Challenges in Analyzing Neural Spike Train Data

Rob Kass

Department of Statistics
Machine Learning Department
and
The Center for the Neural Basis of Cognition
Carnegie Mellon University

April 14, 2008
Understanding Neural Activity

Hugely diverse and interdisciplinary subject.
Understanding Neural Activity

Hugely diverse and interdisciplinary subject.

The general problem of cognitive neuroscience:

To discover how neuronal activity produces behavior.
Hugely diverse and interdisciplinary subject.

The general problem of cognitive neuroscience:

To discover how neuronal activity produces behavior.

The more specific problem of neural coding:

To elucidate “the representation and transformation of information in the nervous system.”

(Perkel and Bullock, 1968)
Hugely diverse and interdisciplinary subject.
The general problem of cognitive neuroscience:

*To discover how neuronal activity produces behavior.*

The more specific problem of neural coding:

*To elucidate “the representation and transformation of information in the nervous system.”*  
*(Perkel and Bullock, 1968)*

To begin ... some background
Challenges in Analyzing Neural Spike Train Data
How do neurons represent and transmit information?

(Adrian, 1920s; Hubel and Wiesel, 1963)
How do neurons represent and transmit information? The simplest ("textbook") answer is that a neuron responds to a relevant stimulus, or contributes to the production of an action, by increasing its firing rate. (Adrian, 1920s; Hubel and Wiesel, 1963)
In many studies, simple summaries of firing rate tell the main story.
Challenges in Analyzing Neural Spike Train Data
Source: Georgopoulos et al.
Variability and the Peri-Stimulus Time Histogram (PSTH)
Some Questions

- Which features of spike trains are “signal” and which are “noise?”

- What time scales are relevant to neural coding?

- Does the PSTH from a single neuron represent well the signal from a population (an “ensemble”) of similar neurons?

- In what ways are signals carried by populations of neurons that are not apparent from individual spike trains?
Some Questions

- Which features of spike trains are “signal” and which are “noise?”
- What time scales are relevant to neural coding?
Some Questions

- Which features of spike trains are “signal” and which are “noise?”
- What time scales are relevant to neural coding?
- Does the PSTH from a single neuron represent well the signal from a population (an “ensemble”) of similar neurons?

In what ways are signals carried by populations of neurons that are not apparent from individual spike trains?
Challenges in Analyzing Neural Spike Train Data

(A) Raw electrode recordings

(B) Local Field Potential

(C) Action Potentials
The Situation in 1998

- Single neurons
  \[ \text{Signal + Noise} = \text{Spike Count} \]
- Pairs of neurons
  \[ \text{Signal + Noise} = \text{Cross-Correlelogram} \]

However, Brown et al. (1998) \textit{J. Neuroscience} appeared. And, on a personal note, Emery Brown and I decided we should write a review article.
Single neurons
Signal + Noise = Spike Count

Pairs of neurons
Signal + Noise = Cross-Correlelogram

Miscellaneous methods
Often sophisticated but rarely principled.

However, Brown et al. (1998) appeared. And, on a personal note, Emery Brown and I decided we should write a review article.
The Situation in 1998

- Single neurons
  Signal + Noise = Spike Count
- Pairs of neurons
  Signal + Noise = Cross-Correlelogram
- Miscellaneous methods
  Often sophisticated but rarely principled.
- However, Brown et al. (1998) *J. Neuroscience* appeared.
The Situation in 1998

- **Single neurons**
  \[ \text{Signal} + \text{Noise} = \text{Spike Count} \]

- **Pairs of neurons**
  \[ \text{Signal} + \text{Noise} = \text{Cross-Correlelogram} \]

- **Miscellaneous methods**
  Often sophisticated but rarely principled.

- **However**, Brown *et al.* (1998) *J. Neuroscience* appeared. And, on a personal note, Emery Brown and I decided we should write a review article.
The Current Situation

- Important gaps in literature have been filled.

(see Kass, Ventura, Brown, 2005, J. Neurophysiology). Overall, foundations for multineuronal modeling are in place, more-or-less, but ... (will return to this at end).
The Current Situation

- Important gaps in literature have been filled.
- Value of statistical modeling emphasized (see Kass, Ventura, Brown, 2005, *J. Neurophysiology*).
The Current Situation

▶ Important gaps in literature have been filled.
▶ Value of statistical modeling emphasized (see Kass, Ventura, Brown, 2005, *J. Neurophysiology*).
▶ Overall, foundations for multineuronal modeling are in place, more-or-less, but ... (will return to this at end).
Some statistical issues, in brief

(a) How should we work with time-varying firing-rate functions?  
   (Data pooled across trials)
(b) How can we analyze non-Poisson spike trains?  
   (Data analyzed within trials)
(c) How should we assess correlation (between pairs of neurons) 
    and its stimulus dependence?
(d) How should we describe population activity?
Key Collaborators

Students/Former Students:

Sam Behseta, Can Cai, Roberto Carta, Elan Cohen, Ilaria DiMatteo, Cari Kaufman, Ryan Kelly, Jeff Liebner, Judy Xi

Postdocs:

Steve Chase, Shinsuke Koyama

Statisticians:

Anthony Brockwell, Emery Brown, Valérie Ventura, Cosma Shalizi

Neurophysiologists:

Carl Olson, Andy Schwartz, Matt Smith, Mark Stopfer, Peter Strick (and many postdoctoral associates)
“Which interval should we use for spike count comparisons across experimental conditions?”

(see fig.)
Source: Georgopoulos et al.
“Which interval should we use for spike count comparisons across experimental conditions?”

Don’t need to pick an interval: instead analyze the firing-rate functions.

Apply “functional data analysis” to noisy functions.
Aesthetically appealing: conceptually simple yet powerful approach.
References

- Behseta, Kass, Moorman, and Olson (2007, *Statistics in Medicine*)
- Olson, Gettner, Ventura, Carta, and Kass (2000, *J. Neurophysiology*)
- Ventura, Carta, Kass, Gettner, and Olson (2002, *Biostatistics*).
Data pooled across trials

Large number of trials ⇒

- Event times follow inhomogeneous Poisson process, approximately
  - $\text{ln} (t, t + dt)$, Probability of spike $= \lambda(t)dt$

Poisson-process likelihood:

$\mathbb{P}(\theta) = e^{-\int_{T_0}^{T} \lambda(u) \, du} \prod_{j=1}^\infty \lambda(t_j)$

$\lambda(t) = \lambda(t; \theta)$ may be modeled using splines or an alternative smoothing method may be used.
Data pooled across trials

Large number of trials ⇒

- event times follow inhomogeneous Poisson process, approximately
  - In \((t, t + dt)\), Probability of spike = \(\lambda(t)dt\)
  - Poisson-process likelihood:

\[
L(\theta) = e^{-\int_0^T \lambda(u)du} \prod_{j=1}^n \lambda(t_j)
\]

\(\lambda(t) = \lambda(t; \theta)\) may be modeled using splines

OR an alternative smoothing method may be used
Test $H_0 : \lambda^1(t) = \lambda^2(t)$ (BELOW: left vs. right)
\[ H_0 : \lambda_1(t) = \lambda_2(t) = \cdots = \lambda_p(t) \]
Time-Varying Firing-Rate Functions

- Apply functional data analysis to noisy functions.
Apply functional data analysis to noisy functions.
Smoothing method can be important.
Sometimes fixed-bandwidth methods are not adequate.
Our Solution:

Bayesian Adaptive Regression Splines (BARS)
Usual curve-fitting framework with “free-knot” splines

\[ Y_i = f(x_i) + \varepsilon_i \]

\( f \) is (approximated by) cubic spline with \( k \) knots at \( \xi_1, \ldots, \xi_k \)
Challenges in Analyzing Neural Spike Train Data
Usual curve-fitting framework with “free-knot” splines

\[ Y_i = f(x_i) + \varepsilon_i \]

\[ f(x) = \sum_{j=1}^{k+2} b_j(x) \beta_j \]

where \( b_j(x) \) are spline basis functions using knots \( \xi_1, \ldots, \xi_k \).

- Conditionally on \((k, \xi)\) would have a linear regression problem.

Kass

Challenges in Analyzing Neural Spike Train Data
Usual curve-fitting framework with “free-knot” splines

\[ Y_i = f(x_i) + \varepsilon_i \]

\[ f(x) = \sum_{j=1}^{k+2} b_j(x) \beta_j \]

where \( b_j(x) \) are spline basis functions using knots \( \xi_1, \ldots, \xi_k \).

- Conditionally on \( (k, \xi) \) would have a linear regression problem.
- Hard (interesting) problem is to determine \( (k, \xi) \).
- This is a “model-selection” problem.
Key Features of BARS

- Reversible-jump MCMC on \((k, \xi)\) after integrating out spline coefficients \(\beta\).
Key Features of BARS

- Reversible-jump MCMC on $(k, \xi)$ after integrating out spline coefficients $\beta$.
- Continuous proposals for $\xi$, with locality heuristic that attempts to place new knots near existing knots.
Key Features of BARS

- Reversible-jump MCMC on \((k, \xi)\) after integrating out spline coefficients \(\beta\).
- Continuous proposals for \(\xi\), with locality heuristic that attempts to place new knots near existing knots.
- Poisson case uses Laplace’s method to approximate the integral over \(\beta\).
Key Features of BARS

▶ Reversible-jump MCMC on \((k, \xi)\) after integrating out spline coefficients \(\beta\).
▶ Continuous proposals for \(\xi\), with locality heuristic that attempts to place new knots near existing knots.
▶ Poisson case uses Laplace’s method to approximate the integral over \(\beta\).
▶ We essentially use BIC to define the chain on \((k, \xi)\).

Kass and Wasserman (1995, *JASA*)

Have yet to find any example where any other method works better.

Can evaluate and propagate uncertainty easily.
Key Features of BARS

- Reversible-jump MCMC on \((k, \xi)\) after integrating out spline coefficients \(\beta\).
- Continuous proposals for \(\xi\), with locality heuristic that attempts to place new knots near existing knots.
- Poisson case uses Laplace’s method to approximate the integral over \(\beta\).
- We essentially use BIC to define the chain on \((k, \xi)\). Kass and Wasserman (1995, JASA)
- Have yet to find any example where any other method works better.
Key Features of BARS

- Reversible-jump MCMC on \((k, \xi)\) after integrating out spline coefficients \(\beta\).
- Continuous proposals for \(\xi\), with locality heuristic that attempts to place new knots near existing knots.
- Poisson case uses Laplace’s method to approximate the integral over \(\beta\).
- We essentially use BIC to define the chain on \((k, \xi)\). Kass and Wasserman (1995, *JASA*)
- Have yet to find any example where any other method works better.
- Can evaluate and propagate uncertainty easily.
BARS fit, together with Gaussian kernel density estimate, for IT neuron.
We have a good implementation (incorporating Kooperberg’s LOGSPLINE code to provide initial values).

Code is available in R and Matlab.
We have a good implementation (incorporating Kooperberg’s LOGSPLINE code to provide initial values).

Code is available in R and Matlab.

We have also developed and studied a random-effects method (hierarchical model) for analyzing a population of noisy random curves.
II. Non-Poisson Spike Trains

Data analyzed within trials

Within-Trial Analyses

Probability of spike in interval \((t, t + dt)\) becomes \(\lambda(t|H_t)dt\) where \(H_t\) is the spike train history prior to time \(t\).
Within-Trial Analyses

Probability of spike in interval \((t, t + dt)\) becomes \(\lambda(t|H_t)dt\)
where \(H_t\) is the spike train history prior to time \(t\).

Modeling starting point: \(\log \lambda(t|H_t) = f(t, H_t, x)\)
Right-hand side typically involves several additive terms
Within-Trial Analyses

Probability of spike in interval \((t, t + dt)\) becomes \(\lambda(t|H_t)dt\) where \(H_t\) is the spike train history prior to time \(t\).

Modeling starting point: \(\log \lambda(t|H_t) = f(t, H_t, x)\)

Right-hand side typically involves several additive terms

First difficulty: \(\lambda(t|H_t) = \lambda(t|s_1, s_2, \ldots, s_{n(t)})\) is function of \(n(t) + 1\) variables. Need some simplifying assumption.
Within-Trial Analyses

Probability of spike in interval \((t, t + dt)\) becomes \(\lambda(t|H_t)dt\)
where \(H_t\) is the spike train history prior to time \(t\).

Modeling starting point: \(\log \lambda(t|H_t) = f(t, H_t, x)\)

Right-hand side typically involves several additive terms

First difficulty: \(\lambda(t|H_t) = \lambda(t|s_1, s_2, \ldots, s_{n(t)})\) is function of \(n(t) + 1\) variables. Need some simplifying assumption.

A first-pass (pretty good) solution: Markov model

\[\lambda(t|H_t) = \lambda(t, s_*(t))\]

where \(s_*(t)\) is time of last spike preceding \(t\)
Goodness-of-fit plot for SEF neuron during delayed saccade task
Locust antennal lobe neuron following exposure to odor
Within-Trial Analyses

Additional issues:

▶ details of Markov modeling;
▶ trial-to-trial variability and latency;
▶ phase dependence with quasi-oscillatory local field potential;
▶ bursting;
▶ accounting for all these effects when trying to identify key coding variables.

Solutions:

▶ Each problem treated separately.
▶ One big model; fit using BARS-type model, via MCMC. (Jeff Liebner PhD thesis)
Bursting/Up-Down States

Judy Xi Ph.D. thesis

“Up” and “down” states (or bursting) may be identified in spike train data recorded extracellularly.

» Can be very hard to tell by eye.
» Are “bursts” real?
Goldfish Retinal Ganglion Data

975 spikes recorded extracellularly \textit{in vitro}

Kass

Challenges in Analyzing Neural Spike Train Data
First thought: Hidden Markov Model (HMM)

- **UP state**: Poisson process, rate \( \lambda_{UP} \)
- **DOWN state**: Poisson process, rate \( \lambda_{DOWN} \)

Markov transitions:
- **DOWN \( \rightarrow \) UP**
- **UP \( \rightarrow \) DOWN**

But probably not Poisson and, especially, probably not Markov!
Bursting/Up-Down States

First thought: Hidden Markov Model (HMM)

**UP** state: Poisson process, rate $\lambda_{UP}$

**DOWN** state: Poisson process, rate $\lambda_{DOWN}$

Markov transitions **DOWN** $\rightarrow$ **UP**

Markov transitions **UP** $\rightarrow$ **DOWN**

But probably not Poisson and, especially, probably not Markov!
Bursting/Up-Down States


UP state: possibly non-Poisson renewal process, rate $\lambda_{UP}$
DOWN state: possibly non-Poisson renewal process, rate $\lambda_{DOWN}$

Transitions DOWN $\rightarrow$ UP possibly non-Exponential waiting times

Transitions UP $\rightarrow$ DOWN possibly non-Exponential waiting times.

UP state: possibly non-Poisson renewal process, rate $\lambda_{UP}$
DOWN state: possibly non-Poisson renewal process, rate $\lambda_{DOWN}$

Transitions DOWN $\rightarrow$ UP possibly non-Exponential waiting times

Transitions UP $\rightarrow$ DOWN possibly non-Exponential waiting times.

Fit with MCMC.
Probability Density

ISI (ms)

HSMM
HMM
Inverse Gaussian

Kass
Challenges in Analyzing Neural Spike Train Data
Challenges in Analyzing Neural Spike Train Data
Challenges in Analyzing Neural Spike Train Data
Goldfish retinal data, from posterior:

- Transitions strongly non-Markov;
- Processes strongly non-Poisson.
III. Within-Trial Analyses: Trial-to-Trial Variation

Can be important.

Kass and Ventura (2005) *Neural Computation*
Spatial trial number
-200 0 200 400 600
Pattern
-200 0 200 400 600
Time (ms)
0 40 80 120
firing rate per second
-200 0 200 400 600
Time (ms)
0 40 80 120

Kass
Challenges in Analyzing Neural Spike Train Data
Challenges in Analyzing Neural Spike Train Data
Within-Trial Analyses

Trial-to-trial variation can be important.

▶ In some cases “shared excitability” model fits the data: for neuron \( i \) on trial \( r \),

\[
\log \lambda_r^i(t|H_t) = \lambda^i(t) + \text{history effects}^i + \text{shared trial effects}_r
\]

which provides a way of separating trial-to-trial correlation from within-trial correlation. Former is often dominant.
Liuxia Wang PhD thesis: firing rate for neuron \( i \) depends on

- a trial-specific “latent driver” \( \gamma \) common to all neurons in a cluster;
- a neuron-specific weight \( w \)

For neuron \( i \) in cluster \( c(i) \) on trial \( j \),

\[
\begin{align*}
\text{count}_{ij} &= \text{rate}_{ij} + \text{noise}_{ij} \\
\text{rate}_{ij} &= w_i \gamma_{c(i),j}
\end{align*}
\]
Method: MCMC; identifies number of clusters.

Simulation result: when posterior probability of cluster membership is high, false detection probability is small. (Typically can not identify clusters by viewing the correlation matrix.)

Analysis: data from monkey V1, moving bar oriented in two directions (stimulus 1, stimulus 4).

Interesting result: cluster membership depends on stimulus.
Stimulus 1

Kass Challenges in Analyzing Neural Spike Train Data
IV. Motor Cortical Decoding and Brain-Machine Interface

Koyama, Pérez-Bolde, Shalizi, Kass (*in progress*)
Challenges in Analyzing Neural Spike Train Data
Directional tuning of motor cortex neurons may be captured to create a neural prosthetic device.

- Motor cortical neurons are broadly directionally tuned; each is roughly cosine-tuned with peak at its preferred direction.
- By combining firing from hundreds of neurons, movement can be predicted reasonably well.
Challenges in Analyzing Neural Spike Train Data
Schwartz movie set-up:

- Monkey’s arms are restrained.
- Electrode array in monkey’s motor cortex sends signal to processor which controls robot arm.
- Monkey has learned that by “thinking” about moving his arm he can control the robot arm.

(Show movie)
Difficulties with Current Implementation

- Degradation in performance with non-uniform preferred directions.
- Boost in performance is desirable, especially as tasks get more difficult (e.g., grasping).
- Need a systematic approach to making improvements.
Each tuning curve may be characterized (reasonably well) by that cell’s preferred direction $\vec{D}$.

The PVA $\vec{P} = \sum w_i \vec{D}_i$ is simple and reasonably effective.

- $\vec{P}$ is predicted movement direction
- $\vec{D}_i$ is $i$-th neuron’s preferred direction
- $w_i$ scaled version of firing rate of $i$-th neuron
Bayesian Decoding

- When Gaussian, becomes Kalman filter (Kalman, 1960).
- More general framework (which applies to non-Gaussian data), often called Bayesian dynamic modeling (West and Harrison, 1997) or simply state-space modeling.
When Gaussian, becomes Kalman filter (Kalman, 1960).

More general framework (which applies to non-Gaussian data), often called Bayesian dynamic modeling (West and Harrison, 1997) or simply state-space modeling.

A general sequential MCMC scheme is “particle filtering” (PF).

In our setting approximate methods improve upon particle filtering.
Basic Idea

1. Probability model:

\[ \text{Prob}(\text{spike}) = f(\text{state}; \text{noise}) \]

where function \( f \) involves parameters that must be determined from training data.

2. \( \text{state} = \text{state}_t \) evolves over time.

3. Bayes’ theorem gives prediction of \( \text{state}_t \) using data together with previous predicted states based on

\[ \text{posterior mean} = \mathbb{E}(\text{state}_t | \text{data}_{1:t}). \]
In our simulation study, Bayesian decoding was 10 times more efficient than the plain PVA. In reconstructing hand movement from motor cortical data Bayesian decoding was 7 times more accurate than the PVA.

Other advantages: Bayesian decoding can accommodate non-uniformly distributed preferred directions, variable time lags for behavior, non-cosine tuning functions, fine resolution in time, etc.
PF vs PVA simulation; 100 neurons; strongly non-uniform preferred directions.
Performance of the particle filter ("the bootstrap filter") is well known to deteriorate as dimensionality of the state increases.
Performance of the particle filter ("the bootstrap filter") is well known to deteriorate as dimensionality of the state increases.
In practice, need very large number of particles.
Particle Filter Difficulties

Performance of the particle filter ("the bootstrap filter") is well known to deteriorate as dimensionality of the state increases.

In practice, need very large number of particles.
In our real time application, we can typically afford only a few hundred particles.
Performance of the particle filter ("the bootstrap filter") is well known to deteriorate as dimensionality of the state increases.

In practice, need very large number of particles.

In our real time application, we can typically afford only a few hundred particles.

What to do in higher-dimensional problems?
Assume $p(state_t|state_{t-1})$ comes from Gaussian time series.

(i) Approximate $p(state_{t-1}|data_{1:t-1})$ with Gaussian; then

(ii) get resulting approximate $p(state_t|data_{1:t})$ to be Gaussian.
Gaussian Approximation

Assume \( p(state_t | state_{t-1}) \) comes from Gaussian time series.

(i) Approximate \( p(state_{t-1} | data_{1:t-1}) \) with Gaussian; then

(ii) get resulting approximate \( p(state_t | data_{1:t}) \) to be Gaussian.

Use \( \hat{E}(state_t | data_{1:t}) \) based on Gaussian approximations.
Gaussian Approximation

Assume \( p(state_t|state_{t-1}) \) comes from Gaussian time series.

(i) Approximate \( p(state_{t-1}|data_{1:t-1}) \) with Gaussian; then
(ii) get resulting approximate \( p(state_t|data_{1:t}) \) to be Gaussian.

Use \( \hat{E}(state_t|data_{1:t}) \) based on Gaussian approximations.

This is a sequential version of “the posterior is approximately Gaussian,” where we use the posterior mode for state prediction.
In static case: $\theta$ replaces $state_t$, usual Gaussian approximation has

$$\hat{E} = E(\theta|data) + O\left(\frac{1}{n}\right).$$
In static case: $\theta$ replaces $state_t$, usual Gaussian approximation has

$$\hat{E} = E(\theta|data) + O\left(\frac{1}{n}\right).$$


$$\hat{E} = E(\theta|data) + O\left(\frac{1}{n^2}\right).$$
In static case: \( \theta \) replaces \( state_t \), usual Gaussian approximation has

\[
\hat{E} = E(\theta|\text{data}) + O\left(\frac{1}{n}\right).
\]


\[
\hat{E} = E(\theta|\text{data}) + O\left(\frac{1}{n^2}\right).
\]

Can we apply this method in sequential case?
Theoretical Results

1. Under suitable regularity conditions, the sequential 2nd order Laplace approximation to the predictive density at time $t$ has error $O(n^{-2})$. 
Theoretical Results

1. Under suitable regularity conditions, the sequential 2nd order Laplace approximation to the predictive density at time $t$ has error $O(n^{-2})$.

2. Under suitable regularity conditions, the sequential Laplace approximation to the predictive density is “stable” in the sense that two approximately equal versions at time $t - 1$ lead to approximately equal versions at time $t$.
Theoretical Results

1. Under suitable regularity conditions, the sequential 2nd order Laplace approximation to the predictive density at time $t$ has error $O(n^{-2})$.

2. Under suitable regularity conditions, the sequential Laplace approximation to the predictive density is “stable” in the sense that two approximately equal versions at time $t-1$ lead to approximately equal versions at time $t$.

3. Under suitable regularity conditions, 2nd-order Laplace approx. gives

$$\hat{E} = E(state_t|data_{1:t}) + O(\frac{1}{n^2}).$$
2nd-order Laplace (LGF-2) should be comparable to PF when number of particles $\approx$ dimension$^2$.
Computation Time

- 2nd-order Laplace (LGF-2) should be comparable to PF when 
  number of particles $\approx$ dimension$^2$

- For dim = 6, LGF-2 is faster than PF with 100 particles.
6-D simulation study: MISE of PF vs. LGF-1 and LGF-2
6-D simulation study: PF with 100 particles vs. LGF-1 and LGF-2.

<table>
<thead>
<tr>
<th>MISE ($\times 10^{-3}$)</th>
<th>Number of neurons</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>20</td>
</tr>
<tr>
<td><strong>LGF-2</strong></td>
<td>1.56 ± 0.46</td>
</tr>
<tr>
<td><strong>LGF-1</strong></td>
<td>9.52 ± 2.31</td>
</tr>
<tr>
<td><strong>PF</strong></td>
<td>249 ± 82.7</td>
</tr>
<tr>
<td><strong>posterior</strong></td>
<td>419 ± 30.6</td>
</tr>
<tr>
<td><strong>PVA</strong></td>
<td>1377 ± 85.2</td>
</tr>
</tbody>
</table>
Simulation: 200 neurons; PF, LGF-1, LGF-2 vs. PVA.
Current State and Future

Situation in 2008

Methods available:

- Inference based on time-varying, trial-averaged firing rate functions (FDA for noisy functions);
- Within-trial, non-Poisson spike-train modeling (applied point process modeling);
- Motor cortical decoding algorithms;
- Correlated activity for pairs of neurons.
Current State and Future

Situation in 2008

▶ Methods available:
  ▶ Inference based on time-varying, trial-averaged firing rate functions (FDA for noisy functions);
  ▶ Within-trial, non-Poisson spike-train modeling (applied point process modeling);
  ▶ Motor cortical decoding algorithms;
  ▶ Correlated activity for pairs of neurons.

▶ Some methods for analyzing correlated multineuronal activity are available but analysis of large-scale multineuronal data continues to be a serious challenge!
Neuron A dependent on time since previous neuron B spike. Liebner Ph.D. Thesis
Challenges in Analyzing Neural Spike Train Data
Neuron A dependent on time since previous neuron B spike.
Liebner Ph.D. Thesis
Conditional Intensity Modeling

- Neuron A dependent on time since previous neuron B spike. Liebner Ph.D. Thesis
Challenges in Analyzing Neural Spike Train Data
Neuron A dependent on time since previous neuron B spike.
   Liebner Ph.D.

Neuron A dependent on past spiking of many other neurons,
   via binned auto-regression.
   Okatan, Wilson, and Brown (2004, *Neural Comput.*)
Neuron A dependent on time since previous neuron B spike. Liebner Ph.D.


Neuron A dependent on past spiking of many other neurons, via integrated contribution since last spike of Neuron A. Rigat, De Gunst, and Van Pelt (2006, *Bayesian Analysis*)
Challenges in Analyzing Neural Spike Train Data
What Might Be Built on These Foundations?

consider magnitude of problem ...
Challenges in Analyzing Neural Spike Train Data
What Might Be Built on These Foundations?

Liebner’s model

$$\log \lambda^A(t|x, H_t) = \text{stimulus} + \text{trial} + \text{LFP} + \text{history}^A + \text{history}^B$$
Liebner’s model

\[ \log \lambda^A(t|x, H_t) = \text{stimulus} + \text{trial} + \text{LFP} + \text{history}^A + \text{history}^B \]

Generalization 1

\[ \log \lambda^A(t|x, H_t) = \text{stimulus} + \text{trial} + \text{LFP} + \text{history}^A + \sum_i \text{history}^i \]
What Might Be Built on These Foundations?

Generalization 2

\[ \log \lambda^A(t|x, H_t) = \text{stimulus} + \text{trial} + \text{LFP} + \text{history}^A \]
\[ + \text{stimulus-dependent burst common driver}_{c(A)} \]
Generalization 2

\[
\log \lambda^A(t|x, H_t) = \text{stimulus} + \text{trial} + \text{LFP} + \text{history}^A \\
+ \text{stimulus-dependent burst common driver}_{c(A)}
\]

Generalization 3

\[
\log \lambda^A(t|x, H_t) = \text{stimulus} + \text{trial} + \text{LFP} + \text{history}^A \\
+ \text{state-dependent common driver}_{c(A)}
\]
What Might a General Strategy Look Like?

- **Exploratory methods**
  1. Check for stimulus effects
  2. Check for bursting/oscillatory effects
  3. Check for connectivity
  4. Check for synchrony
  ...

- **Modeling**
  1. Fit probable effects
  2. Assess fit; consider others; assess fit ...
  3. Inference
  4. New experiment
  ...

Kass Challenges in Analyzing Neural Spike Train Data
Conclusion about Multineuronal Statistical Modeling

Challenges

- Scaling (part of devising general strategy).
- Lots of details!
Conclusion about Multineuronal Statistical Modeling

- **Challenges**
  - Scaling (part of devising general strategy).
  - Lots of details!
- **Prediction: within next 5-10 years**
  - Emergence of standard methods and strategies;
  - Lots of interesting results.
Cognitive Neuroscience is Multidisciplinary

- Biophysics—Computer Science—Engineering—Genetics
- Mathematics—Neurophysiology—Philosophy—Psychiatry
- Psychology—Radiology—Statistics
A Statistician’s View of Cognitive Neuroscience
A Statistician’s View of Cognitive Neuroscience

- COGNITIVE PSYCHOLOGY
  - human behavior
- NEUROPSYCH.
  - animal behavior
- GENETICS
  - altered animal behavior
- RADIOLOGY
  - imaging
- BIOPHYS.
  - algorithms
- STATISTICS
  - artificial neural networks
- NEUROBIOLOGY
  - synapses
  - systems analysis
- BIOMEDICAL ENGINEERING
  - single neurons
  - multiple neurons
  - networks of neurons
- MATHEMATICS
- NEUROPHYSIOLOGY
- PHILOSOPHY
  - synapses
- CS/AI