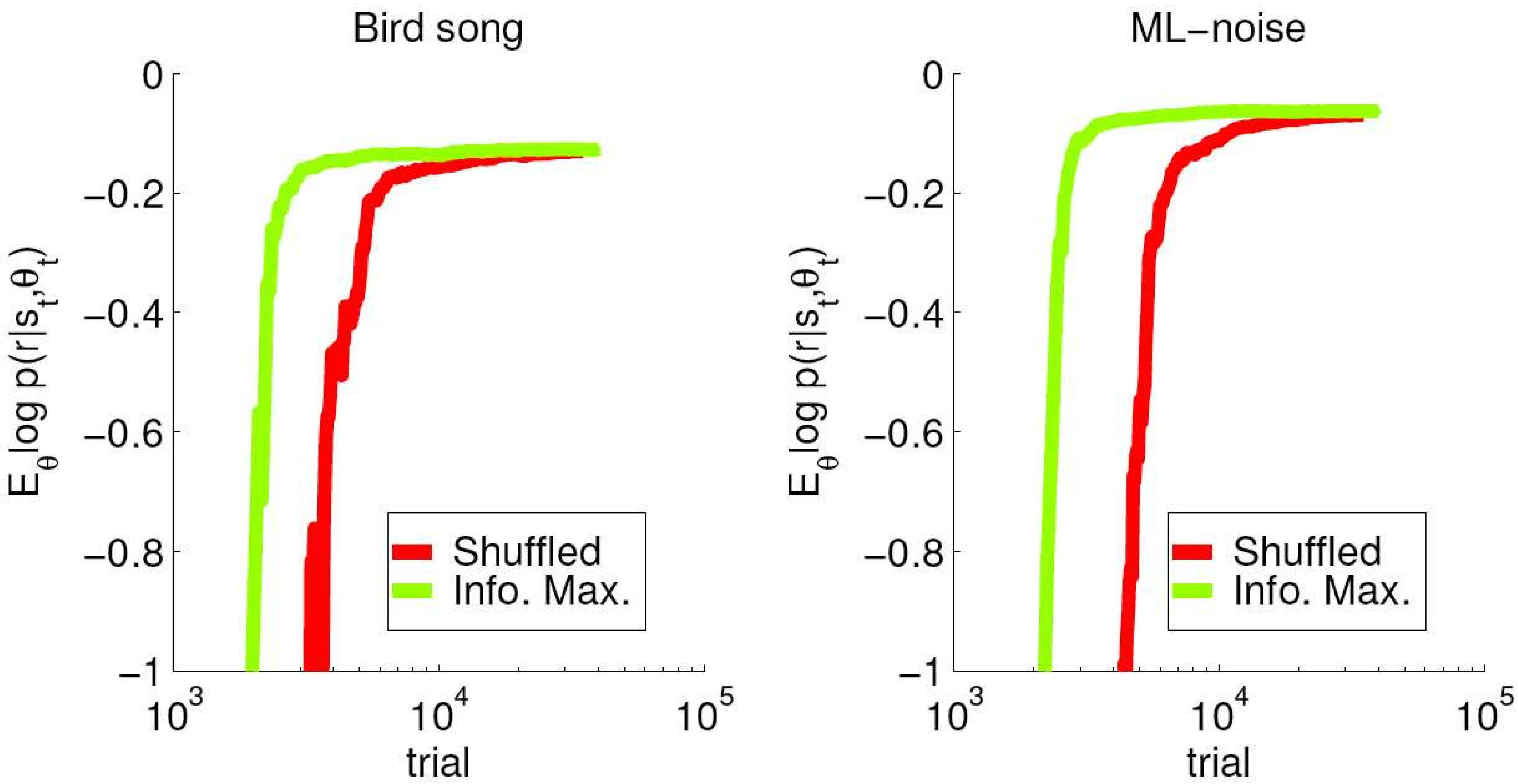


Gabor example



— infomax approach is an order of magnitude more efficient.

Application to songbird data: choosing an optimal stimulus sequence



— stimuli chosen from a fixed pool; greater improvements expected if we can choose arbitrary stimuli on each trial.

Handling nonstationary parameters

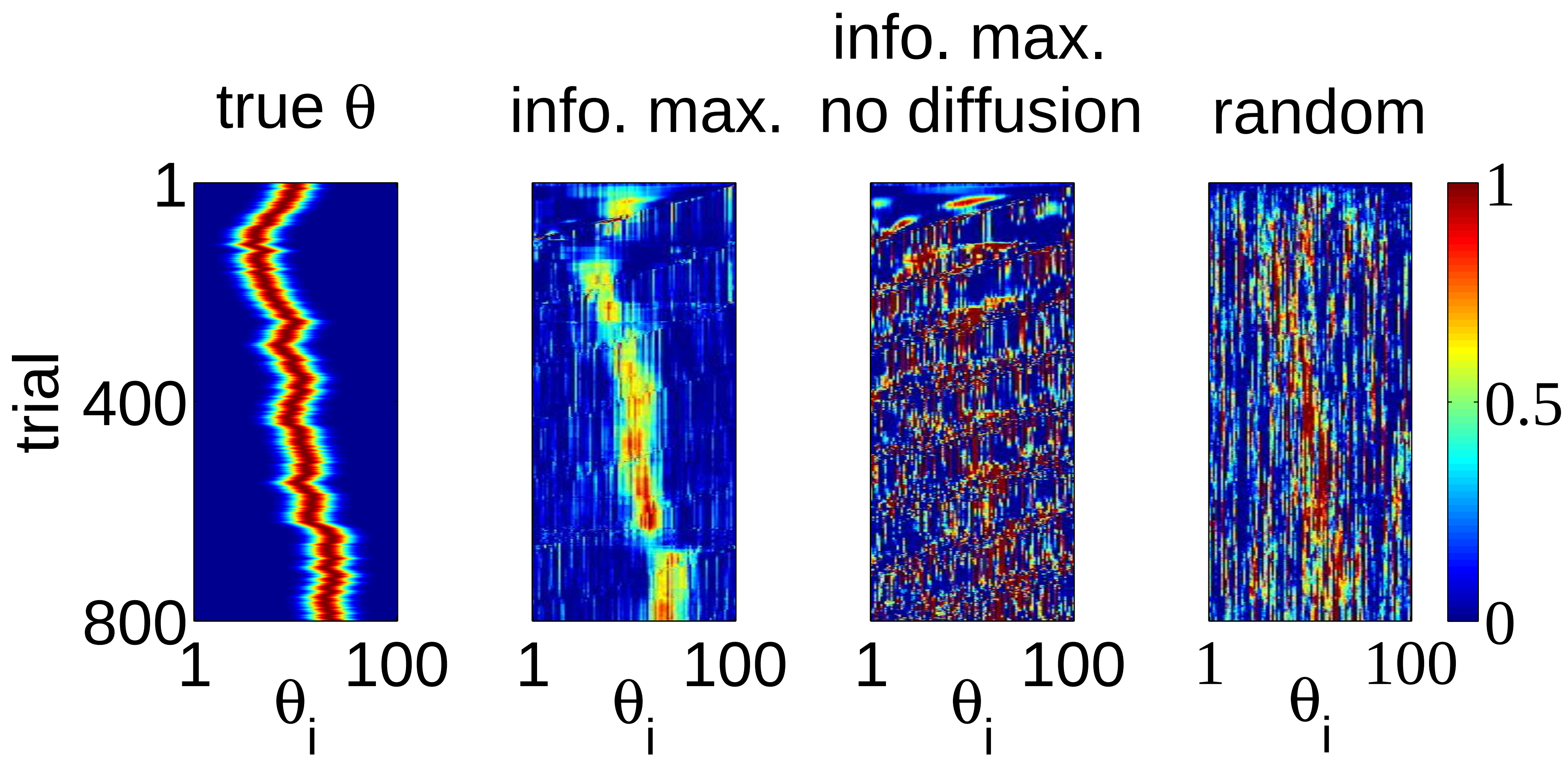
Various sources of nonsystematic nonstationarity:

- Eye position drift
- Changes in arousal / attentive state
- Changes in health / excitability of preparation

Solution: allow diffusion in extended Kalman filter:

$$\vec{\theta}_{N+1} = \vec{\theta}_N + \epsilon; \quad \epsilon \sim \mathcal{N}(0, Q)$$

Nonstationary example



Asymptotic efficiency

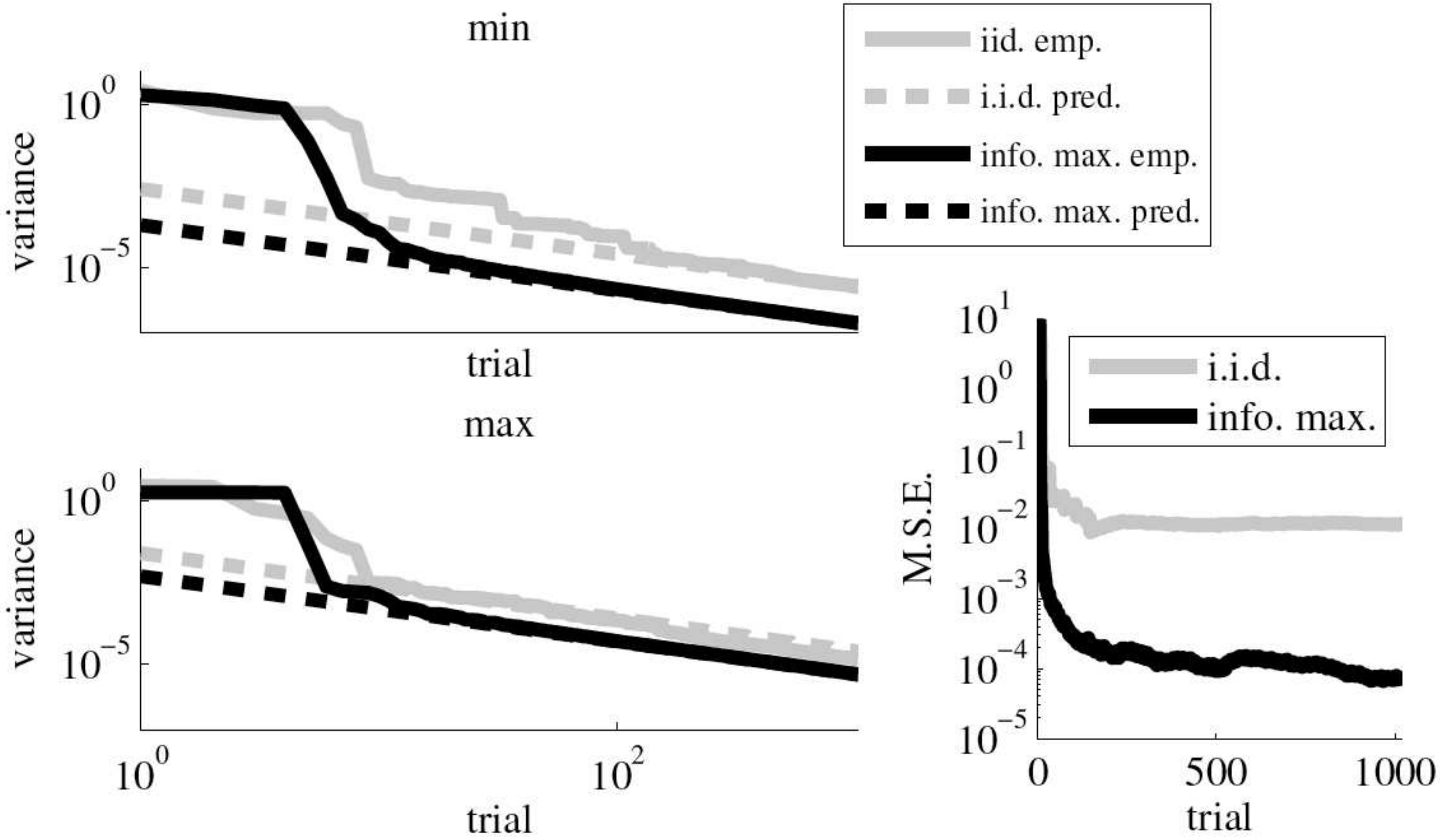
We made a bunch of approximations; do we still achieve correct asymptotic rate?

Recall:

- $(\sigma_{iid}^2)^{-1} = E_x(I_x(\theta_0))$
- $(\sigma_{info}^2)^{-1} = \text{argmax}_{C \in co(I_x(\theta_0))} \log |C|$

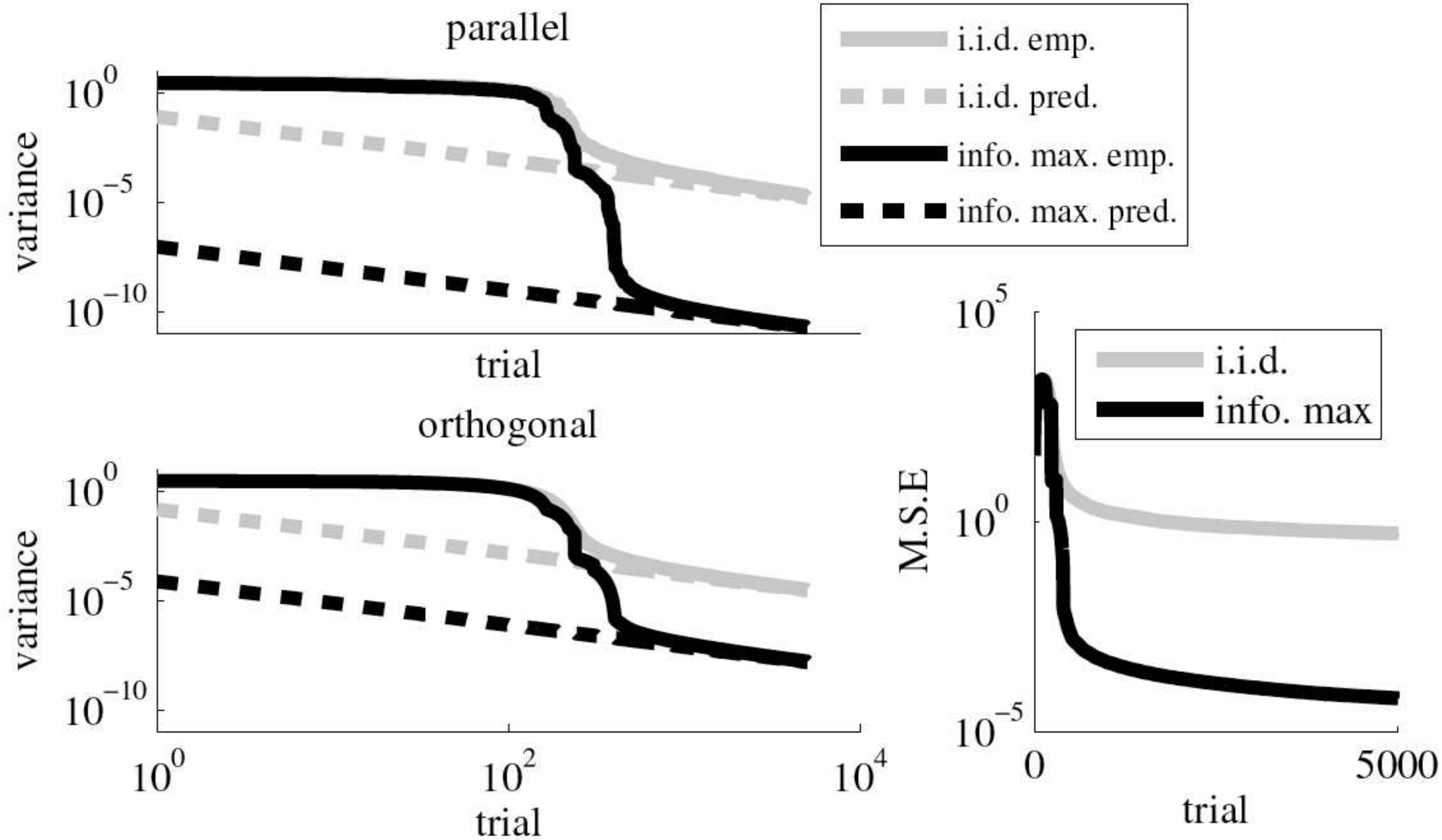
Asymptotic efficiency: finite stimulus set

If $|\mathcal{X}| < \infty$, computing infomax rate is just a finite-dimensional (numerical) convex optimization over $p(x)$.



Asymptotic efficiency: bounded norm case

If $\mathcal{X} = \{\vec{x} : ||\vec{x}||_2 < c < \infty\}$, optimizing over $p(x)$ is now infinite-dimensional, but symmetry arguments reduce this to a two-dimensional problem (Lewi et al., 2009).



— $\sigma^2_{iid}/\sigma^2_{opt} \sim \dim(\vec{x})$: infomax is most efficient in high-d cases

Conclusions

- Three key assumptions/approximations enable real-time ($O(d^2)$) infomax stimulus design:

  — generalized linear model

  — Laplace approximation

  — first-order approximation of log-determinant

- Able to deal with adaptation through spike history terms and nonstationarity through Kalman formulation

- Directions: application to real data; optimizing over sequence of stimuli $\{\vec{x}_t, \vec{x}_{t+1}, \ldots \vec{x}_{t+b}\}$ instead of just next stimulus $\vec{x}_t$.

References

Berkes, P. and Wiskott, L. (2006). On the analysis and interpretation of inhomogeneous quadratic forms as receptive fields. *Neural Computation*, 18:1868–1895.

Gu, M. and Eisenstat, S. (1994). A stable and efficient algorithm for the rank-one modification of the symmetric eigenproblem. *SIAM J. Matrix Anal. Appl.*, 15(4):1266–1276.

Lewi, J., Butera, R., and Paninski, L. (2009). Sequential optimal design of neurophysiology experiments. *Neural Computation*, 21:619–687.

Paninski, L. (2005). Asymptotic theory of information-theoretic experimental design. *Neural Computation*, 17:1480–1507.