State Space Methods in Neuronal Data Analysis

Zhe (Sage) Chen, PhD

zhechen@neurostat.mit.edu

HST Harvard-MIT Health Sciences and Technology

Massachusetts Institute of Technology

The Picower Institute for learning and memory

MGH 1811

VE RI TAS HARVARD

MIT Intelligence Initiative

Objective: what I will do

- Tutorial overview of state space models
- Brief overview of modeling neuronal spike data
- Brief high-level overview of statistical inference methods
- representative examples of neuroscience applications

What I will not do

- Derive each algorithm in detail
- Discuss every possible extension (in model or algorithm)
- Present detailed interpretations & address limitations of each method
- Go through method/conclusion of each neuroscience example

Outline

- **Part I, State space model**: formalism and examples (linear Gaussian SSM, hidden Markov models, …)
- **Part II, Foundation of state space analysis**: Bayes' s rule and recursive Bayesian estimation
- **Part III, Models of neuronal observations**: generalized linear model, point process, goodness-of-fit assessment
- **Part IV, Inference and learning**: Filtering/smoothing, likelihood/Bayesian approaches, EM, VB, EP, Gibbs sampling, MCMC
- **Part V, Applications in Neuroscience**: overview and hand-on examples

Part I: State Space Model

Markov chain and Markov process

- A random process usually characterized as **memoryless**: the next state depends on the current state and not on the sequence of events that preceded it--- Markovian

- Many real-world phenomena: physics, biology, bioinformatics, economics and finance, social sciences, games, documents, music and language, …

Semi-Markov process

- The probability of there being a change in the state depends on the amount of the time that has elapsed since entry into the state

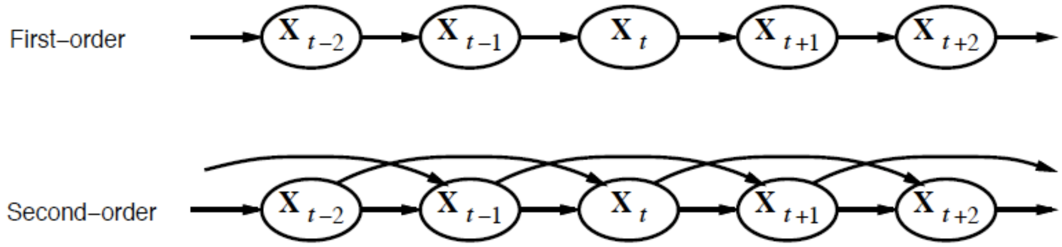
- Example: sleep transition, behavioral transition

Temporal dependence

- Variable-order Markov (VOM) model: context tree

First-order Markov process: $\mathbf{P}(\mathbf{X}_t|\mathbf{X}_{0:t-1}) = \mathbf{P}(\mathbf{X}_t|\mathbf{X}_{t-1})$

Second-order Markov process: $\mathbf{P}(\mathbf{X}_t|\mathbf{X}_{0:t-1}) = \mathbf{P}(\mathbf{X}_t|\mathbf{X}_{t-2}, \mathbf{X}_{t-1})$

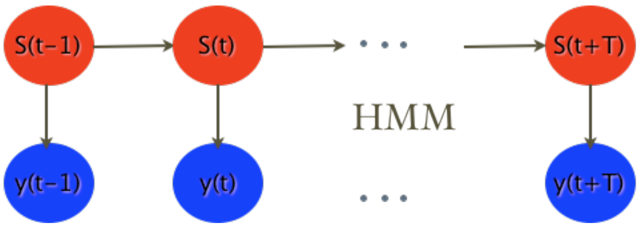


What is state?

- A latent variable of modeling/estimation interest
- Can be either a physical quantity or an abstract notion (e.g., cognitive)
- With known or unknown dimensionality (e.g. speech phoneme/DNA nucleotide ‘ACGT’ /movement kinematics)
- Neural state: single-unit/ensemble/system level

What is state space model (SSM)?

- A class of probabilistic graphical model that describes the statistical dependence and dynamics between latent state and observed measurements
- Other terms: Kalman filter, HMM, latent process model, directed acyclic graph



Formalism

- **State (system, process) equation**: $p(\mathbf{x}_t|\mathbf{x}_{t\text{-}1})$ or $p(S_t|S_{t\text{-}1})$, can be deterministic or stochastic

- **Observation equation**: $p(\mathbf{y}_t|\mathbf{x}_t)$ or $p(\mathbf{y}_t|S_t)$, often stochastic

- Together defines a stochastic dynamical system
- $\mathbf{x}_t$ (or S_t) and $\mathbf{y}_t$ can be continuous, discrete (finite or infinite but countable), or mixed-value

Example: linear Gaussian SSM

- **State equation**: $p(\mathbf{x}_{t+1}|\mathbf{x}_t) \sim N(\mathbf{0},\mathbf{Q})$

$$\mathbf{x}(t+1) = \mathbf{A}\mathbf{x}(t) + \mathbf{n}(t)$$

- **Observation equation**: $p(\mathbf{y}_t|\mathbf{x}_t) \sim N(\mathbf{0},\mathbf{R})$

$$\mathbf{y}(t) = \mathbf{B}\mathbf{x}(t) + \mathbf{v}(t)$$

- Complete data likelihood: $\theta=\{\mathbf{A}, \mathbf{B}, \mathbf{Q}, \mathbf{R}, \mathbf{x}(0)\}$

$$p(X,Y|\theta) = \frac{1}{(2\pi)^{n/2}|\mathbf{Q}|^{1/2}} \exp\left\{\sum_{t=1}^{T_0-1}(-0.5(\mathbf{x}(t+1) - \mathbf{A}\mathbf{x}(t))^T\mathbf{Q}^{-1}(\mathbf{x}(t+1) - \mathbf{A}\mathbf{x}(t))\right\}$$

$$+ \frac{1}{(2\pi)^{m/2}|\mathbf{R}|^{1/2}} \exp\left\{\sum_{t=1}^{T_0}(-0.5(\mathbf{y}(t) - \mathbf{B}\mathbf{x}(t))^T\mathbf{R}^{-1}(\mathbf{y}(t) - \mathbf{B}\mathbf{x}(t))\right\}$$

Note for linear Gaussian SSM

- Special cases: (fully observed) AR and vector AR processes

- Higher-order state dependence can be transformed into first-order dependence by state embedding, e.g. $\mathbf{x}_{new}(t)=[\mathbf{x}(t), \mathbf{x}(t\text{-}1)]$

Finite-state HMM

- Finite m-state $\{S(t)\}$: m-by-m state-transition probability matrix $\boldsymbol{P} = \{P_{ij}\}$, each row vector is a multinomial vector (sum to 1)

  P_{ij} = Prob($S(t+1)=j$ |$S(t)=i$)

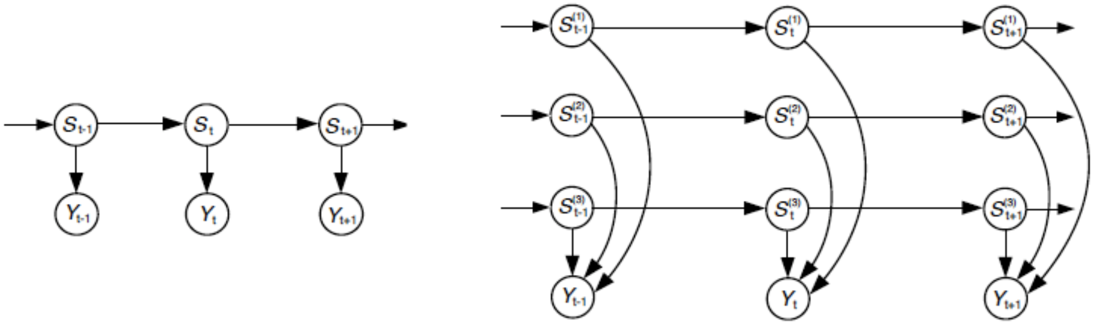
$$\boldsymbol{P} = \begin{pmatrix} P_{11} & P_{12} & \dots & P_{1m} \\ P_{21} & P_{22} & \dots & P_{2m} \\ \vdots & \vdots & \dots & \vdots \\ P_{m1} & P_{m2} & \dots & P_{mm} \end{pmatrix}$$

- Finite discrete observations: m-by-n state-emission probability matrix $\boldsymbol{O} = \{O_{ik}\}$ O_{ik} (t)= Prob($y=k$| $S= i$)

- Continuous/discrete observations: Gaussian /Poisson/binomial distributions

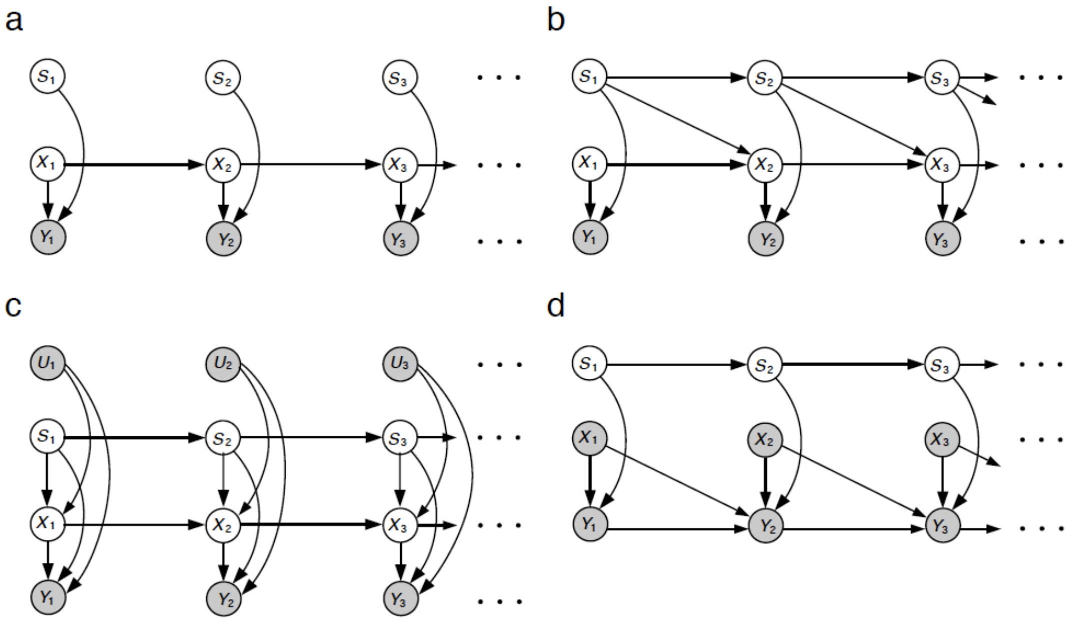
Factorial HMM

- Multiple independent Markov chains



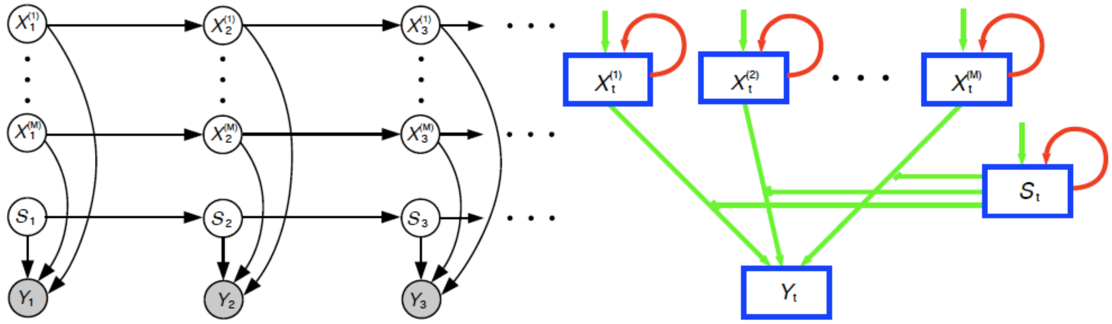
Ghahramani and Jordan (1997) *Machine Learning*

Switching SSM



Switching SSM

- Generalization of mixtures of experts, the states of gating $\{S_t\}$ and experts are all Markovian



Ghahramani and Hinton (1999) *Neural Computation*

Other SSM variants

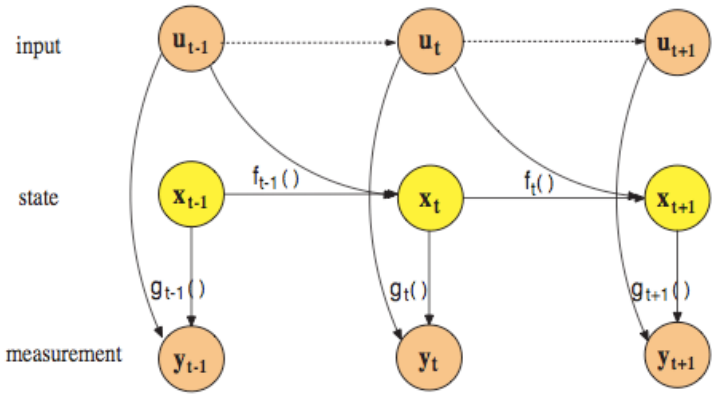
- Observations are non-Gaussian (binary/discrete): generalized linear model (GLM)

- State and/or observation equations involves nonlinearity

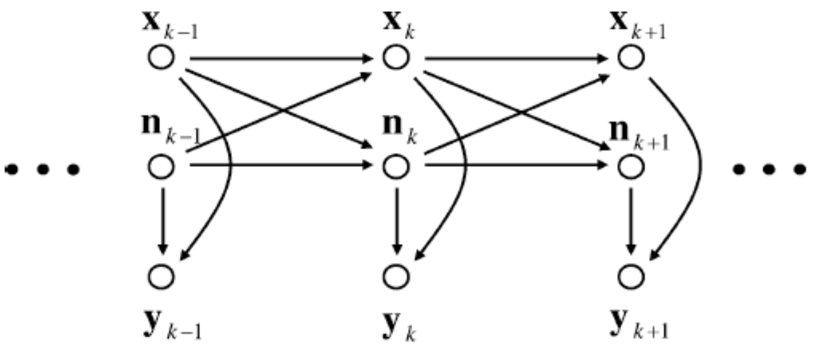
- State equation involves a control variable: optimal control

Generic non-switching SSM

- Control or covariate input u_t
- Nonlinearity (implies non-Gaussianity)



Neuroscience example 1: neural decoding

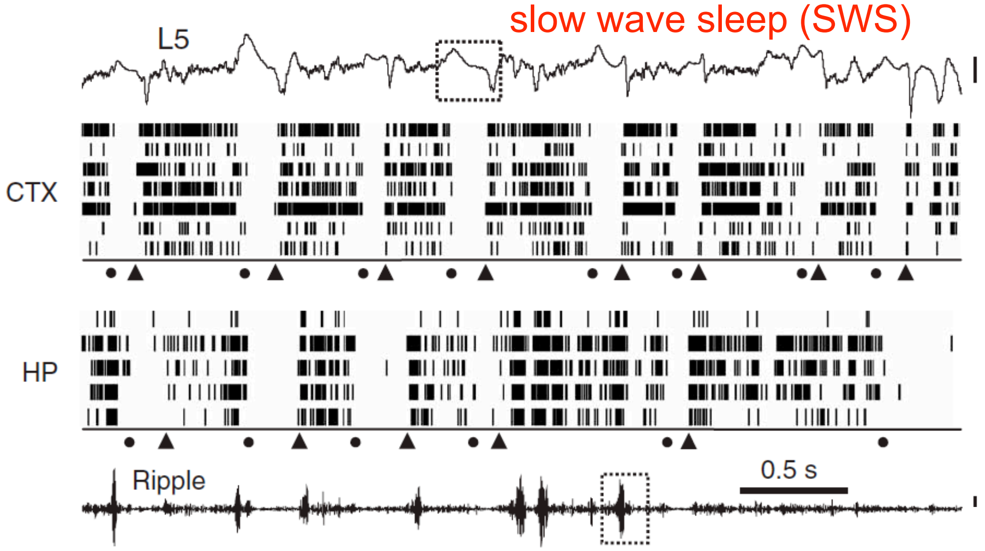


$$\mathbf{y}_k = \mathbf{H}\mathbf{x}_k + \mathbf{G}\mathbf{n}_k + \mathbf{q}_k$$
$$\begin{pmatrix} \mathbf{x}_{k+1} \\ \mathbf{n}_{k+1} \end{pmatrix} = \mathbf{A} \begin{pmatrix} \mathbf{x}_k \\ \mathbf{n}_k \end{pmatrix} + \mathbf{w}_k$$
$$\mathbf{n}_1 \sim N(\mu, \Sigma)$$

x: monkey's hand kinematics
n: hidden state
y: neuronal firing rate vector

Wu et al. (2008) *IEEE TBME*

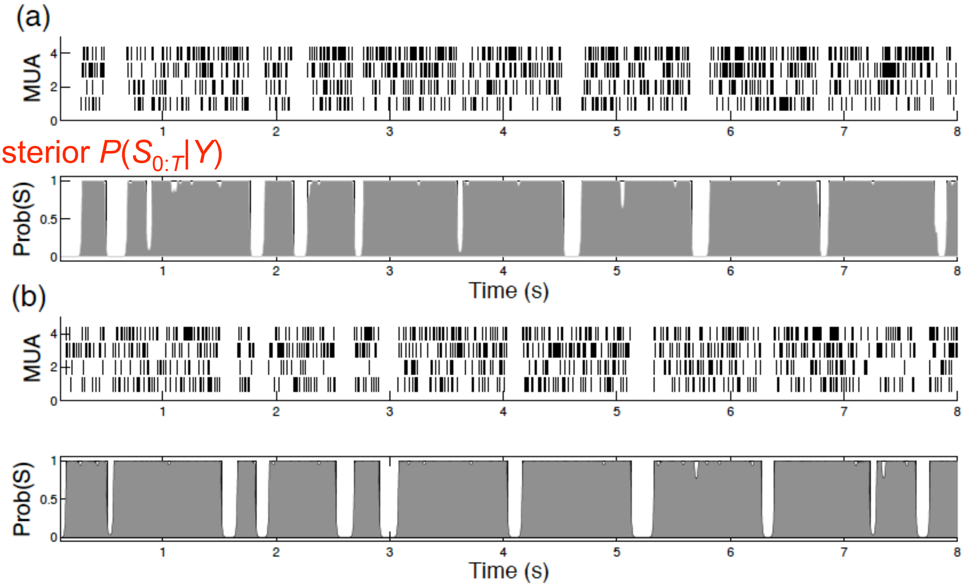
Neuroscience example 2: rat's cortical & hippocampal UP and DOWN states during SWS



slow wave sleep (SWS)

Ji and Wilson (2007) *Nature Neurosci*

Two-state HMM: Markov-driven Poisson firing for multi-unit activity (MUA) in rat S1



posterior $P(S_{0:T}|Y)$

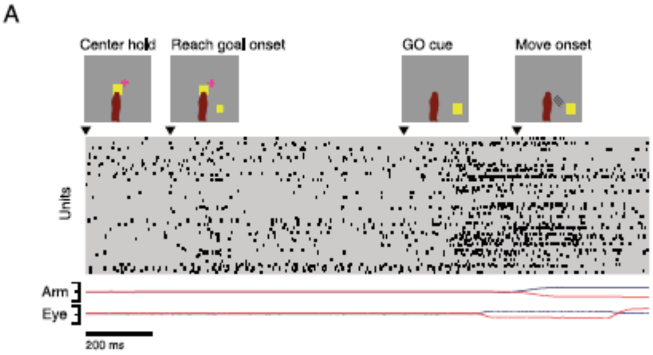
Chen et al. (2009) *Neural Computation*

Neuroscience example 3

mixture of trajectory models

m={1,…, *M*}: reach goal index

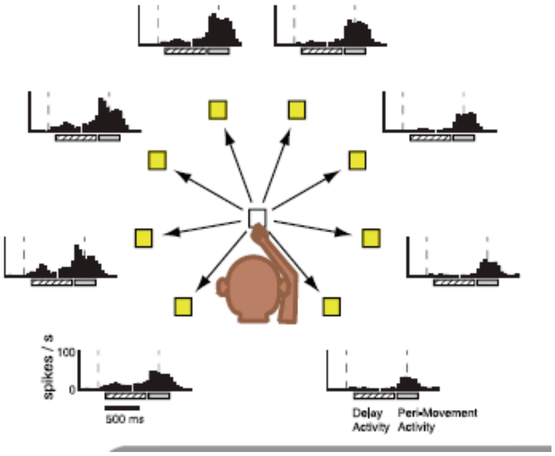
$\mathbf{X}_t$: movement kinematics



$$\mathbf{x}_t|\mathbf{x}_{t-1}, m \sim \mathcal{N}(A_m\mathbf{x}_{t-1} + \mathbf{b}_m, Q_m)$$

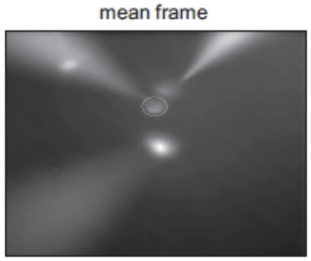
$$\mathbf{x}_1|m \sim \mathcal{N}(\boldsymbol{\pi}_m, \mathrm{V}_m)$$

$$s^{\mathrm{i}}_{t-\mathrm{lag}_i}|\mathbf{x}_t \sim \text{Poisson}\,[\exp(\mathbf{c}_i'\mathbf{x}_t + d_i)\Delta]$$



Yu et al. (2007) *J Neurophysiol*

Neuroscience example 4: deconvolution of spike trains from calcium imaging



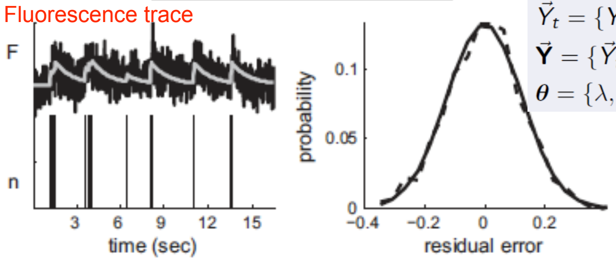
Intracellular Calcium concentration [Ca $^{2+}$]

$$C_t = \gamma C_{t-1} + n_t, \qquad n_t \sim \text{Possion}(\lambda\Delta)$$

$$\vec{Y}_t = \alpha C_t + \epsilon_t, \qquad \epsilon_t \sim \mathcal{N}\left(\alpha\beta, \sigma_Y^2 \vec{\mathbf{I}}\right)$$

Transition distribution: $P_{\boldsymbol{\theta}}(C_t|C_{t-1})$ is Linear-Poisson

Observation distribution: $P_{\boldsymbol{\theta}}(\vec{\mathbf{Y}}_t|C_t)$ is Linear-Gaussian



$$F_t = \alpha[\mathrm{Ca}^{2+}]_t + \beta + \varepsilon_t, \qquad \varepsilon_t \overset{iid}{\sim} \mathcal{N}(0, \sigma^2).$$

$\vec{Y}_t = \{Y_{x,t}\}_{x=0}^{P}$ is the t-th image frame

$\vec{\mathbf{Y}} = \{\vec{Y}_t\}_{t=1}^{T}$ is the entire movie

$\boldsymbol{\theta} = \{\lambda, \tau, \sigma_c, \alpha, \beta, \sigma_Y\}$ is the set of model parameters

$\mathbf{n} = \{n_t\}_{t=0}^{T}$ is the spike train

Likelihood: $P_{\boldsymbol{\theta}}(\vec{\mathbf{Y}}|\mathbf{n}) = \mathcal{N}(\vec{\mathbf{Y}}; \boldsymbol{\mu}, \boldsymbol{\Sigma})$

Prior: $P_{\boldsymbol{\theta}}(\mathbf{n}) = \text{Poisson}(\lambda\Delta)$

want the posterior: $P_{\boldsymbol{\theta}}(\mathbf{n}|\vec{\mathbf{Y}})$

know Bayes Rule: $P_{\boldsymbol{\theta}}(\mathbf{n}|\vec{\mathbf{Y}}) \propto P_{\boldsymbol{\theta}}(\vec{\mathbf{Y}}|\mathbf{n})P_{\boldsymbol{\theta}}(\mathbf{n})$

Vogelstein et al. (2010) *J Neurophysiol*

Neuroscience example 5: behavioral facilitation with DBS

State-space analysis

$$p(\theta, x|n) = \frac{p(n|x,\theta)p(x|\theta)p(x_0)p(\theta)}{p(n)}$$

where   *n*: behavioral response (0/1)

$$p(n|x,\theta) = \prod_{k=1}^{K} p_k^{n_k}(1-p_k)^{1-n_k}$$

for the Bernoulli model

$$p(n|x,\theta) = \prod_{k=1}^{K} \binom{N_k}{n_k} p_k^{n_k}(1-p_k)^{N_k-n_k}$$

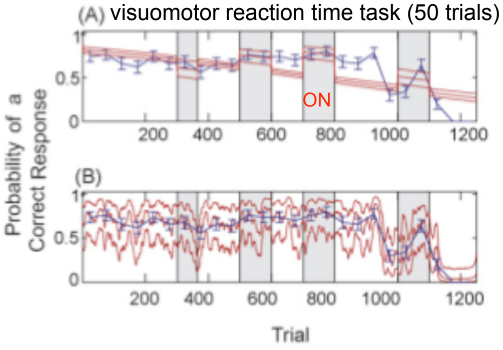
for the binomial model   *n*: range 0~4

x: arousal and attention state

$$x_k = \rho x_{k-1} + \varepsilon_k$$

$$Pr(n_k|x_k,\theta) = p_k^{n_k}(1-p_k)^{1-n_k}$$

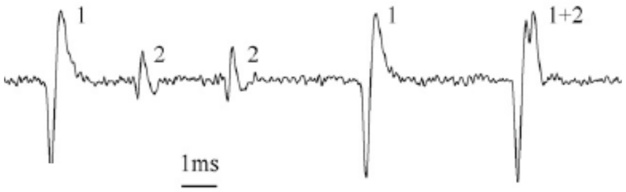
$$\log[p_k(1-p_k)^{-1}] = x_k.$$

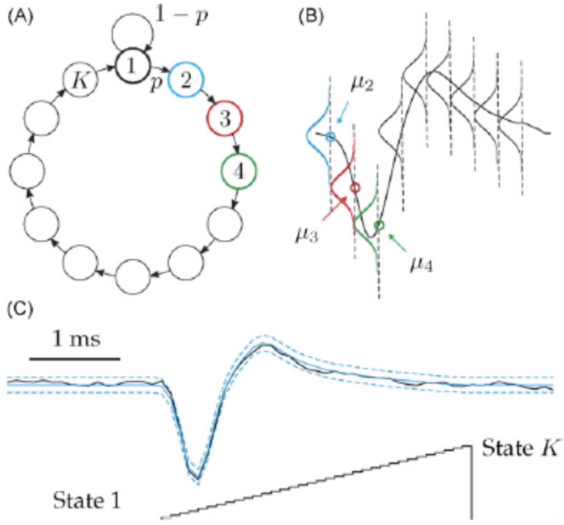


(A) visuomotor reaction time task (50 trials)

Smith et al. (2009) *J Neurosci Methods*

Factorial HMM for spike sorting





P(S_{t+1}=2|S_t=1)=*p*

P(S_{t+1}=1|S_t=1)=1-*p*

P(S_{t+1}=*i* |S_t= *i*-1)=1   *i*>2

P(O_t|S_t=*k*)=*N*(μ_k;σ^2)

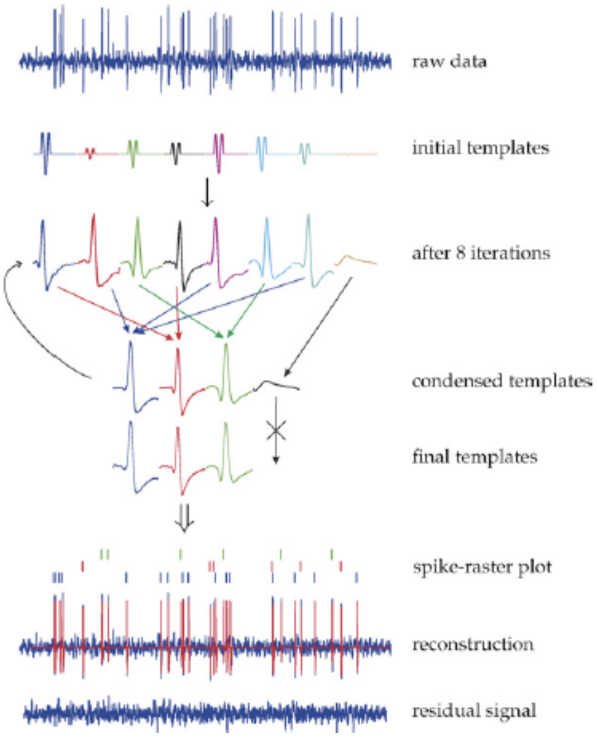
K: for a sample rate of 30 kHz, refractory period of 1 ms → *K*=30

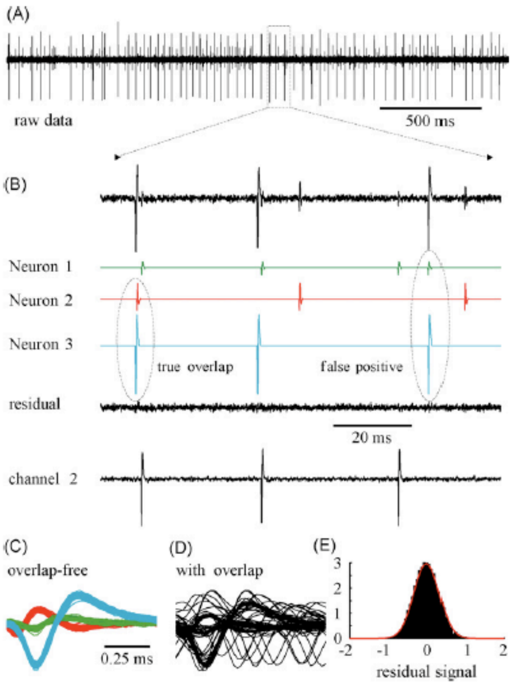
N neurons per electrode

$$P(\underline{S}_t|\underline{S}_{t-1}) = \prod^{N} P(S_t^n|S_{t-1}^n).$$

$$\underline{\mu}(k_1, k_2, \ldots, k_N) = \sum_{n=1}^{N} \mu_{k_n}^n,$$

Herbst J et al. (2008) *J Neurosci Methods*





Short summary

- Procedure of neural data analysis with SSM

1) Understand the underlying scientific question
2) Make assumptions (and justify them)
3) Come up a statistical model that incorporates all known information
4) Fit the model with data
5) Model assessment
6) Refine the model
7) Result interpretation

Part II: Foundation of State Space Analysis

Two tasks

- **State-space modeling**: seek a statistical model that best characterizes the data (dependence, structure, etc) according to some statistical criterion

- **State-space analysis**: compute the "optimal" estimate (in a pre-defined statistical sense) of hidden state given observed data

Bayes' rule

(after the Reverend Thomas Bayes)

conditional, joint, marginal joint probabilities
product rule & sum rule

$$p(X|Y) = \frac{p(X,Y)}{p(Y)} = \frac{p(Y|X)p(X)}{p(Y)} = \frac{p(Y|X)p(X)}{\int p(Y|X)p(X)dX}$$

$$p(X_i|Y) = \frac{p(X_i,Y)}{p(Y)} = \frac{p(Y|X_i)p(X_i)}{p(Y)} = \frac{p(Y|X_i)p(X_i)}{\sum_j p(Y|X_j)p(X_j)}$$

Recursive Bayesian estimation

- Assumptions: First-order Markovian in {**x**(*t*)}, conditional independence between {**y**(*t*)}

$$p(\mathbf{x}(t)|\mathbf{y}(0:t)) = \frac{p(\mathbf{x}(t),\mathbf{y}(0:t))}{p(\mathbf{y}(0:t))} = \frac{p(\mathbf{x}(t)|\mathbf{y}(0:t-1)p(\mathbf{y}(0:t)|\mathbf{x}(t),\mathbf{y}(0:t-1))}{p(\mathbf{y}(t)|\mathbf{y}(0:t-1))}$$

$$= \frac{p(\mathbf{x}(t)|\mathbf{y}(0:t-1))p(\mathbf{y}(t)|\mathbf{x}(t),\mathbf{y}(0:t-1))}{p(\mathbf{y}(t)|\mathbf{y}(0:t-1))}$$

Optimal filtering & optimality

- Minimum mean squared error→ conditional mean

$$\mathbb{E}[\|\mathbf{x}_n - \hat{\mathbf{x}}_n\|^2|\mathbf{y}_{0:n}] = \int \|\mathbf{x}_n - \hat{\mathbf{x}}_n\|^2 p(\mathbf{x}_n|\mathbf{y}_{0:n})d\mathbf{x}_n$$

- Maximum a posteriori (MAP)→ conditional mode

$$\mathcal{E} = \mathbb{E}[1 - \mathbb{I}_{\mathbf{x}_n:\|\mathbf{x}_n - \hat{\mathbf{x}}_n\| \le \zeta}(\mathbf{x}_n)]$$

- Maximum likelihood (ML) (uniform or flat prior)

- Minimax → conditional median (robust estimation)

Prediction/Filtering/Smoothing

- **One-step prediction**: *Chapman-Kolmogorov equation*

$$p(\mathbf{x}(t)|\mathbf{y}(0:t-1)) = \int p(\mathbf{x}(t)|\mathbf{x}(t-1))p(\mathbf{x}(t-1)|\mathbf{y}(0:t-1))d\mathbf{x}(t-1)$$

- **Filtering**: compute posterior density: $p(\mathbf{x}(t)|\mathbf{y}(0{:}t))$

- **Smoothing**: computer posterior density
  1) *fixed-lag* smoothing: $p(\mathbf{x}(t)|\mathbf{y}(0{:}t+L))$
  2) *fixed-interval* smoothing: $p(\mathbf{x}(t)|\mathbf{y}(0{:}T))$

Kalman filtering

- Prediction (*a priori* estimate)

$$\begin{aligned}\hat{\mathbf{x}}_{t|t-1} &= \mathbf{A}\hat{\mathbf{x}}_{t-1|t-1}\\ \mathbf{P}_{t|t-1} &= \mathbf{A}\mathbf{P}_{t-1|t-1}\mathbf{A}^\top + \mathbf{Q}\end{aligned}$$

- Update (*a posteriori* estimate)

[innovations covariance]

Kalman gain

$$\begin{aligned}\mathbf{K}_t &= \mathbf{P}_{t|t-1}\mathbf{B}^\top\left[\mathbf{B}\mathbf{P}_{t|t-1}\mathbf{B}^\top + \mathbf{R}\right]^{-1}\\ \hat{\mathbf{x}}_{t|t} &= \hat{\mathbf{x}}_{t|t-1} + \mathbf{K}_t(\mathbf{y}_t - \mathbf{B}\hat{\mathbf{x}}_{t|t-1})\\ \mathbf{P}_{t|t} &= \mathbf{A}\mathbf{P}_{t-1|t-1}\mathbf{A}^\top + \mathbf{Q}\end{aligned}$$

Error correction = Kalman gain × (innovations)

- Variants: information filter

Kalman smoothing

- Fixed-lag smoother
- Fixed-interval RTS smoother

1) forward pass: Kalman filtering

2) backward pass:

$$\begin{aligned}\mathbf{C}_t &= \mathbf{P}_{t|t}\mathbf{A}^\top\mathbf{P}_{t+1|t}^{-1}\\ \hat{\mathbf{x}}_{t|T} &= \hat{\mathbf{x}}_{t|t} + \mathbf{C}_t(\mathbf{x}_{t+1|T} - \hat{\mathbf{x}}_{t+1|t})\\ \mathbf{P}_{t|T} &= \mathbf{P}_{t|t} + \mathbf{C}_t(\mathbf{P}_{t+1|T} - \mathbf{P}_{t+1|t})\mathbf{C}_t^\top\end{aligned}$$

Bayesian filtering: beyond Kalman filtering

- How to overcome the limitations of Gaussianity and nonlinearity
- Linearization: extended Kalman filter
- Gaussian approximation: Gaussian-sum filter
- Deterministic approximation: Unscented Kalman filter
- Numerical approximation: cubature, quadrature
- Stochastic approximation: Monte Carlo

When states are finite and discrete

- Kalman filter → HMM filter (observations can be multinomial or Gaussian)
- Learning: Baum-Welch algorithm O(m^2T): how to deal with large *m*?
- Inference: Viterbi algorithm (dynamical programming)
- When states are mixed (e.g., switching SSM), exact inference is often computationally intractable

Practical issues

- Naïve data transformation for Gaussian approximation: square root and variance stabilization: Poisson→ Gaussian
- Data representation and variance: coordinate transformation or reparameterization
- How to perform Gaussian approximation?
- Missing data & divergence
- Constraints in the state estimate (equality or inequality)

Short summary

- State space modeling and estimation: filtering /smoothing/prediction
- Define the “optimal criterion” for specific estimation problems
- Aware of the assumptions and limitations of individual estimation method

Part III: Models of Neuronal Spike Trains

Neural encoding

- Establish a statistical model between stimulus input and neuronal responses (not necessarily a biophysical model)
- Neuronal firing: stochastic, dynamic and non-stationary (across time and trials)
- Missing (unobserved) variables: many other neurons, common modulatory input, intent, …

Exponential family of distributions

One class of probability distributions

$$L(\theta) = f_X(x|\theta) = h(x)\ \exp[\ \eta(\theta) \cdot T(x)\ -\ A(\theta)\ ]$$
$$= \exp[\ \eta(\theta) \cdot T(x)\ -\ A(\theta) + B(x)\ ]$$

where *x* is a vector of measurements, θ denotes the parameter

T(*x*) is a sufficient statistic of the distribution,

η is a natural parameter, *A*(*η*) is a log-partition function

Canonical form: *η*(θ)=θ

$$f_X(x|\boldsymbol{\theta}) = h(x) g(\boldsymbol{\theta}) \exp\Big(\ \boldsymbol{\eta}(\boldsymbol{\theta}) \cdot \mathbf{T}(x)\ \Big)$$

Natural form (parameterized by natural parameter)

$$f_X(x|\boldsymbol{\eta}) = h(x) \exp\Big(\ \boldsymbol{\eta} \cdot \mathbf{T}(x) - A(\boldsymbol{\eta})\ \Big)$$

Generalized linear model (GLM)

- Linear regression model:

$$y = X\beta + \varepsilon \qquad\qquad \varepsilon \sim N(0, \Sigma)$$

are useful for relating continuous-valued observations to a set of covariates (can be nonlinear).

- Generalized linear models extend a simple class of models to non-Gaussian data.

natural parameters $\eta(\theta) = X\beta$

Count data:

$$\log\left(\hat{\lambda}\right) = X\beta$$

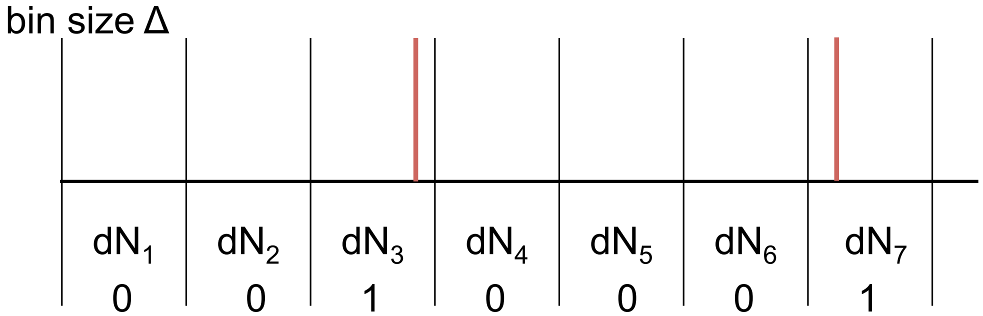
Binary data:

$$\log\left(\frac{\hat{p}}{1-\hat{p}}\right) = X\beta$$

Random processes

- Memoryless: Poisson process (IEI i.i.d. exponential), renewal process (IEI i.i.d. but non-exponential)
- Point process (simple, spatiotemporal, marked)
- doubly stochastic point process

Discrete-time spike train as a point process



dN_1	dN_2	dN_3	dN_4	dN_5	dN_6	dN_7
0	0	1	0	0	0	1

dN_t =N(t+Δ)-N(t) is the spike indicator function in ((t-1)Δ, tΔ]

Conditional intensity function: probability of spiking at time t, given the history H_t

$$\lambda(t \mid H_t) = \lim_{\Delta t \to 0} \frac{\Pr(\text{Spike in } (t, t+\Delta t) \mid H_t)}{\Delta t}$$

GLM for the spike data

Link function	Distribution	Equation
logit	Binomial	$\log\left(\frac{p_k}{1-p_k}\right)=\alpha_0+\sum_{j=1}^{order}\alpha_j dN_{k-j}$
log	Poisson	$\log\left(\lambda_k\right)=\beta_0+\sum_{j=1}^{order}\beta_j dN_{k-j}$

Conditional intensity function

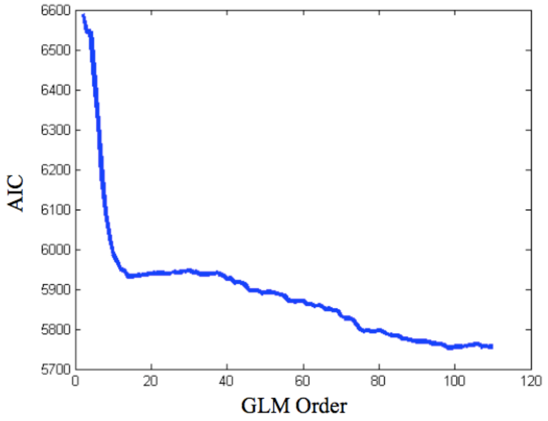
$$
\begin{aligned}
\log(\lambda_k) = \theta_0 &+ \sum_{i=1}^{I}\alpha_i f_i(\text{Extrinsic Covariates}) \\
&+ \sum_{j=1}^{J}\beta_j g_j(\text{Spiking History}) \\
&+ \sum_{k=1}^{K}\sum_{c=1}^{C}\gamma_{k,c} h_{k,c}(\text{Ensemble Activity})
\end{aligned}
$$

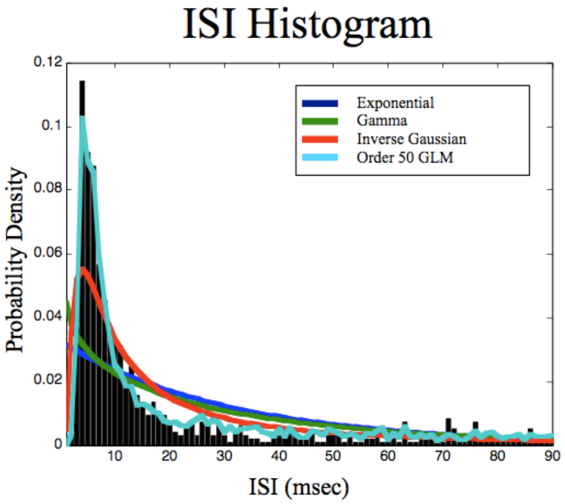
- By selecting an appropriate set of basis functions we can capture arbitrary functional relations.
- Analysis of relative contributions of components to spiking

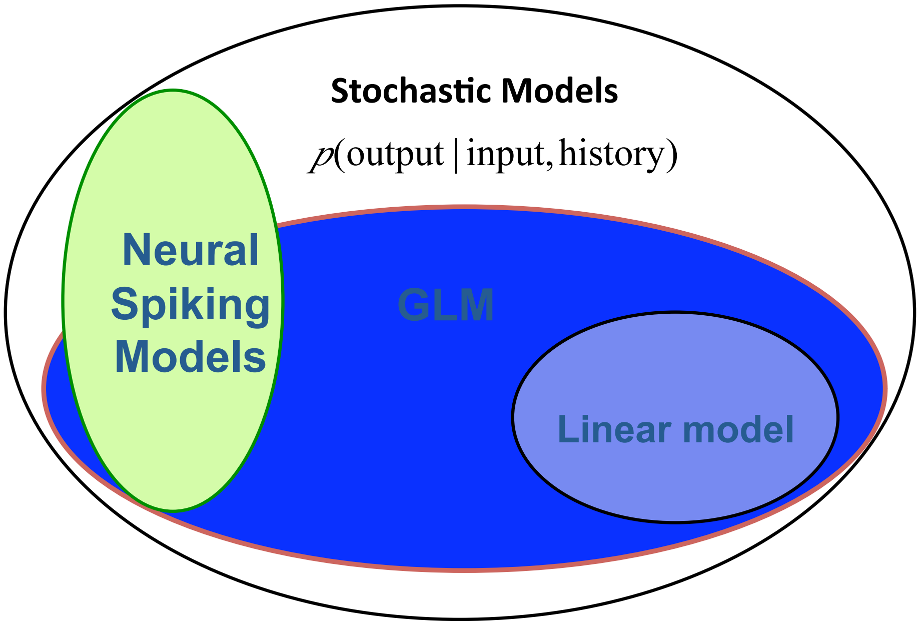
Truccolo W et al. (2004) *J Neurophysiol* 93:1074-1089

Model selection

- Heuristic: ISI histogram, partial correlation
- Statistical criterion: AIC, BIC (only works well asymptotically)
- Cross-validation







Stochastic Models

$p(\text{output} \mid \text{input}, \text{history})$

Neural Spiking Models

GLM

Linear model

Fitting GLM

- Newton method
- Iterative reweighted least squares (IRWLS)
- Conjugate gradient (efficient if sparse)

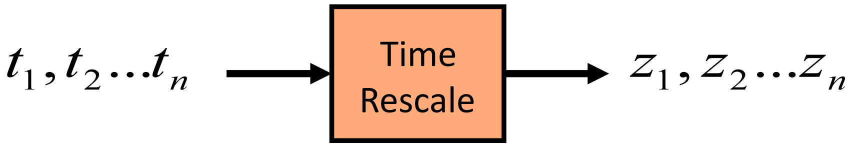
- Matlab: "glmfit"

Properties of GLM:

- Concave likelihood surface
- Estimators asymptotically have minimum MSE

Time-Rescaling Theorem

$$z_i = \int_{t_i}^{t_{i+1}} \lambda(u \mid H_u)du$$



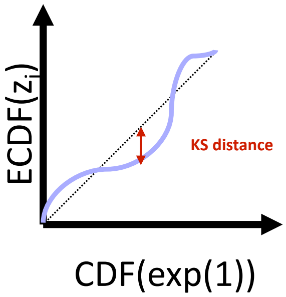
Time-Rescaling Theorem (Brown et al., 2002)***:*** If the estimated CIF is correct, any point process can be time-rescaled into a unit-rate Poisson process. z_i's are i.i.d. (identically and independently distributed) exponential rate 1.

Goodness-of-fit Tests

Kolmogorov-Smirnov (KS) Plot

Graphical measure of goodness-of-fit: If the model is correct, then the time-rescaled interspike intervals are i.i.d. Expn(1), and the ordered quantiles from empirical cdf and true cdf shall produce a 45° line.

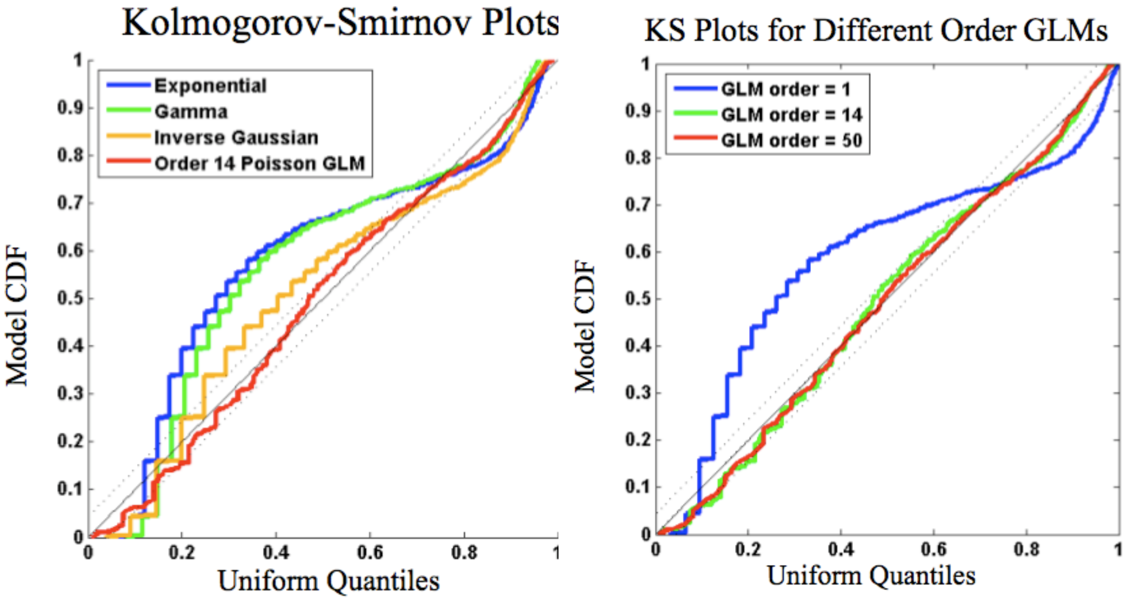
one-sample KS test: between empirical & theoretical CDF



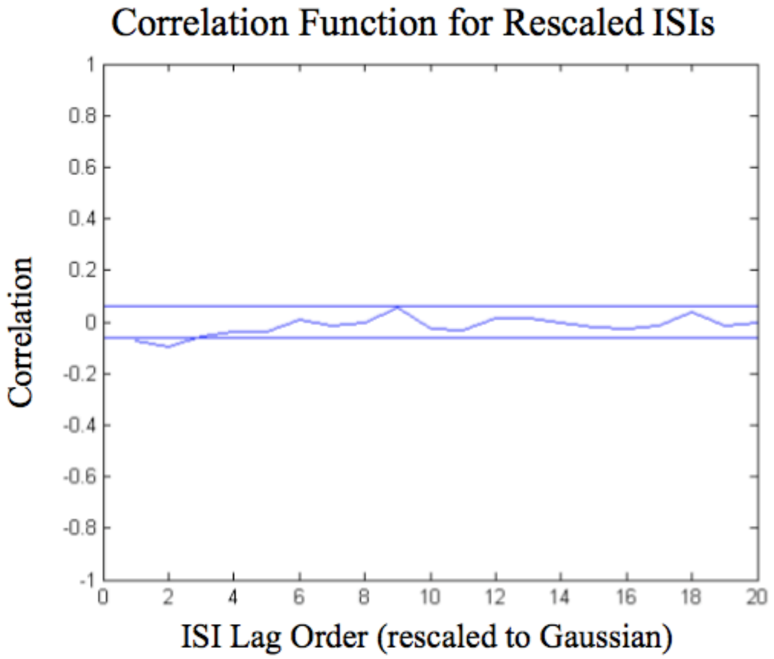
KS statistic (KS distance) is the maximum distance between an empirical and a theoretical probability distribution.

$$KS_{dis\tan ce} = \max | F_n(u) - F(u) |$$

KS plot example

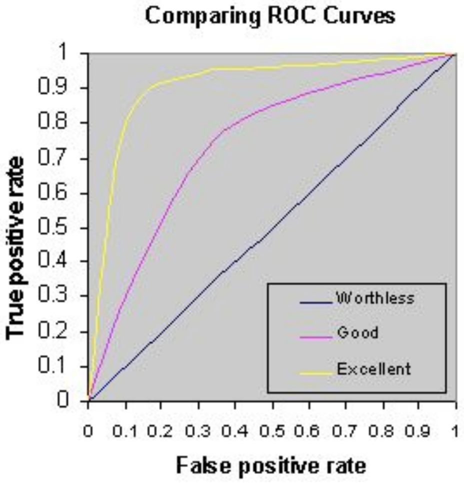


Test the independence



Test predictability: AUC

- ROC (receiver operating characteristic curve)
- AUC =0.5 (chance level), 1.0 (perfect)



Short summary

- Neural spike trains as point processes (fine timescale)
- Can be well approximated with the GLM framework (but sometimes it GLM might not be efficient)
- GLM inference is fast
- Goodness-of-fit assessment: KS plot, ACC plot, predictive likelihood (on unseen data), ROC-AUC

Part IV: Inference and Learning

Statistical data analysis

- Modeling → assessment → interpretation
- Model assumption & model mismatch: "*All models are wrong, but some are useful*"– George E. P. Box
- Goal: inferring the unknowns (model, parameters, data, …) and predicting the future (unseen data)

Mathematics of learning

The problem of learning consists of

- **Estimation**: statistical problem (seek a hypothesis space among a large class of spaces)
- **Approximation**: representation problem (approximate the hypothesis space, trade-off between bias and variance)
- **Computation**: numerical optimization problem (define the cost function and learning procedure)

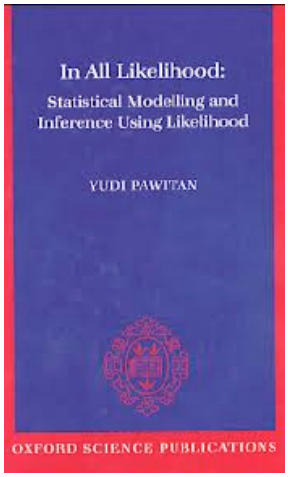
Statistical (machine) learning

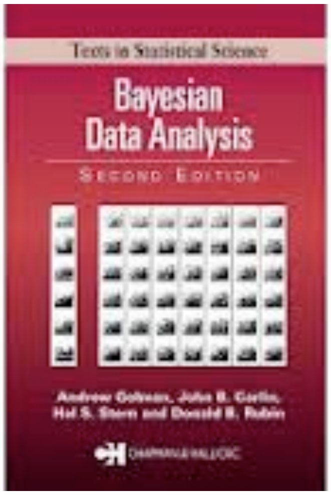
- **Supervised**: labeled data (e.g., regression and classification)
- **Unsupervised**: unlabeled data (e.g., clustering, density estimation, segmentation)
- **Semi-supervised**: partially labeled data
- **Reinforcement**: temporal credit assignment problem, optimal control

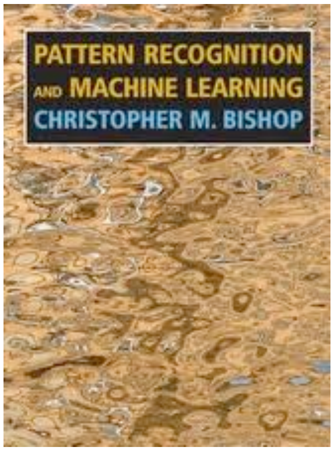
Two approaches for state and/or parameter estimation

- **Likelihood approach**: point estimate, uncertainty is represented by confidence intervals
  - asymptotically: consistent/normal/efficient
  - functional (reparameterization) invariance
  - data overfitting (for a complex model)
- **Bayesian approach**: full posterior, let data speak for themselves
  - rules of probability theory
  - modeling uncertainties (not necessarily random)
  - no overfitting

Recommended book references


In All Likelihood:
Statistical Modelling and Inference Using Likelihood
YUDI PAWITAN
OXFORD SCIENCE PUBLICATIONS


Texts in Statistical Science
Bayesian Data Analysis
SECOND EDITION
Andrew Gelman, John B. Carlin, Hal S. Stern and Donald B. Rubin
CHAPMAN & HALL/CRC



Likelihood

- Probability of data as a function of the parameter is the likelihood of the parameter θ
- Observed data likelihood: $p(Y|X,\theta)$---- ML inference
- Complete data likelihood: $p(X,Y|\theta)$
- Expected complete data log-likelihood: $E_x[\log p(X,Y|\theta)]$ --- EM algorithm
- Marginal likelihood: $p(Y)$ --- VB inference

EM algorithm

- **E-step**: Compute the complete data log-likelihood (Q -function) based on the estimate $\theta^{(k)}$

$$Q(\theta|\theta^{(k)}) = E[\log p(X,Y|\theta)||\theta^{(k)}]$$

and compute the expected statistics with respect to the latent process, e.g., $E[x(t)||\theta^{(k)}], E[x^2(t)||\theta^{(k)}]$ and $E[x(t)x(t+1)||\theta^{(k)}]$

- **M-step**: Update $\theta^{(k)}$ to $\theta^{(k+1)}$ s.t. $\theta^{(k+1)} = \arg\max_\theta Q(\theta|\theta^{(k)})$
  This is can be achieved by setting $\frac{\partial Q}{\partial \theta} = 0$

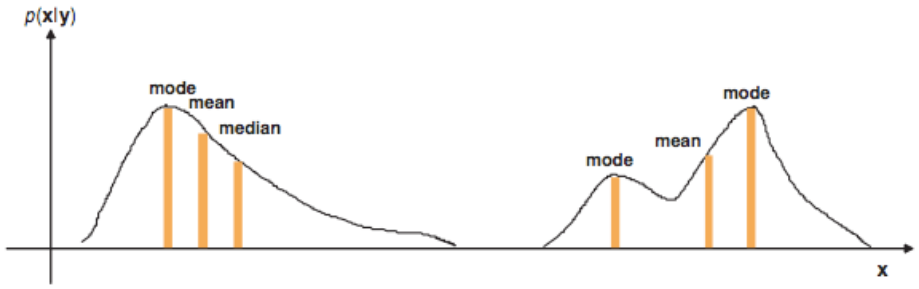
The E- or M-step can be either exact or approximate.

Note

- *Majorize-maximization*: instead of maximize the likelihood, maximize a surrogate function (lower bound), which is concave w.r.t. the parameters (EM is a special case of MM)
- EM can be viewed as a variational technique (Neal and Hinton, 1999)
- EM works well for both static and dynamic models
- Link from EM to Bayesian: iterative optimization via data augmentation (auxiliary variables/simulated samples)---Gibbs sampling is a special case

What is about Bayesian?

- **Goal**: estimate full posterior distribution, posteriori mean/mode/variance, predictive posterior



Bayesian computation (three basic operations)

- Normalization $$p(\mathbf{x}|\mathbf{y}) = \frac{p(\mathbf{y}|\mathbf{x})p(\mathbf{x})}{\int_X p(\mathbf{y}|\mathbf{x})p(\mathbf{x})d\mathbf{x}}$$
- Marginalization $$p(\mathbf{x}|\mathbf{y}) = \int_Z p(\mathbf{x},\mathbf{z}|\mathbf{y})d\mathbf{z},$$
- Expectation $$\mathbb{E}_{p(\mathbf{x}|\mathbf{y})}[f(\mathbf{x})] = \int_X f(\mathbf{x})p(\mathbf{x}|\mathbf{y})d\mathbf{x}$$

Most integrations are computationally intractable!

Approximate Bayesian methods

- Gaussian approximation (deterministic)
- Mean-field and variational approximation (deterministic)
- Expectation propagation (deterministic)
- Monte Carlo methods (stochastic)

1) sequential Monte Carlo (particle filter)

2) Gibbs sampling

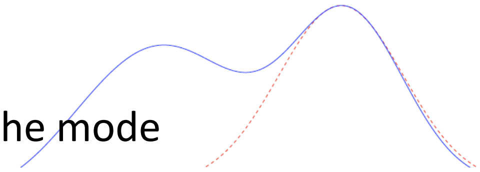
3) Markov chain Monte Carlo (MCMC)

4) Reversible jump-MCMC

Tradeoff between speed and accuracy, use of each method is often problem specific

Gaussian (Laplace) approximation

- Saddle-point method
- Taylor expansion around the mode



$$\ln f(x) = \ln f(x_0) + \underbrace{\frac{\partial \ln f(x)}{\partial x}\bigg|_{x=x_0} \cdot (x-x_0)}_{\text{second term}} + \frac{1}{2}\frac{\partial^2 \ln f(x)}{\partial x^2}\bigg|_{x=x_0} \cdot (x-x_0)^2 + h.o.t...$$

$$\ln f(x) = \ln f(x_{\text{max}}) + \frac{1}{2}\frac{\partial^2 \ln f(x)}{\partial x^2}\bigg|_{x=x_{\text{max}}} \cdot (x-x_{\text{max}})^2$$

$$\int \mathrm{e}^{\ln f(x)}\mathrm{d}x = \underbrace{\mathrm{e}^{\ln f(x_{\text{max}})}}_{\text{constant}} \int \exp\left[\underbrace{\frac{1}{2}\frac{\partial^2 \ln f(x)}{\partial x^2}\bigg|_{x=x_{\text{max}}}}_{\text{constant}} \cdot (x-x_{\text{max}})^2\right]$$

$$\approx \mathrm{e}^{L(x_{\text{max}})} \int \exp\left[-\frac{(x-x_{\text{max}})^2}{2\sigma^2}\right]\mathrm{d}x$$

First-order Laplace method

- Procedure

$$(1)\ \text{find a local maximum } x_{\max} \text{ of the given } pdf\ f(x)$$

$$(2)\ \text{calculate the variance } \sigma^2 = -\frac{1}{f''(x_{\max})}$$

$$(3)\ \text{approximate the } pdf \text{ with } p(x) \approx \mathcal{N}[x_{\max}, \sigma^2]$$

- 2nd-order Laplace method (Tierney et al., 1989): however, inherent statistical error of the posterior is often larger than the numerical error of the 1st-order approximation

Koyama et al (2010) *J Amer Statist Assoc*

An example of non-Gaussian SSM

- **State equation**: 1st-order AR process
- **Observation equation**: univariate point process

$$x(t+1) = \rho x(t) + n(t)$$

$$\log \lambda(t) = \mu + \alpha x(t) + \beta u(t)$$

observed data likelihood

$$p(Y|X,\theta) = \exp\left\{ \int_0^{T_0} \log \lambda(\tau) dy(\tau) - \int_0^{T_0} \lambda(\tau) d\tau \right\}$$

$$p(X,Y|\theta) = \left[\frac{(1-\rho^2)}{2\pi\sigma^2}\right]^{1/2} \exp\left\{ -\frac{1}{2\sigma^2}\left[(1-\rho^2)x_0^2 + \sum_{t=1}^{T_0-1}(x(t+1)-\rho x(t))^2\right]\right\} + \exp\left\{ \int_0^{T_0} \log \lambda(\tau) dy(\tau) - \int_0^{T_0} \lambda(\tau) d\tau \right\}$$

complete data likelihood

Gaussian approximation of filtered log-posterior

$$\begin{aligned}\frac{\partial^2 \log p(x_k \mid H_k)}{\partial x_k^2} = &-\frac{1}{\sigma^2_{k|k-1}} \\ &+ \sum_{c=1}^{C}\left[\left(\frac{\partial^2 \lambda_c(k\Delta)}{\partial x_k^2}\frac{1}{\lambda_c(k\Delta)} - \left(\frac{\partial \lambda_c(k\Delta)}{\partial x_k}\right)^2 \frac{1}{\lambda_c(k\Delta)^2}\right)\right. \\ &\times [dN^c(k\Delta) - \lambda_c(k\Delta)\Delta] \\ &\left. - \left(\frac{\partial \lambda_c(k\Delta)}{\partial x_k}\right)^2 \frac{1}{\lambda_c(k\Delta)}\Delta\right], \qquad \text{(A.7)}\end{aligned}$$

$$\begin{aligned}\sigma^2_{k|k} = -\Bigg[&-\frac{1}{\sigma^2_{k|k-1}} + \sum_{c=1}^{C}\left[\left(\frac{\partial^2 \lambda_c(k\Delta)}{\partial x_k^2}\frac{1}{\lambda_c(k\Delta)} - \left(\frac{\partial \lambda_c(k\Delta)}{\partial x_k}\right)^2 \frac{1}{\lambda_c(k\Delta)^2}\right)\right. \\ &\times [dN^c(k\Delta) - \lambda_c(k\Delta)\Delta] \\ &\left. - \left(\frac{\partial \lambda_c(k\Delta)}{\partial x_k}\right)^2 \frac{1}{\lambda_c(k\Delta)}\Delta\right]\Bigg]^{-1}. \qquad \text{(A.8)}\end{aligned}$$

Brown et al (1998) *J Neurosci*; Smith and Brown (2003) *Neural Computation*

Example: Point process filtering

- Discrete-analogy of Kalman filtering
- “innovation” , “Kalman gain”

$$x(t+1|t) = \rho x(t|t) \quad \text{(one-step mean prediction)}$$

$$\sigma_x^2(t+1|t) = \rho^2 \sigma_x^2(t|t) + \sigma^2 \quad \text{(one-step variance prediction)}$$

$$x(t+1|t+1) = x(t+1|t) + \sigma_x^2(t+1|t)\alpha\Big[dy(t+1) - \exp\Big(\mu + \alpha x(t+1|t+1) + \beta u(t+1)\Big)\Delta\Big] \quad \text{(posterior mode)}$$

$$\sigma_x^2(t+1|t+1) = \left[\Big(\sigma_x^2(t+1|t)\Big)^{-1} + \alpha^2 \exp\Big(\mu + \alpha x(t+1|t+1) + \beta u(t+1)\Big)\Delta\right]^{-1} \quad \text{(posterior variance)}$$

Conjugate prior

- Analytically tractable: posterior has the same form as the prior (analogous to eigenfunction in operator theory)

- Hyperpermeters viewed as pseudo-observations

- All distributions (likelihood function) from exponential family have conjugate prior (e.g., Bernoulli/binomial—beta, multinomial—Dirichlet, Poisson—gamma)

- Example: $p(\mathrm{X}) = q^{\mathrm{x}}(1\text{-}q)^{(1\text{-}\mathrm{x})}$, $f(q)$ =Beta(a,b)= $q^{\mathrm{a}-1}(1\text{-}q)^{\mathrm{b}-1}/\mathrm{B(a,b)}$, $p(q)$ = Beta($\mathrm{a+x_1+x_2\ldots+x_n}$, $\mathrm{b+n\text{-}x_1\text{-}x_2\ldots\text{-}x_n}$)

Variational approximation

- **EM→vEM→VB-EM**: Variational Bayes (VB) is an extension of EM

- **Idea**: (i) estimate the lower bound of the marginal likelihood (evidence); (ii) assume a factorial posterior form

- **VB-EM**: with conjugate priors, inference is efficient and analytically tractable

- **Issues**: bound may be loose (esp. mean field approximation), estimate may be biased, variance or uncertainty may be underestimated (Turner and Sahani, 2011)

Simple derivation

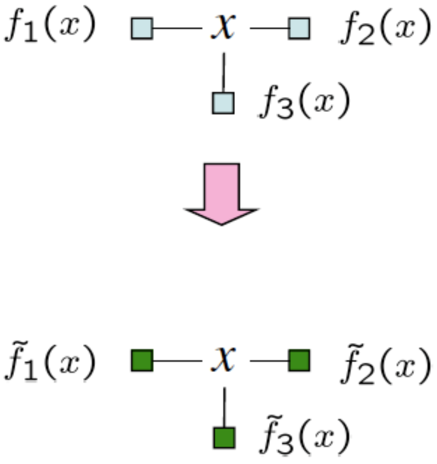
- Unobserved variables X={x_1…x_n} (parameters, or state and parameters), variational distribution Q(X)≈P(X|Y)
- Minimize KL(Q||P) = -F(Q) + log P(X) [negative free energy + evidence] or maximize F(Q) = log P(X) - KL(Q||P) ≤ log P(X)

- F(Q) = E_Q[log P(X,Y)] + *Entropy*(Q)
- Mean field approximation: Q(X) = q_1(x_1|Y)… q_n(x_n|Y)

- Minimize F(Q): iteratively update the posteriors in turn
- **Note**: similar to EM (different in M-step); minimize KL(P||Q) produces another approximation (expectation-propagation)

Expectation propagation

- Approximate each "factor" of a factored graph by a Gaussian



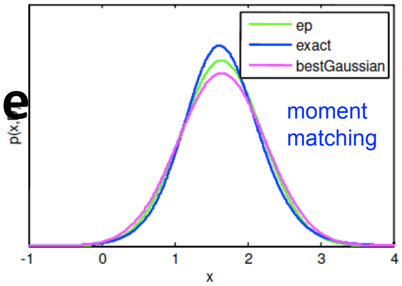
$$p(x) = \prod_a f_a(x)$$

$$f_a(x) \rightarrow \tilde{f}_a(x)$$

$$q(x) = \prod_a \tilde{f}_a(x)$$

Tom Minka (2001)

From global to local divergence



Global divergence:

$$D(p(x) \;||\; q(x)) =$$

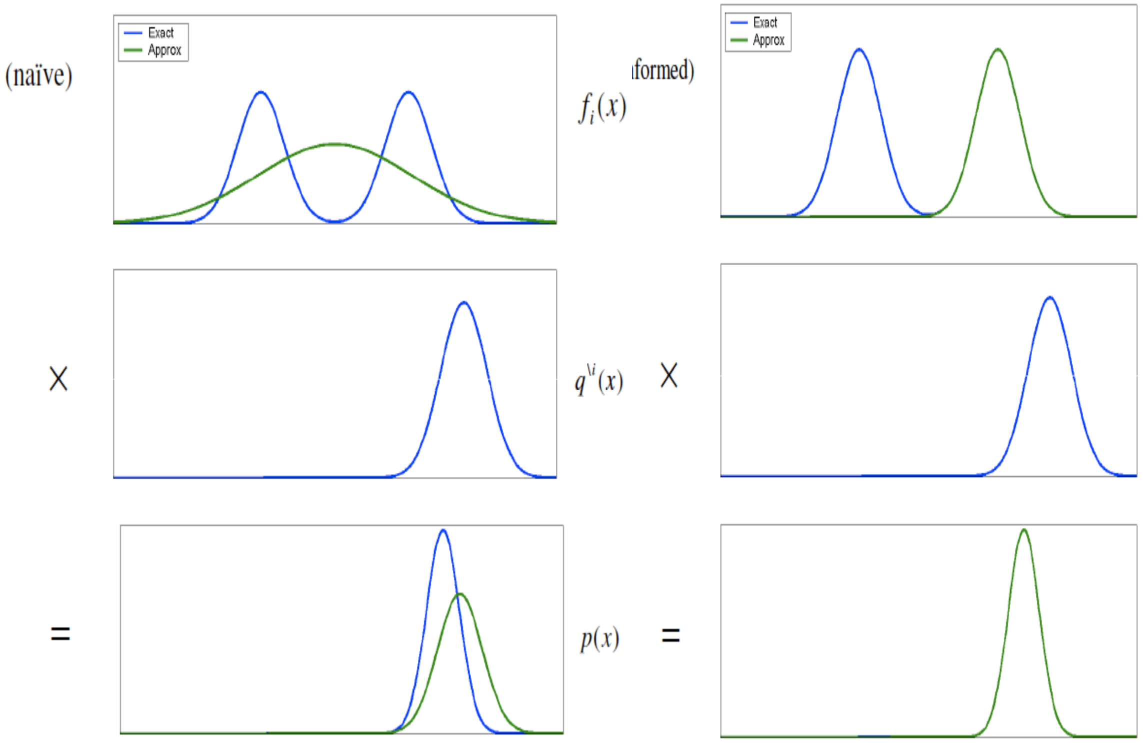
$$D(f_a(x) \prod_{b \neq a} f_b(x) \;||\; \tilde{f}_a(x) \prod_{b \neq a} \tilde{f}_b(x))$$

Local divergence:

$$D(f_a(x) \prod_{b \neq a} \tilde{f}_b(x) \;||\; \tilde{f}_a(x) \prod_{b \neq a} \tilde{f}_b(x))$$

Message passing: iteratively update the messages until convergence

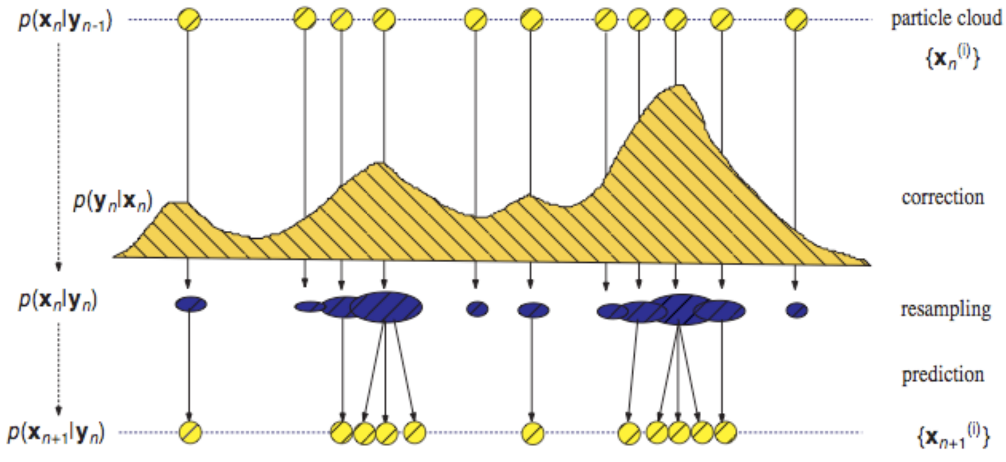
An illustration of the benefit of distributed optimization



Sequential Monte Carlo (particle filter)

- Use simulated i.i.d. samples (“particles”) to form a point mass approximation of the posterior
- Propagate the “cloud of particles” through the state and observation equations
- Importance Sampling-Resampling (“weight degeneracy”)
- Compute the posterior mean via the sample mean

Sequential importance sampling and resampling (SIR)


$p(\mathbf{x}_n|\mathbf{y}_{n-1})$
particle cloud
$\{\mathbf{x}_n^{(i)}\}$
$p(\mathbf{y}_n|\mathbf{x}_n)$
correction
$p(\mathbf{x}_n|\mathbf{y}_n)$
resampling
prediction
$p(\mathbf{x}_{n+1}|\mathbf{y}_n)$
$\{\mathbf{x}_{n+1}^{(i)}\}$

Importance sampling & importance weight update

$$\int f(\mathbf{x})p(\mathbf{x})d\mathbf{x} = \int f(\mathbf{x})\frac{p(\mathbf{x})}{q(\mathbf{x})}q(\mathbf{x})d\mathbf{x}, \qquad \hat{f} = \frac{1}{N_p}\sum_{i=1}^{N_p} W(\mathbf{x}^{(i)})f(\mathbf{x}^{(i)}),$$

$$\mathrm{Var}_q[\hat{f}] = \frac{1}{N_p}\int \left[\left(\frac{(f(\mathbf{x})p(\mathbf{x}))^2}{q(\mathbf{x})}\right)\right]d\mathbf{x} - \frac{(\mathbb{E}_p[f(\mathbf{x})])^2}{N_p}$$

$$W_n^{(i)} = W_{n-1}^{(i)}\frac{p(\mathbf{y}_n|\mathbf{x}_n^{(i)})p(\mathbf{x}_n^{(i)}|\mathbf{x}_{n-1}^{(i)})}{q(\mathbf{x}_n^{(i)}|\mathbf{x}_{0:n-1}^{(i)},\mathbf{y}_n)} \qquad W_n = W_{n-1}^{\alpha}\frac{p(\mathbf{y}_n|\mathbf{x}_n)p(\mathbf{x}_n|\mathbf{x}_{n-1})}{q(\mathbf{x}_n|\mathbf{x}_{0:n-1},\mathbf{y}_n)},$$

Q: what is the optimal proposal distribution *q*?

Choosing the proposal distribution

- Transition prior distribution (simplest)

- Use of auxiliary variables (data augmentation)

$$q(\mathbf{x}_n,\xi|\mathbf{y}_{0:n}) \propto q(\xi|\mathbf{y}_{0:n})q(\mathbf{x}_n|\xi,\mathbf{y}_{0:n}).$$

- Data-driven proposal (likelihood model, gradient or Hessian of the likelihood)

- Objective: improve the efficiency and reduce the estimate variance

Fully Bayesian estimation

- Everything unknown (state, parameters, model size or even structure) is treated as a random variable (assigned with prior)

- Assume a factorial form (mean-field approximation)

$$p(X,\theta|Y) \approx p(X|Y)p(\theta|Y) = \frac{p(Y|X,\theta)p(X)p(\theta)}{p(Y)} = \frac{p(Y|X,\theta)p(X)p(\theta)}{\int p(Y|X,\theta)p(X)p(\theta)dXd\theta}$$

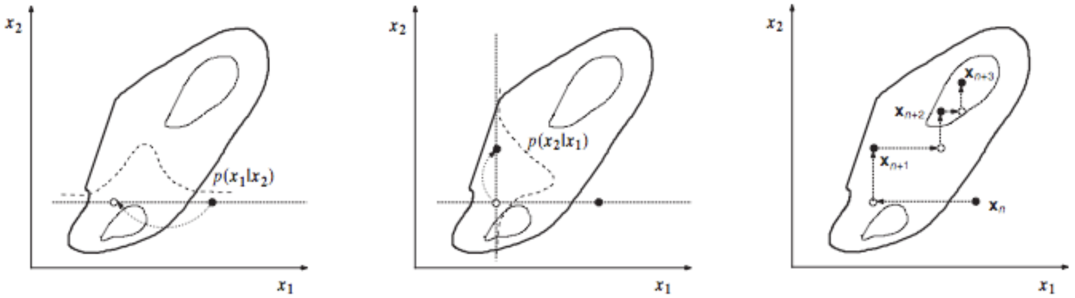
Gibbs sampling

- Joint distribution *p*(X)=*p*(x_1,x_2,…x_n)
- Conditional distribution *p*(x_i|X\x_i), conditional mean and conditional variance

- Sampling the individual conditional distributions in turn while holding others fixed

A two-dimensional example

- Non-Gaussian distribution $p(x_1, x_2)$
- Conditional Gaussian $p(x_1|x_2)$ and $p(x_2|x_1)$



MCMC

- Simulate a Markov chain to reach the equilibrium (target distribution: posterior)

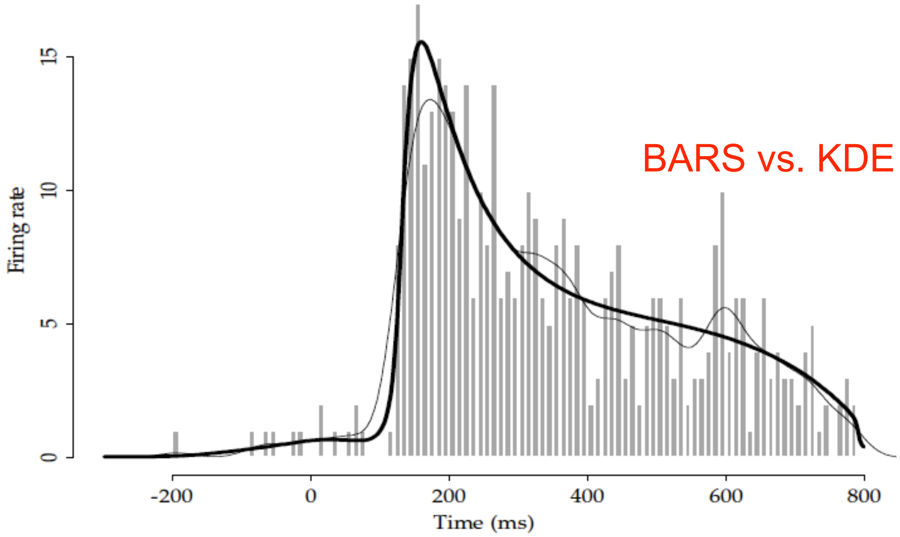
- Metropolis-Hastings, hybrid Monte Carlo, reversible jump MCMC

- Flexible, but convergence can be slow→ data-driven MCMC algorithm

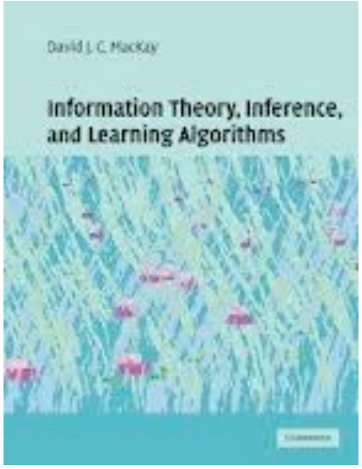
- Resources: BUGS, WINBUGS, …

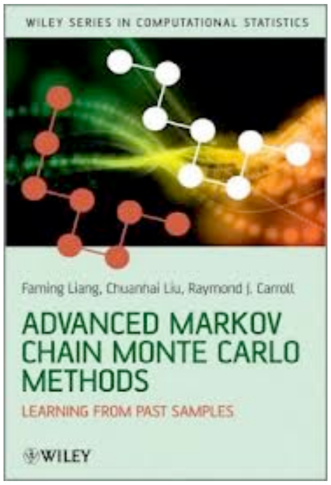
Smoothing PSTH: BARS

- How to adaptively choose bandwidth?
- Bayesian adaptive regression splines (DiMattero et al., 2001; Wallstrom et al., 2007): RJMCMC



Recommended reading




WILEY SERIES IN COMPUTATIONAL STATISTICS
Faming Liang, Chuanhai Liu, Raymond J. Carroll
ADVANCED MARKOV CHAIN MONTE CARLO METHODS
LEARNING FROM PAST SAMPLES
WILEY

Short summary

- Different levels of learning: structural, model size, parameter
- ML vs. Bayesian inference: pros and cons
- Trade-off between accuracy & computation (choose the right solution for specific problem)
- Ultimate inference goal: fast, scalable, robust

Part V: Applications in Neuroscience

Analysis of neuroscience data

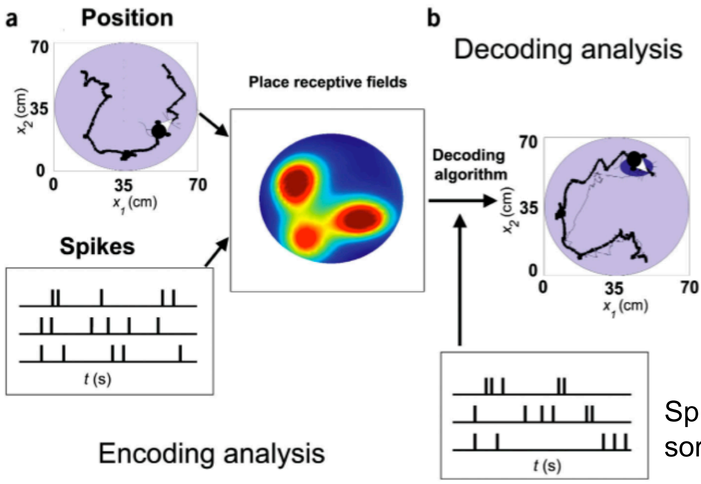
- Develop statistical techniques to characterize the dynamic features inherent in neural and behavioral responses of subjects in neuroscience experiments

- Technology advancement introduced multichannel & multiscale measurements: intracellular/extracellular, spike trains, local field potentials, EEG, ECoG, MEG, fMRI, calcium imaging, behavioral response

Representative examples

- 1) Neural decoding (rat hippocampus & primate motor cortex)
- 2) Neural plasticity
- 3) Between-trial neuronal dynamics
- 4) Behavioral analysis in learning
- 5) Neuronal rate or tuning curve estimation
- 6) MEG inverse problem
- 7) Deconvolution of fMRI time series
- 8) Optimal feedback control

1) Neural decoding: Position reconstruction from rat hippocampal ensemble spike activity



Spike sorting

Spike sorting

Encoding

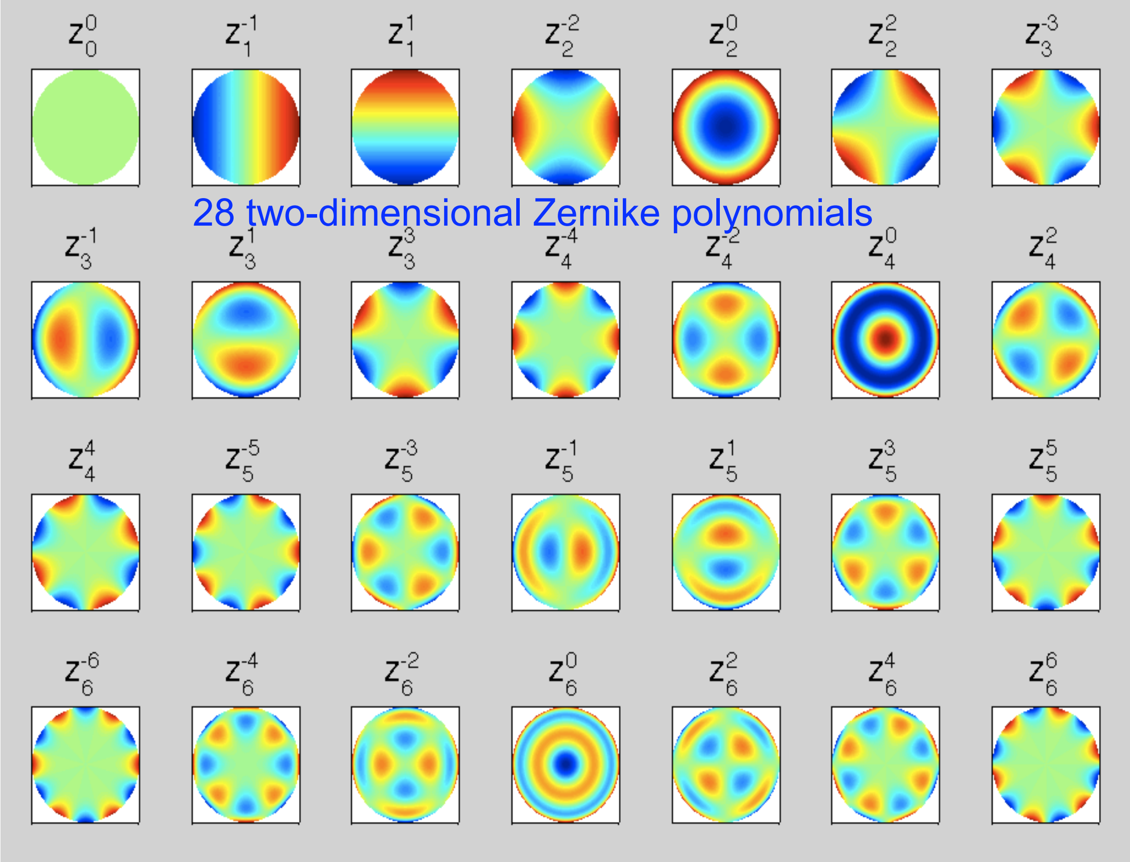
- Rat's position-path model: random walk or AR(*p*) model

$$x_k = \mu_x + F x_{k-1} + R^{\frac{1}{2}} \varepsilon_k,$$

- Modeling HPC place receptive fields: Gaussian (Brown et al., 1998) or Zernike polynomials (Barbieri et al., 2004)

$$\lambda_G^c(t \mid x(t), \xi_G^c) = \exp\left\{\alpha^c - \frac{1}{2}(x(t) - \mu^c)'(Q^c)^{-1}(x(t) - \mu^c)\right\}$$

$$\lambda_z^c(t \mid \xi_z^c) = \exp\left\{\sum_{\ell=0}^{L} \sum_{m=-\ell}^{\ell} \xi_{\ell,m}^c z_\ell^m(\rho(t), \phi(t))\right\}$$



28 two-dimensional Zernike polynomials

Point process filtering

(One-step prediction)

$$x_{k|k-1} = \mu_x + \hat{F} x_{k-1|k-1}, \tag{2.10}$$

(One-step prediction variance or learning rate)

$$W_{k|k-1} = \hat{F} W_{k-1|k-1} \hat{F}' + R\hat{W}_\varepsilon, \tag{2.11}$$

(Posterior mode)

$$x_{k|k} = x_{k|k-1} + W_{k|k-1} \sum_{c=1}^{C} \nabla \log \lambda^c(x_{k|k} \mid \hat{\xi}_j^c) \times [N_{k-1:k}^c - \lambda^c(x_{k|k} \mid \hat{\xi}_j^c)\Delta], \tag{2.12}$$

(Posterior variance)

$$W_{k|k}^{-1} = \left[W_{k|k-1}^{-1} - \sum_{c=1}^{C} [\nabla^2 \log \lambda^c(x_{k|k} \mid \hat{\xi}_j^c)[N_{k-1:k}^c - \lambda^c(x_{k|k} \mid \hat{\xi}_j^c)\Delta] - \nabla \log \lambda_c(x_{k|k} \mid \hat{\xi}_j^c)[\nabla \lambda_c(x_{k|k} \mid \hat{\xi}_j^c)\Delta]'] \right], \tag{2.13}$$

Barbieri et al. (2004) *Neural Computation*

Results

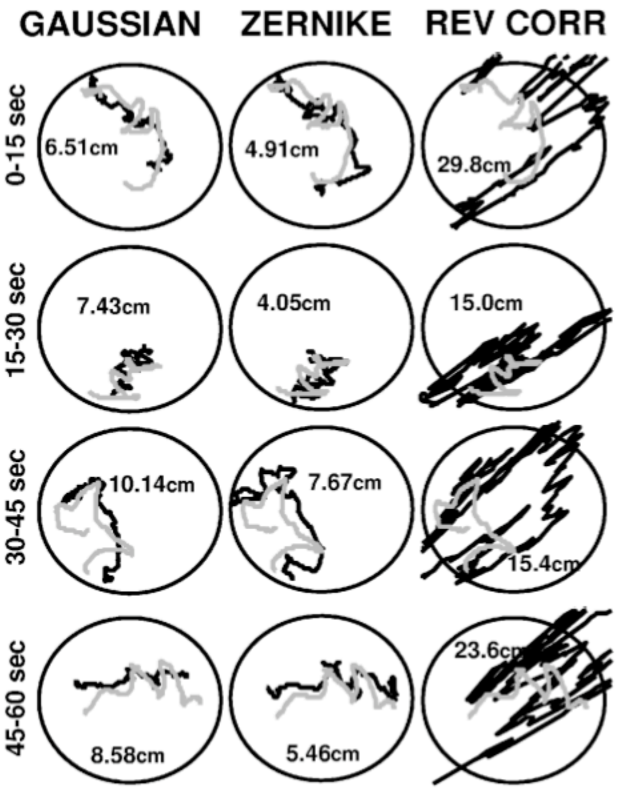
Confidence regions and coverage probabilities

$$(x_k - x_{k|k})' W_{k|k}^{-1} (x_k - x_{k|k}) \le 6,$$

6 is the 0.95 quantile of χ^2 distribution with 2 DoF

Video demo

(courtesy of Dr. Barbieri)

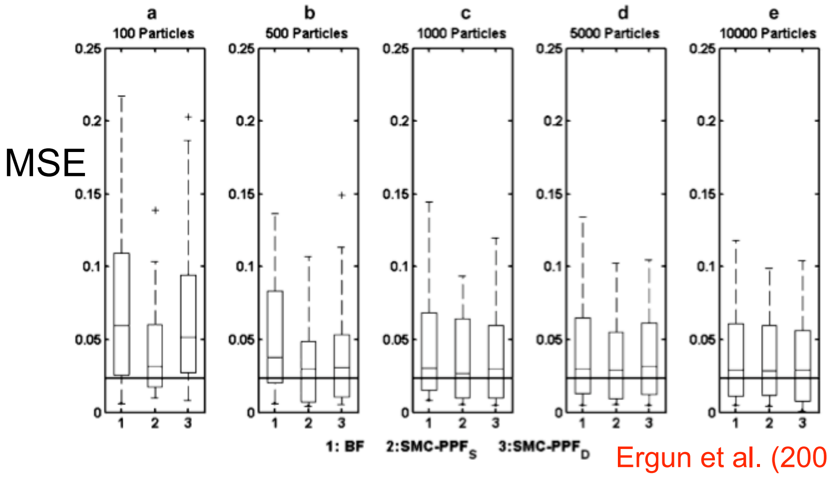


SMC + Point process filtering

- The Gaussian posterior from point process filter is used as the proposal distribution for particle filter

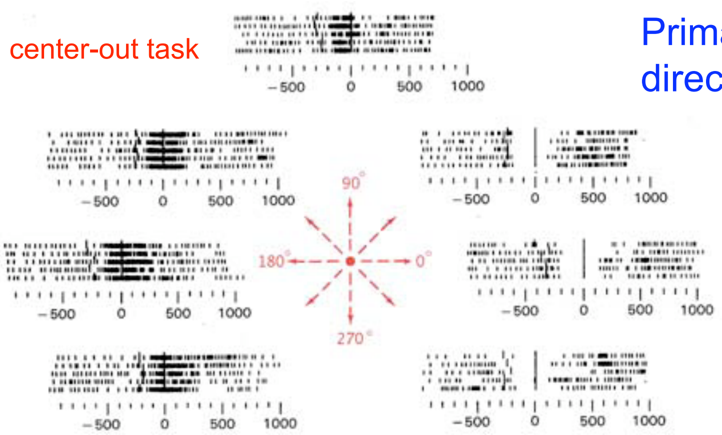
replace covariance update with learning rate

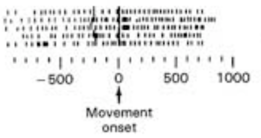
$$\psi_{k|k} = \psi_{k-1|k-1} + \varepsilon \sum_{c=1}^{C} \left(\frac{\partial \log \lambda_k^c}{\partial \psi_k} \right)' (\Delta N_k - \lambda_k^c \Delta)] \Bigg|_{\psi_{k-1|k-1}}$$



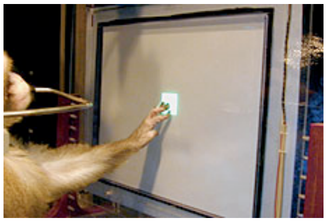
Ergun et al. (2007) *IEEE TBME*

center-out task



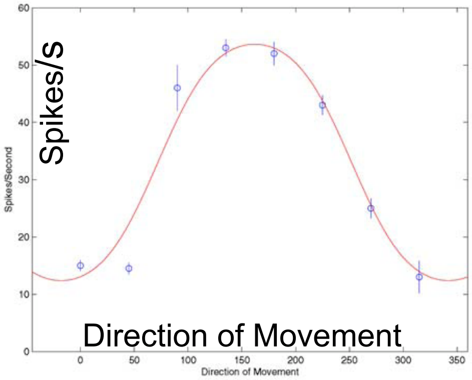


Primate M1 neuron: directional (cosine) tuning



$$K_t = \begin{bmatrix} X_t \\ V_t \end{bmatrix} = \begin{bmatrix} I_{3\times 3} & \delta I_{3\times 3} \\ 0 & \phi I_{3\times 3} \end{bmatrix} K_{t-1} + \begin{bmatrix} 0 \\ \epsilon_t \end{bmatrix}$$

Brockwell et al. (2010) *Proc. IEEE*



SSM + particle filter to decode the hand velocity from population M1 spikes

- d_i: unit-vector representing the PD (preferred direction)
- $k_i>0$, $m_i>0$

$$\lambda_i(v) = \exp(k_i + m_i v \cdot d_i + s_i \|v\|)$$

Lag: [-600, 600] ms

$$y_t^{(i)} | v_{t+\mathrm{lag}_i} \sim \mathrm{Poisson}(\lambda_i(v_{t+\mathrm{lag}_i})) \quad i = 1, 2, \ldots, N$$

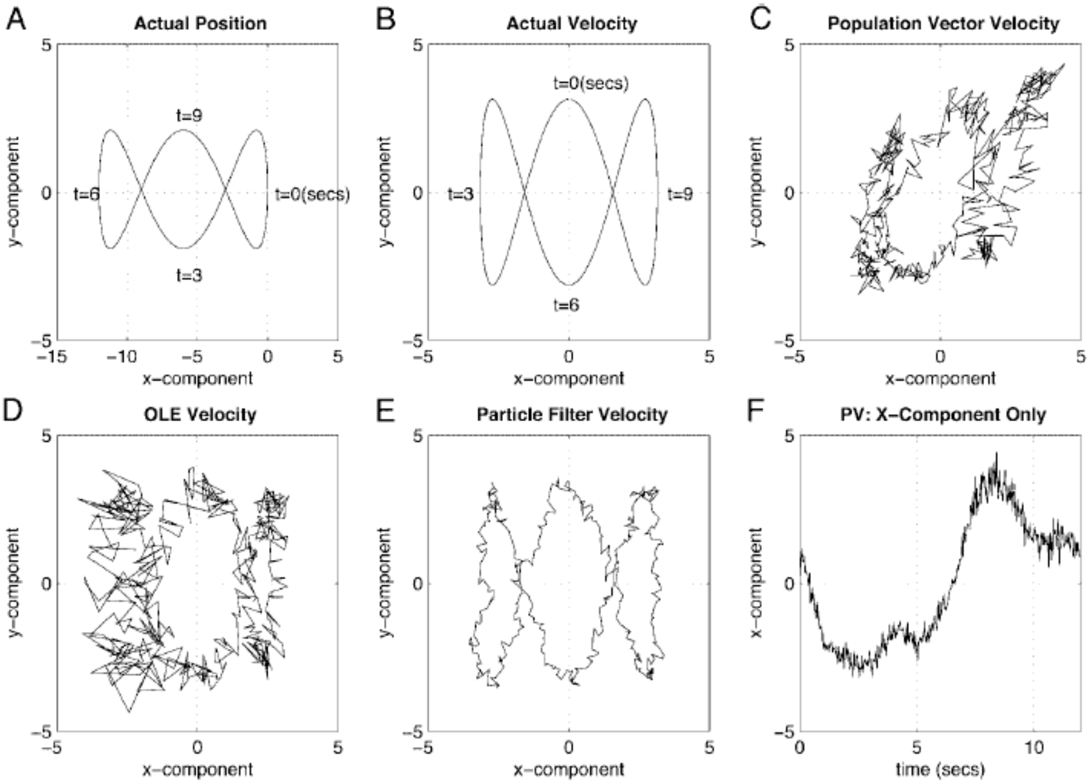
$$p(y_t|v_t) = \prod_{i=1}^{N} \frac{\exp(-\lambda_i(v_t))[\lambda_i(v_t)]^{y_t^{(i)}}}{y_t^{(i)}!}$$

$$v_t = v_{t-1} + \varepsilon_t \qquad v_{t+1} - v_t = (v_t - v_{t-1}) + \varepsilon_{t+1} \qquad v_t^* = (v_t, v_{t-1})^T,$$

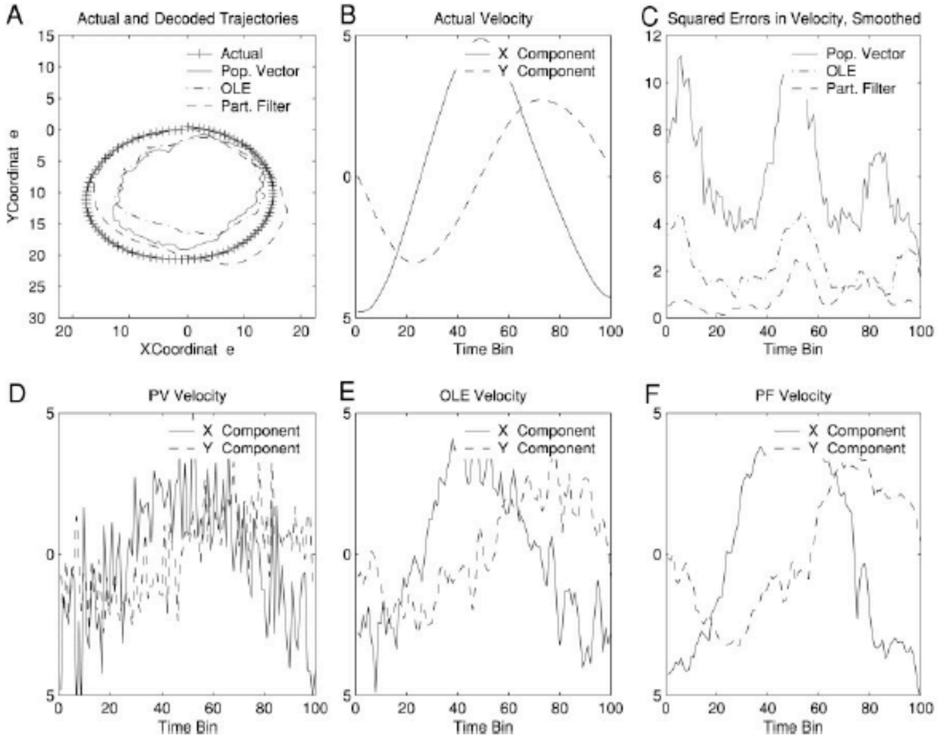
$$v_{t+1}^* = \begin{bmatrix} 2I & -I \\ I & 0 \end{bmatrix} v_t^* + \begin{bmatrix} \varepsilon_{t+1} \\ 0 \end{bmatrix}$$

Brockwell et al. (2004) *J Neurophysiol*

Simulation



Simulation



2) Neural plasticity

- Hippocampal place fields change with experiences

$$\lambda(t|\theta) = \exp\left\{\alpha - \frac{(x(t)-\mu)^2}{2\sigma^2}\right\} \qquad \theta = (\alpha, \sigma, \mu)$$

- Use of an adaptive point process filter (step size ε) to optimize the *instantaneous* log-likelihood

$$\ell_t(\theta) = \log[\lambda(t|H_t, \theta)]\frac{dN(t)}{dt} - \lambda(t|H_t, \theta).$$

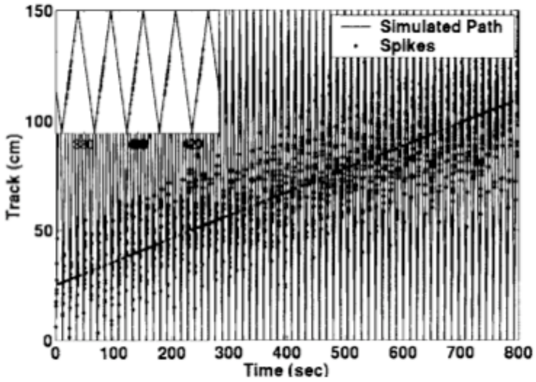
$$\hat{\theta}_k = \hat{\theta}_{k-1} - \varepsilon\frac{1}{\lambda(k\Delta|H_k, \hat{\theta}_{k-1})}\frac{\partial\lambda(k\Delta|H_k, \hat{\theta}_{k-1})}{\partial\theta}$$
$$[dN(k\Delta) - \lambda(k\Delta|H_k, \hat{\theta}_{k-1})\Delta].$$

Brown et al. (2001) *Proc Nat Acad Sci USA*

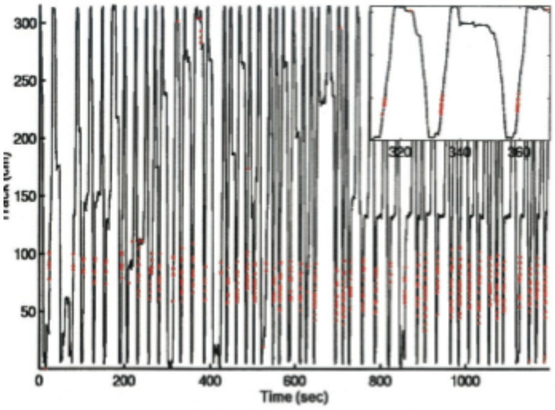
Simulation & real data

rat's hippocampus CA1 place cell

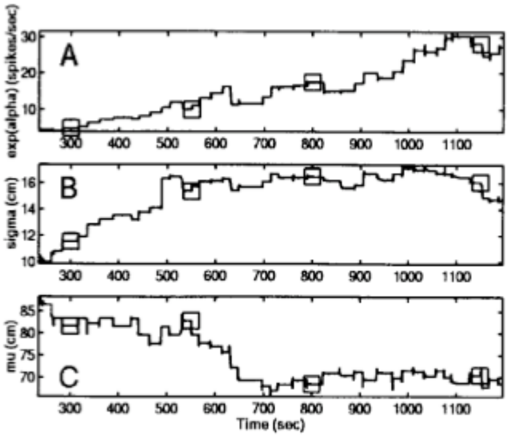
Simulated trajectory

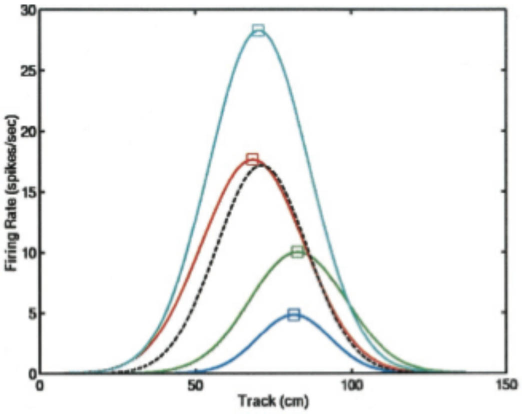


Experimental trajectory

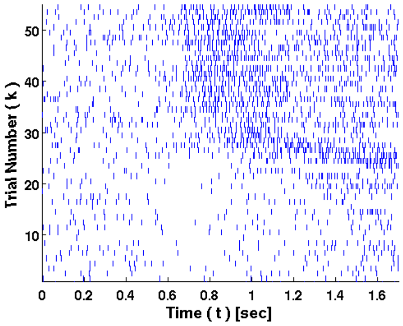


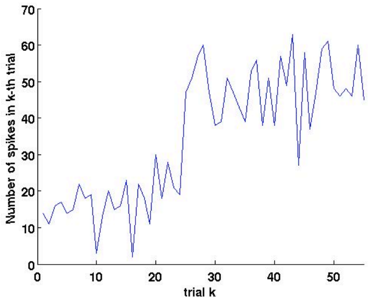
Tracking the place field in time

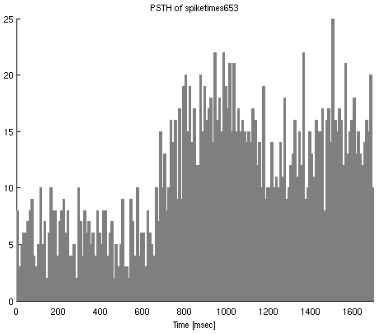




3) Between-trial neuronal dynamics







PSTH (bin size 10 ms)

- Characterize trial-by-trial changes in neuronal firing
- Neither the PSTH nor the spike count per trial is adequate.

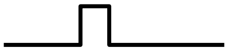
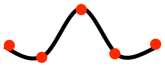
Czanner et al. (2008) *J Neurophysiol*

Neuronal spiking model

$$\lambda_k(t|\theta_k) = \exp\left\{\sum_{r=1}^{R} \theta_{k,r}\; g_r(t)\right\}$$

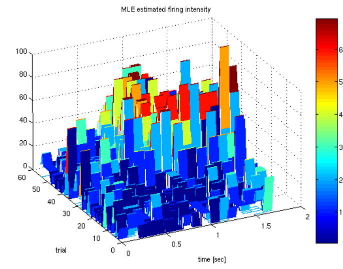
Parameter vector: $\theta_k = [\theta_{k,1}, \ldots, \theta_{k,R}]$

Basis functions: $g_r(t)$

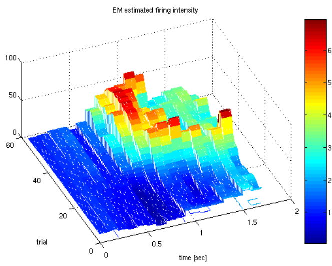
- Indicator Functions: 
- Splines: 

Assume a random-walk model (continuity) *between* trials: $\theta_k = \theta_{k-1} + \epsilon_k$

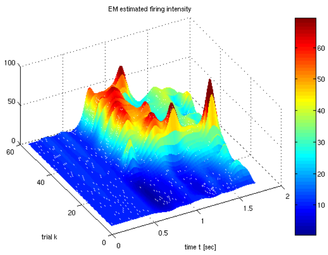
Fitting experimental data



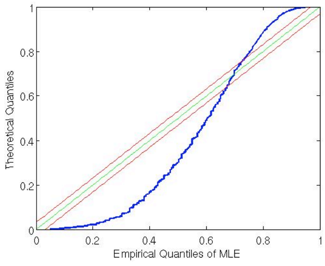
Naïve estimate of intensity

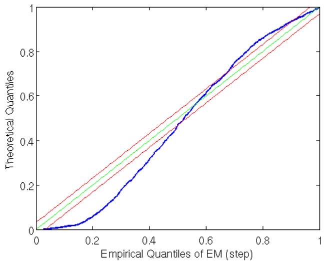


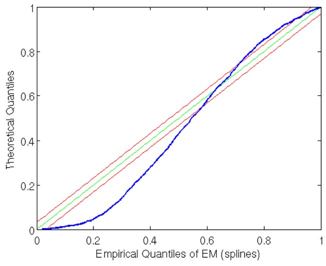
EM estimate: state-space model with step-functions



EM estimate: state-space model with splines



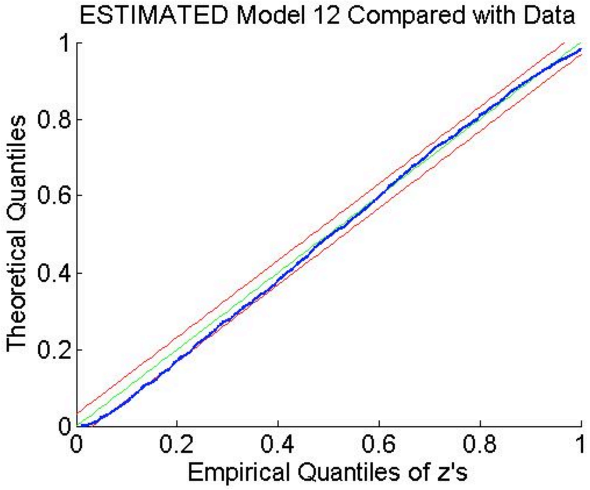




Adding spike history

$$\lambda_k = \exp\left\{\sum_{r=1}^{R}\theta_{k,r}g_r(t) + \sum_{i=0}^{9}\gamma_i(N_{k-5i} - N_{k-5i-5})\right\}$$

θ change across trials
γ constant



4) Analysis of behavioral learning

- Single cell recording in monkey hippocampus
- Trial and error learning of association between picture and response: monkeys were trained to saccade to one of four (1/4) targets, based on displayed images.

Wirth et al. (2003) *Science*

Behavioral Learning Data Model

Goal: estimate a learning curve to characterize performance as a function of trial

Question: What will be a statistical learning criterion?

Observation equation: logistic regression

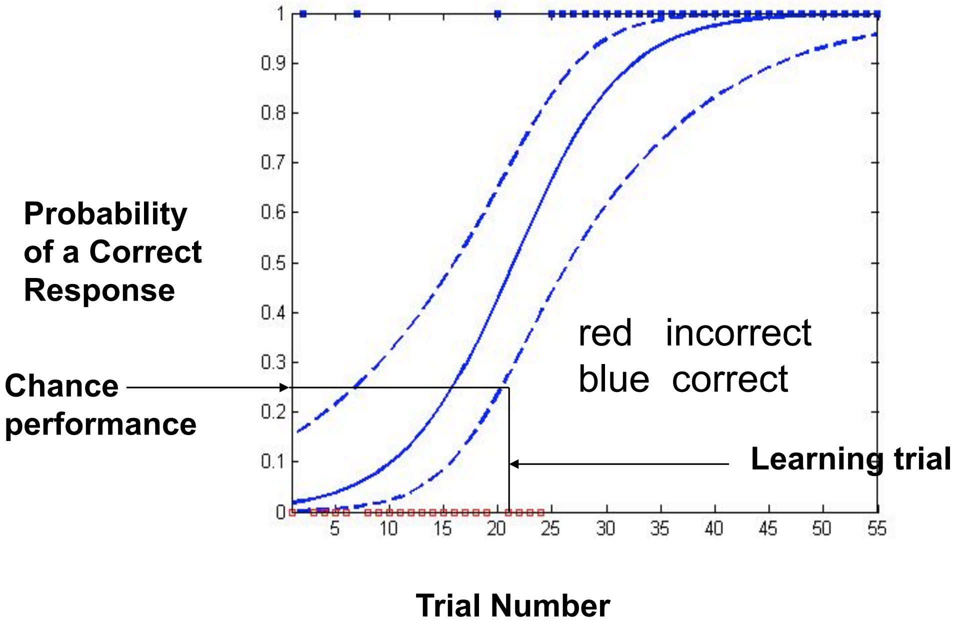
where
$$\log\left(\frac{p_k}{1-p_k}\right) = \alpha_0 + \alpha_1 k$$

k the trial number

p_k the probability of a correct response on trial *k*

α_0, α_1 the logistic regression coefficients

Static GLM

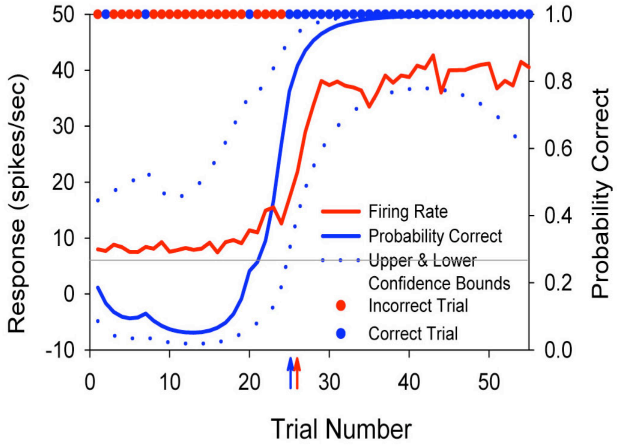


Assessment

Parameter	Estimate	*p*-value	AIC	AIC difference
const α_0	-4.224	0.0059	75	
α_1	0.1972	0.0017	36	39

Note: There is a significant improvement in performance during the experiment. Performance is better than would be expected by chance (0.25) from trial 21 onward. This behavior is consistent with the animal's task learning. Logistic regression provides an estimate of the animal's learning curve and a framework for defining the learning trial.

Dynamic SSM



Note: do not require the monotonic assumption of the logistic regression model.

Wirth et al. (2003) *Science*, Smith et al. (2004) *J Neurosci*

5) Estimation of neuronal tuning curve or firing rate function

- One-dimensional: State-space model / EM

$$x_k = x_{k-1} + \varepsilon_k,$$

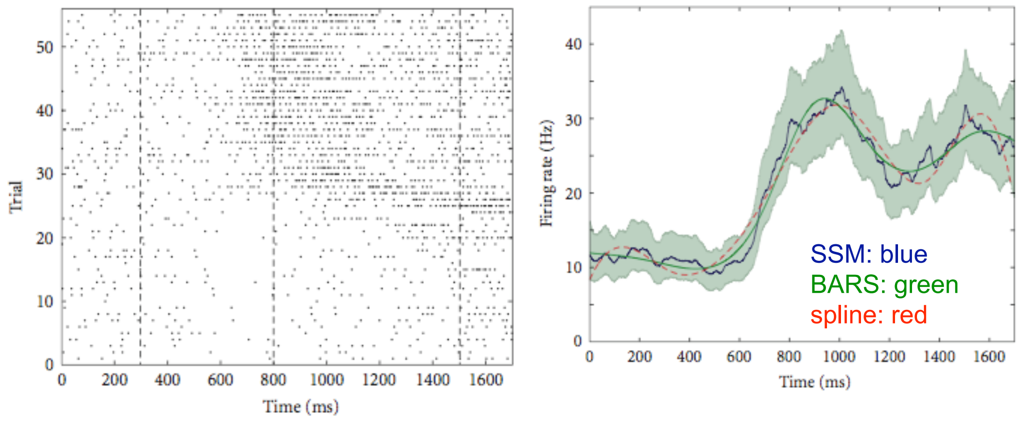
$$\Pr(n_{jk}) = \exp\{n_{jk}\log\ \lambda(k\Delta \mid x_k)\Delta - \lambda(k\Delta \mid x_k)\Delta\},$$

$$\lambda(k\Delta \mid x_k) = \exp\ (x_k). \quad p\left(\lambda_k \mid x_{k|K}, \hat{\theta}\right)$$

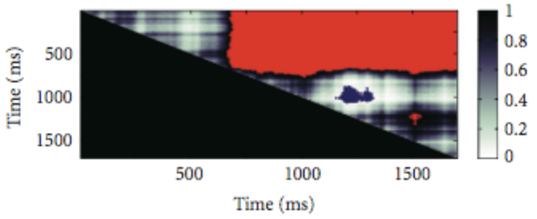
$$= (2\pi\sigma_\varepsilon^2)^{-1/2}\lambda_k{}^{-1}\exp\left\{-(2\sigma_\varepsilon^2)^{-1}(\log\lambda_k - x_{k|K})^2\right\}$$

temporal smoothing (compared to PSTH): fast and efficient (compared to BARS) and work for any temporal bin size

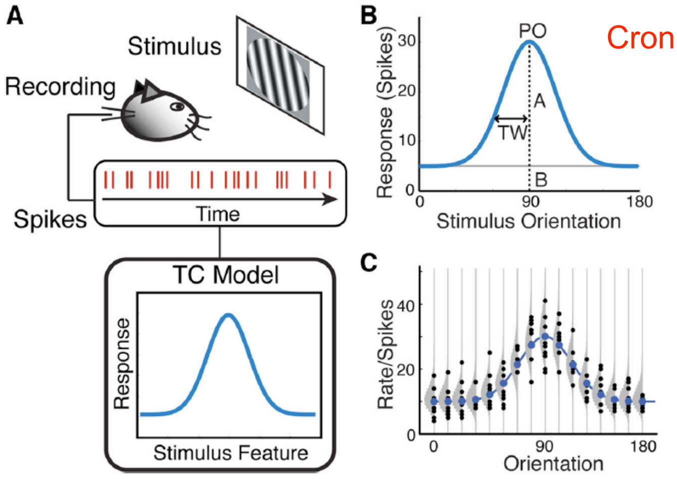
- Multi-dimensional: MCMC



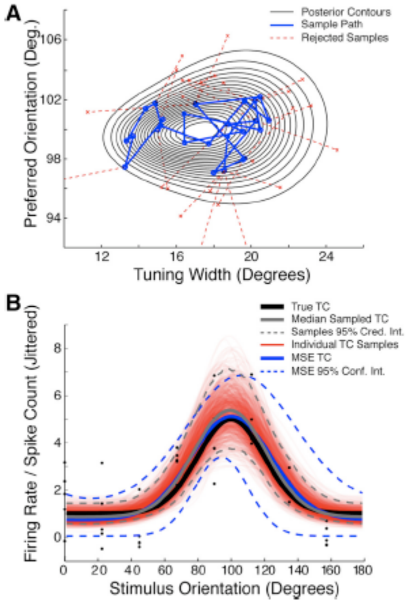
Probability of firing rate at time *i* (x-axis) > time *j* (y-axis)

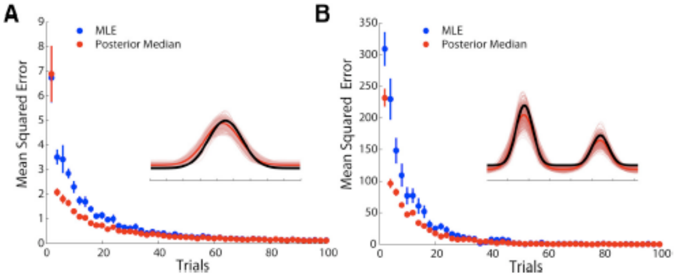


Smith et al. (2010) *Comp Intell Neurosci*



Cronin et al. (2010) *J Neurophysiol*





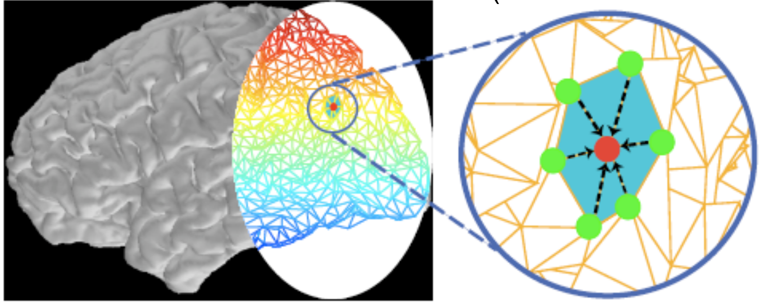
6) MEG inverse problem

- Cortical state *x* at dipole source *dim*(*x*)=N_x >5000
- Sensor *y* *dim*(*y*)=N_y

$$x_{n,t} = \lambda \left[\underbrace{a_n x_{n,t-1}}_{\text{Past activity}} + \underbrace{(1-a_n) \sum_{i\in N(n)} d_{n,i} x_{i,t-1}}_{\text{Past activity of neighbors}} \right] + \underbrace{w_{n,t}}_{\text{Unaccounted factors}} \qquad x_t = F x_{t-1} + w_t,$$

$$y_t = G x_t + v_t,$$

G: lead field gain matrix
(solution of the forward problem)



Lamus et al. (2012) *Neuroimage*

Inference

- 0<*a*<1, $d_{n,i} \propto \frac{1}{\text{distance from } n\text{th to } i\text{th dipole}},$
- *G* is estimated separately from geometrical & biophysical info
- *w*~*N*(0,*Q*), *Q*=diag{θ}, conjugate priors: inverse Gamma(*α,β*)

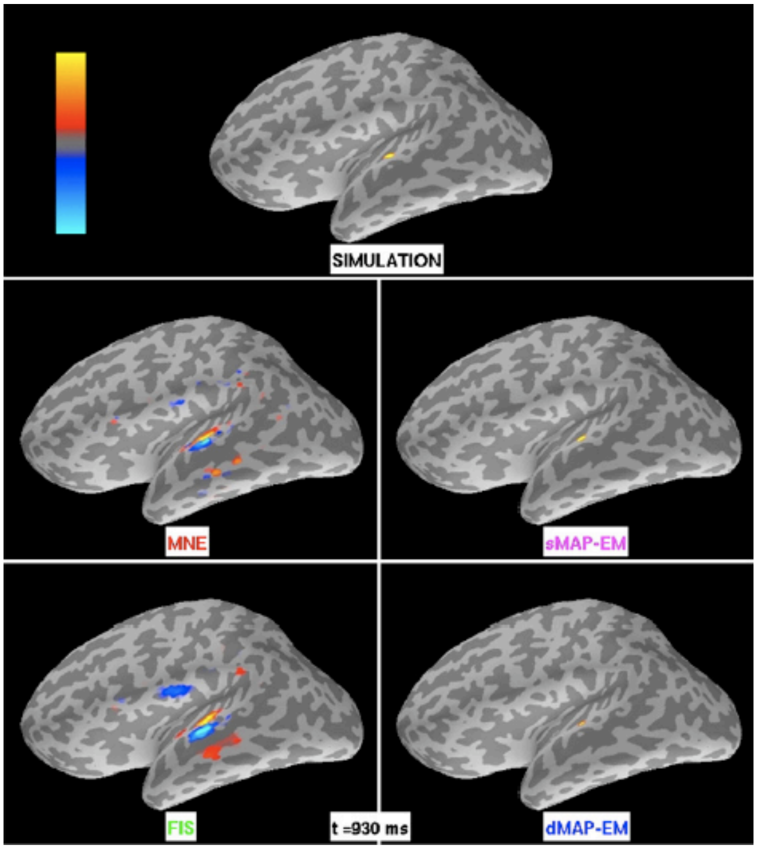
$$p(\theta) = \prod_{n=1}^{N_x} \frac{\beta^{\alpha}}{\Gamma(\alpha)} \left(\frac{1}{\theta_n}\right)^{\alpha+1} \exp\left(\frac{-\beta}{\theta_n}\right)$$

- **E-step**: fixed-interval Kalman smoother
- **M-step**: {θ} has an analytic soluion

$$\theta_n^{(r)} = \frac{\left(A^{(r)}\right)_{n,n} + 2\beta}{T + 2(\alpha+1)}, \qquad A^{(r)} \equiv A_1^{(r)} - A_2^{(r)} F' - F A_2^{(r)\prime} + F A_3^{(r)} F'$$

Results

- Video



7) Deconvolution of fMRI time series

- Bilinear dynamical system (neural activity *s*, modulatory input *u*, stimuli driving input *v*, rows of ϕ: convolution kernels, basis of hemodynamic response function)

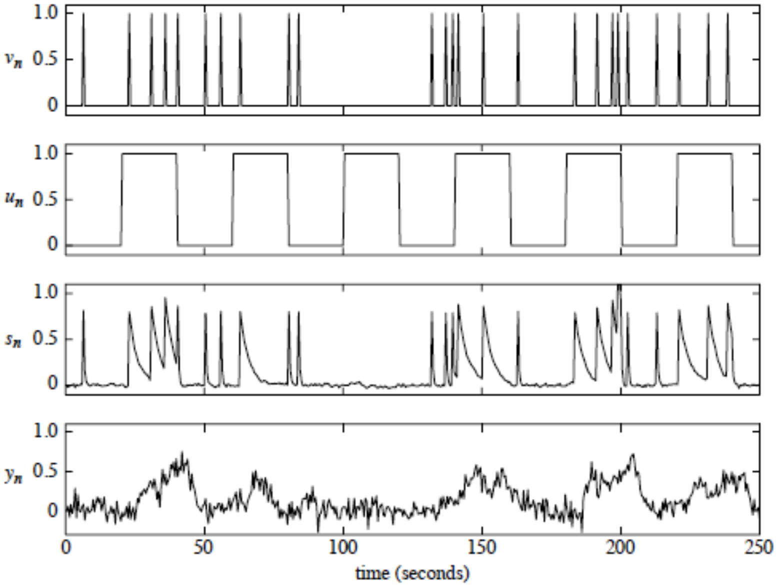
$$\begin{aligned} s_n &= \left(a + \mathbf{b}^T \boldsymbol{u}_n\right) s_{n-1} + \mathbf{d}^T \boldsymbol{v}_n + w_n \\ \boldsymbol{x}_n &= \left[s_n, s_{n-1}, s_{n-2}, .., s_{n-L+1}\right]^T \\ y_n &= \boldsymbol{\beta}^T \boldsymbol{\Phi} \boldsymbol{x}_n + e_n \end{aligned}$$

embedding

- Combination of GLM (**hemodynamics**) & stochastic dynamic casual model--DCM (**neurodynamics**)

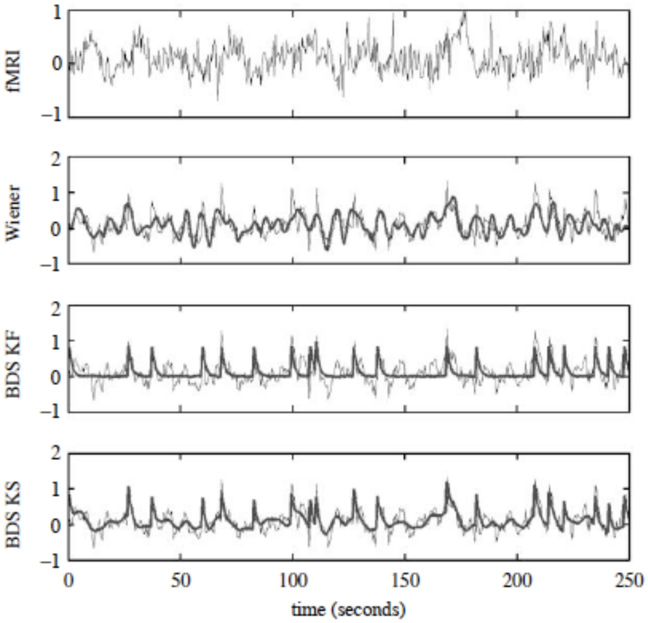
Penny et al. (2005) *Phil Trans R Soc B360*

Simulated time series



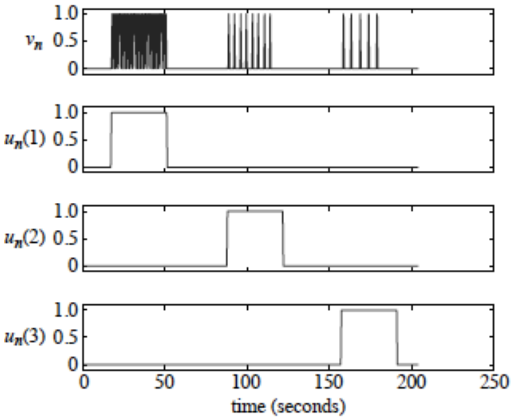
EM inference / simulation

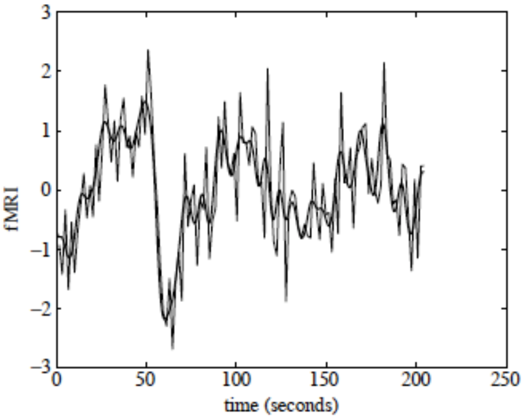
- E-step (Kalman smoothing) + M-step (***b, d, β***)



Word fMRI data

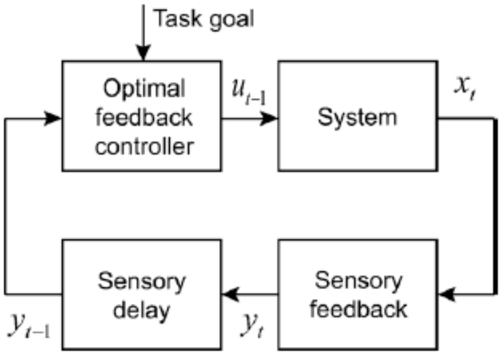
- Driving input *v*: delta function represents the presentation of words via headphone
- Modulatory input *u*: 3-dimensional step function indicates epoch with different presentation rate





8) Optimal feedback control

- Real-time BMI or neuroprosthetic application
- Sensory feedback: visual and somatosensory (tactile)



Shanechi et al. (2013) *IEEE TNSRE*

Feedback-controlled SSM

- Linear Gaussian SSM + LQG control

$$\mathbf{x}_{t+1} = \mathbf{A}\mathbf{x}_t + \mathbf{B}\mathbf{u}_t + \mathbf{w}_t . \qquad \mathbf{y}_t = \mathbf{x}_t$$

movement duration T

$$J = \sum_{t=1}^{\mathsf{T}-1} (\mathbf{x}_t' \mathbf{Q}_t \mathbf{x}_t + \mathbf{u}_t' \mathbf{R} \mathbf{u}_t) + \mathbf{x}_\mathsf{T}' \mathbf{Q}_\mathsf{T} \mathbf{x}_\mathsf{T}$$

$$\mathbf{u}_t = -\mathbf{L}_t(\mathsf{T})\mathbf{x}_t \qquad \mathbf{L}_t = (\mathbf{R} + \mathbf{B}'\mathbf{P}_{t+1}\mathbf{B})^{-1}\mathbf{B}'\mathbf{P}_{t+1}\mathbf{A}$$

$$\mathbf{P}_t = \mathbf{Q}_t + \mathbf{A}' \left(\mathbf{P}_{t+1} - \mathbf{P}_{t+1}\mathbf{B} \left(\mathbf{R} + \mathbf{B}'\mathbf{P}_{t+1}\mathbf{B} \right)^{-1} \mathbf{B}'\mathbf{P}_{t+1} \right) \mathbf{A}$$

- Optimal feedback-controlled SSM

$$\mathbf{x}_{t+1} = (\mathbf{A} - \mathbf{B}\mathbf{L}_t(\mathsf{T}))\mathbf{x}_t + \mathbf{w}_t$$

Optimal feedback-controlled prior model for a reaching movement

- Position/velocity/acceleration/force in two dimensions

$$J = \| \mathbf{d}_{\mathsf{T}} - \mathbf{d}^{*} \|^{2} + w_{v} \| \mathbf{v}_{\mathsf{T}} \|^{2} + w_{a} \| \mathbf{a}_{\mathsf{T}} \|^{2} + w_{r} \sum_{t=1}^{\mathsf{T}-1} \| \mathbf{u}_{t} \|^{2}$$

$$\mathbf{x}_t = [\mathsf{d}_1(t), \mathsf{v}_1(t), \mathsf{a}_1(t), \mathsf{d}_2(t), \mathsf{v}_2(t), \mathsf{a}_2(t)]'$$

$$\begin{bmatrix} \mathsf{d}_i(t+1) \\ \mathsf{v}_i(t+1) \\ \mathsf{a}_i(t+1) \end{bmatrix} = \begin{bmatrix} 1 & \Delta & 0 \\ 0 & 1-\frac{b\Delta}{m} & \frac{\Delta}{m} \\ 0 & 0 & 1-\frac{\Delta}{\tau} \end{bmatrix} \begin{bmatrix} \mathsf{d}_i(t) \\ \mathsf{v}_i(t) \\ \mathsf{a}_i(t) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \frac{\Delta}{\tau} \end{bmatrix} \mathsf{u}_i(t) + \begin{bmatrix} 0 \\ 0 \\ \mathsf{w}_i(t) \end{bmatrix}$$

biomechanics

b=10 Ns/m, *m* = 1kg, *τ* = 0.05 s

- Augmented state (with known desired position)

$$\begin{bmatrix} \mathsf{d}_i(t+1) \\ \mathsf{v}_i(t+1) \\ \mathsf{a}_i(t+1) \\ \mathsf{d}_i^{*} \end{bmatrix} = \begin{bmatrix} 1 & \Delta & 0 & 0 \\ 0 & 1-\frac{b\Delta}{m} & \frac{\Delta}{m} & 0 \\ 0 & 0 & 1-\frac{\Delta}{\tau} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \mathsf{d}_i(t) \\ \mathsf{v}_i(t) \\ \mathsf{a}_i(t) \\ \mathsf{d}_i^{*} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \frac{\Delta}{\tau} \\ 0 \end{bmatrix} \mathsf{u}_i(t) + \begin{bmatrix} 0 \\ 0 \\ \mathsf{w}_i(t) \\ 0 \end{bmatrix}$$

$$\mathbf{x}_{\text{aug}}(t+1) = \begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} \\ \mathbf{A}_{21} & \mathbf{A}_{22} \end{bmatrix} \mathbf{x}_{\text{aug}}(t) + \begin{bmatrix} \mathbf{B}_{11} & \mathbf{B}_{12} \\ \mathbf{B}_{21} & \mathbf{B}_{22} \end{bmatrix} \mathbf{u}(t) + \mathbf{w}(t)$$

$$p(\mathbf{x}_t | \mathbf{N}_{1:t}, \mathsf{T}) = \frac{p(\mathbf{N}_t | \mathbf{x}_t, \mathbf{N}_{1:t-1})\, p(\mathbf{x}_t | \mathbf{N}_{1:t-1}, \mathsf{T})}{p(\mathbf{N}_t | \mathbf{N}_{1:t-1}, \mathsf{T})}$$

Modified PPF in feedback control

$$\mathbf{x}_{t|t-1,\mathsf{T}} = (\mathbf{A} - \mathbf{B}\mathbf{L}_t(\mathsf{T}))\mathbf{x}_{t-1|t-1,\mathsf{T}} \tag{18}$$

$$\mathbf{W}_{t|t-1,\mathsf{T}} = (\mathbf{A} - \mathbf{B}\mathbf{L}_t(\mathsf{T}))\mathbf{W}_{t-1|t-1,\mathsf{T}}(\mathbf{A} - \mathbf{B}\mathbf{L}_t(\mathsf{T}))' + \mathbf{W} \tag{19}$$

$$\mathbf{W}^{-1}_{t|t,\mathsf{T}} = \mathbf{W}^{-1}_{t|t-1,\mathsf{T}} + \sum_{c=1}^{C} \left[\left(\frac{\partial \log \lambda_c}{\partial \mathbf{x}_t} \right)' \left(\frac{\partial \log \lambda_c}{\partial \mathbf{x}_t} \right) \lambda_c \Delta - (\mathsf{N}^c_t - \lambda_c \Delta) \frac{\partial^2 \log \lambda_c}{\partial \mathbf{x}_t \partial \mathbf{x}'_t} \right]_{\mathbf{x}_{t|t-1,\mathsf{T}}} \tag{20}$$

$$\mathbf{x}_{t|t,\mathsf{T}} = \mathbf{x}_{t|t-1,\mathsf{T}} + \mathbf{W}_{t|t,\mathsf{T}} \sum_{c=1}^{C} \left[\left(\frac{\partial \log \lambda_c}{\partial \mathbf{x}_t} \right)' (\mathsf{N}^c_t - \lambda_c \Delta) \right]_{\mathbf{x}_{t|t-1,\mathsf{T}}} \tag{21}$$

Discretization

- The arrival time T of the controlled system is unknown for external observer (vs. brain as internal observer)
- Discretize the arrival time to *J* possibilities and set a prior

$$p(\mathbf{x}_t|\mathbf{N}_{1:t}) = \sum_{j=1}^{J} p(\mathbf{x}_t|\mathbf{N}_{1:t}, T_j)p(T_j|\mathbf{N}_{1:t})$$

$$p(T_j|\mathbf{N}_{1:t}) = \frac{p(\mathbf{N}_{1:t}|T_j)p_T(T_j)}{p(\mathbf{N}_{1:t})}$$

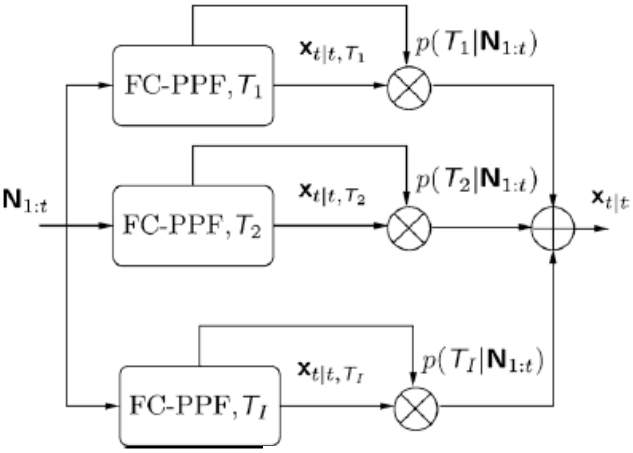
$$p(\mathbf{N}_{1:t}|T_j) = \prod_{i=1}^{t} P(\mathbf{N}_i|\mathbf{N}_{1:i-1}, T_j) = \prod_{i=1}^{t} g(\mathbf{N}_i|T_j)$$

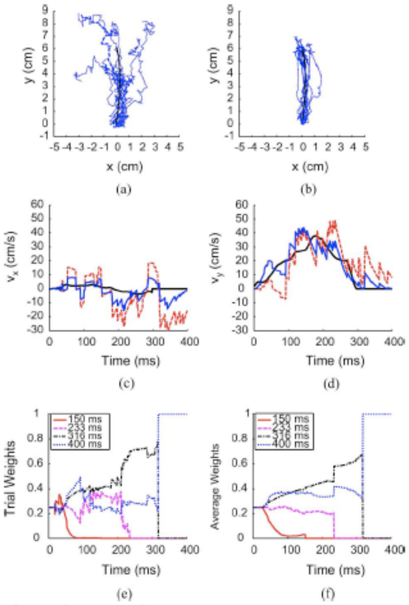
$$g(\mathbf{N}_i|T_j) = \sqrt{\frac{|\mathbf{W}_{i|i,T_j}|}{|\mathbf{W}_{i|i-1,T_j}|}}\, p(\mathbf{N}_i|\mathbf{x}_{i|i,T_j}, \mathbf{N}_{1:i-1}) \times$$

$$\exp\left[-\frac{1}{2}(\mathbf{x}_{i|i,T_j} - \mathbf{x}_{i|i-1,T_j})'\mathbf{W}^{-1}_{i|i-1,T_j}(\mathbf{x}_{i|i,T_j} - \mathbf{x}_{i|i-1,T_j})\right]$$

Feedback-controlled parallel PPF

- Finite set of arrival time {T$_j$}





Final posterior $\mathbf{x}_{t|t} = E(\mathbf{x}_t|\mathbf{N}_{1:t}) = \sum_j p(\mathsf{T}_j|\mathbf{N}_{1:t})\mathbf{x}_{t|t,\mathsf{T}_j}$

Some other neuroscience applications

- Spike sorting (Calabrese & Paninski, 2011; Herbst et al., 2008)
- Assessing higher-order neuronal synchrony (Shimazaki et al., 2012)
- Prediction of spike timing (e.g., Kobayashi & Shinomoto, 2007)
- Estimation of biophysical neuronal models from spikes (Meng et al., 2011)
- Causality analysis (EEG/fMRI multivariate time series)
- Online experimental design (e.g., Huggins & Paninski, 2011)

Take home message

- State-space analysis provides a framework for analyzing (complex) stochastic dynamical systems: well suited for dynamic nature of neural / behavioral data
- Development of stochastic models that accurately characterize the observed data & goodness-of-fit assessment
- Likelihood & fully Bayesian inference: seek efficient approximate inference for specific problems

Challenges & active research topics

- Model selection: nonparametric Bayesian methods
- Data-dependent variational bound, structured approximation
- Distributed sensors (with intermittent observations), multiscale (space and time) and multi-modal (EEG/LFP/MUA /imaging/behavioral/physiological) data fusion
- Random effects model (variability in subject/trial/session /response amplitude and latency)

Cont’

- Inference for large-scale SSM (# channels, units, time …)
- Exploit the structure of data and system (*factorial*, *sparsity, smoothness, convexity*), impose domain-specific priors for regularization, and use state-of-the-art optimization routines
- Translational neuroscience applications: BMI, DBS, seizure prediction
- Depending on the objective, neural data analysis can take different routes

  e.g., BMI vs. biased or without spike sorting

Some collective resources

- Chen Z (2003) Bayesian filtering: from Kalman filters to particle filters, and beyond. Online Tech. Rep.
- Chen Z, Brown EN (2013). State space model. *Scholarpedia* (online)
- Chen Z, Barbieri R, Brown EN (2010). State-space modeling of neural spike train and behavioral data. *Statistical Signal Processing for Neuroscience Neurotechnology* (Chap. 6), Academic Press.
- Paninski L et al. (2009) A new look at state-space models for neural data. *J Comp Neurosci*, 29:107-126.
- Durbin J, Koopman SJ (2001) *Time Series Analysis by State Space Methods*. Oxford Univ. Press.
- Cappe O, Moulines E, Ryden T (2005) *Inference in Hidden Markov Models*. Springer.
- Barber D, Cemgil, AT, Chiappa S (2011) *Bayesian Time Series Models*. Cambridge Univ. Press.
- Ozaki T (2012). *Time Series Modeling of Neuroscience Data*. Chapman & Hall/CRC.
- Paninski L, Kass RE, Eden U, Brown EN (forthcoming) *Analysis of Neural Spike Train Data*, Springer.

Forthcoming events

- Invited session “Advanced State Space Methods for Neurophysiological and Clinical Data ”, EMBC’ 13, July 3-7, Osaka, Japan (6 speakers)

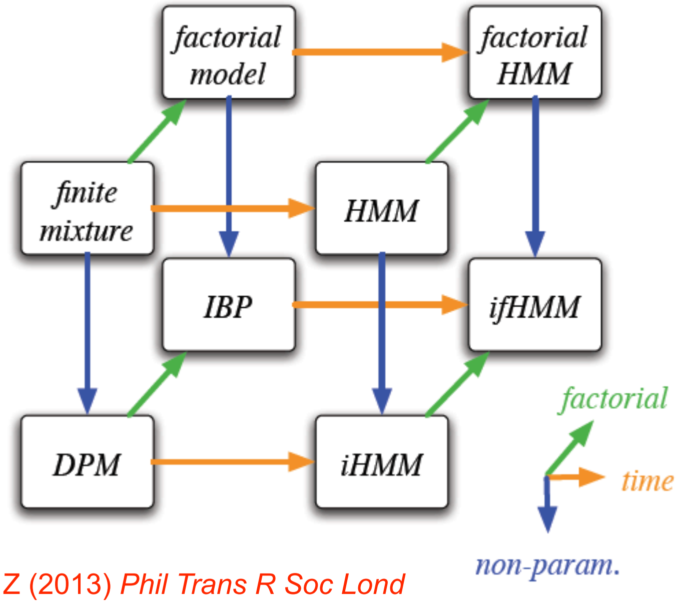
- Post-EMBC workshop in Kyoto University, July 8, 2013.

Gaussian multiplication formula

$$\begin{aligned}
\mathcal{N}(x; m_1, v_1)\mathcal{N}(x; m_2, v_2) &= \mathcal{N}(m_1; m_2, v_1 + v_2)\mathcal{N}(x; m, v) \\
\text{where } v &= \frac{1}{\frac{1}{v_1} + \frac{1}{v_2}} \\
m &= v\left(\frac{m_1}{v_1} + \frac{m_2}{v_2}\right)
\end{aligned}$$

$$\begin{aligned}
\mathcal{N}(x; m_1, v_1)/\mathcal{N}(x; m_2, v_2) &= \frac{v_2\mathcal{N}(x; m, v)}{(v_2 - v_1)\mathcal{N}(m_1; m_2, v_2 - v_1)} \\
\text{where } v &= \frac{1}{\frac{1}{v_1} - \frac{1}{v_2}} \\
m &= v\left(\frac{m_1}{v_1} - \frac{m_2}{v_2}\right)
\end{aligned}$$

HMM extension



Ghahramani Z (2013) *Phil Trans R Soc Lond*