Network Inference from Spike Trains and Time Series

Tim Sauer

tsauer@gmu.edu

Department of Mathematical Sciences
George Mason University
Fairfax, Virginia 22030

MPIPKS Workshop on
Causality, Information Transfer, and Dynamical Networks
June 17, 2014

Collaborators

- Nathalia Peixoto, Bioengineering, GMU
- Rob Cressman, Physics, GMU
- Tyrus Berry, PostDoc, Penn State Univ.
- Franz Hamilton, Graduate Student

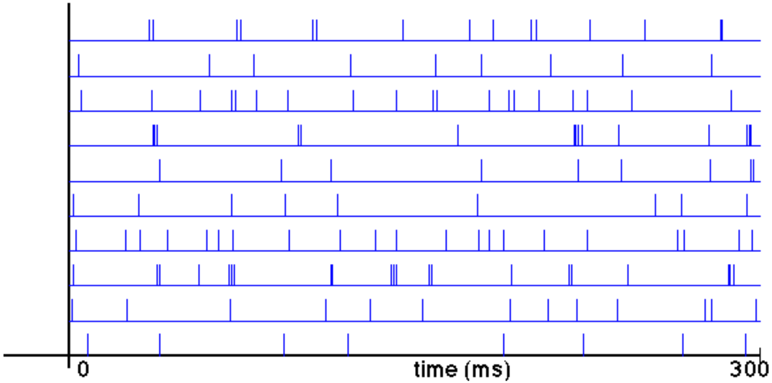
Support from:

NSF Division of Mathematical Sciences (DMS) and
Division of Civil, Mechanical and Manufacturing Innovation (CMMI).

Outline

1. Network structure from spike trains
    - Cox method from survival analysis
    - Levenberg-Marquardt
    - Statistics
2. Network structure from time series
    - Data assimilation approach
    - Simulation and results
    - Neural cultures
3. Learning unmodeled variables
    - Model error
    - Hodgkin-Huxley,
    - Potassium from neural culture

Spike trains



Network of 10 Izhikevich model neurons

History of Cox method

Survival Analysis: Study of time to failure.

Popular in: Medicine, pharma, actuarial science, insurance, financial engineering, ...

The Cox Method is maximum likelihood estimation of structure parameters of the failure pdf

We will borrow the statistical techniques from survival analysis by assigning failure = spike, and effects from other spiking neurons are described by parameters to be estimated.

History of Cox method

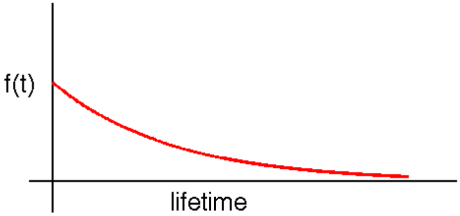
Cox (1972, 1975): Applied MLE to survival analysis

Borisyuk, Borisyuk, Kirillow, Kovalenko, Kryukov (1985): Applied Cox to spikes

Masud, Borisyuk (2011): Applied Cox to spiking networks

Berry, Hamilton, Peixoto, Sauer (2012): Minor upgrades (inc. LM)

Hazard function



pdf $f(t)$ is the probability distribution of failure times
cdf $F(t)$ = probability of failure at time $\leq t$

Hazard function, or Age-specific failure rate, is the proportional loss rate of survivors at time t:

$$\lambda(t) = -\lim_{\Delta t \to 0} \frac{1 - F(t + \Delta t) - (1 - F(t))}{(1 - F(t))\Delta t} = \frac{f(t)}{1 - F(t)}$$

If $\lambda(t) = 0.03$, then expect 3% of survivors to fail in next time unit.

Hazard function for exponential distribution

pdf $f(t)$ is the probability distribution of failure times
cdf $F(t) =$ probability of failure at time $\leq t$

Hazard function, or Age-specific failure rate, is the proportional loss rate of survivors at time t:

$$\lambda(t) = -\lim_{\Delta t \to 0} \frac{1 - F(t+\Delta t) - (1 - F(t))}{(1 - F(t))\Delta t} = \frac{f(t)}{1 - F(t)}$$

If $\lambda(t) = 0.03$, then expect 3% of survivors to fail in next time unit.

Example. (Exponential distribution.)

$f(t) = ce^{-ct}$
$F(t) = 1 - e^{-ct}$

$$\lambda(t) = \frac{f(t)}{1 - F(t)} = \frac{ce^{-ct}}{e^{-ct}} = c$$

The exponential distribution has constant hazard function.

Hazard function for spikes

pdf $f(t)$ is the probability distribution of failure times
cdf $F(t)$ = probability of failure at time $\leq t$

Hazard function, or Age-specific failure rate, is the proportional loss rate of survivors at time t:

$$\lambda(t) = -\lim_{\Delta t \to 0} \frac{1 - F(t+\Delta t) - (1 - F(t))}{(1 - F(t))\Delta t} = \frac{f(t)}{1 - F(t)}$$

If $\lambda(t) = 0.03$, then expect 3% of survivors to fail in next time unit.

Example. (Spike train.) For a given target neuron,

$$\begin{aligned} \lambda(t) &= \frac{\Pr\left(\mathrm{ISI} < t + \Delta t \mid \mathrm{ISI} > t\right)}{\Delta t} \\ &\approx \frac{\#\ \mathrm{ISI}\ \ \mathrm{lengths}\ \ \mathrm{in}\ \ [t, t+\Delta t]}{\#\ \mathrm{ISI}\ \ \mathrm{lengths}\ \ \mathrm{in}\ \ [t, \infty) \cdot \Delta t} \end{aligned}$$

Cox method for spikes

Let $\beta = (\beta_1, \ldots, \beta_n)$ denote the connections weights of n possible sources.

Assume the hazard function of the target neuron has form $\lambda(t) = \lambda_0(t) e^{\beta^T z_t}$ where $z_t = (z_{t1}, \ldots, z_{tn})$ are influences at time t of other "source" neurons.

Then calculate β by maximum likelihood estimation.

$$L = \prod_{i=1}^{k} \Pr(\tau_i) = \prod_{i=1}^{k} \frac{\lambda(\tau_i)}{\sum\limits_{\tau_j \geq \tau_i} \lambda(\tau_j)} = \frac{e^{\beta^T z_{t_1} + \ldots + \beta^T z_{t_k}}}{\prod_{i=1}^{k} \sum\limits_{\tau_j \geq \tau_i} e^{\beta^T z_{t_j}}}$$

where $\tau_1, \ldots, \tau_k$ are ISI's of the target.

To maximize log likelihood
$\mathcal{L} = \log L = \beta^T (z_{t_1} + \ldots + z_{t_k}) - \sum_{i=1}^{k} \log \sum\limits_{\tau_j \geq \tau_i} e^{\beta^T z_{t_j}}$
we solve $\nabla \mathcal{L} = 0$ for β.

Cox method for spikes – MLE

To maximize $\mathcal{L}$, we set $\nabla\mathcal{L} = 0$.

$$0 = \frac{\partial\mathcal{L}}{\partial\beta_s} = \sum_{i=1}^{k} z_{si} - \sum_{i=1}^{k} \frac{\sum_{\tau_j \geq \tau_i} z_{sj} e^{\beta^T z_{t_j}}}{\sum_{\tau_j \geq \tau_i} e^{\beta^T z_{t_j}}}.$$

Here $z_{t_j} = [e^{t_i - t_{s_1}}, \ldots, e^{t_i - t_{s_n}}]$, where t_i is target spike time and t_{s_p} is most recent spike of source node s_p. Other choices are possible, including making use of more previous spikes.

The second derivative is $H = \dfrac{\partial^2\mathcal{L}}{\partial\beta_{s_1}\partial\beta_{s_2}}$.

Use Newton iteration to solve $0 = \nabla\mathcal{L}$.

$$\beta^{m+1} = \beta^m - H(\beta^m)^{-1}\nabla\mathcal{L}(\beta^m)$$

Cox method for spikes – using Levenberg-Marquardt

To maximize $\mathcal{L}$, one can solve $\nabla\mathcal{L} = 0$ by Newton iteration

$$\beta^{m+1} = \beta^m - H(\beta^m)^{-1}\nabla\mathcal{L}(\beta^m).$$

However, much better to use Levenberg-Marquardt to regularize the iteration:

$$\beta^{m+1} = \beta^m - [(H^m)^T H^m + \mu_m \,\mathrm{diag}\,((H^m)^T H^m)]^{-1}(H^m)^T\nabla\mathcal{L}(\beta^m)$$

where $H^m \equiv H(\beta^m)$.

The scalar μ is chosen according to:

$\mu_0 = 1$ and for $m > 0$,

$\mu_m = \mu_{m-1}/2$ if $||\nabla\mathcal{L}||$ decreases;
$\mu_m = 10 \cdot \mu_{m-1}$ if $||\nabla\mathcal{L}||$ increases.

Statistics of Cox method

$I(\beta) \equiv -H(\beta)$ is the approximate Fisher information matrix.

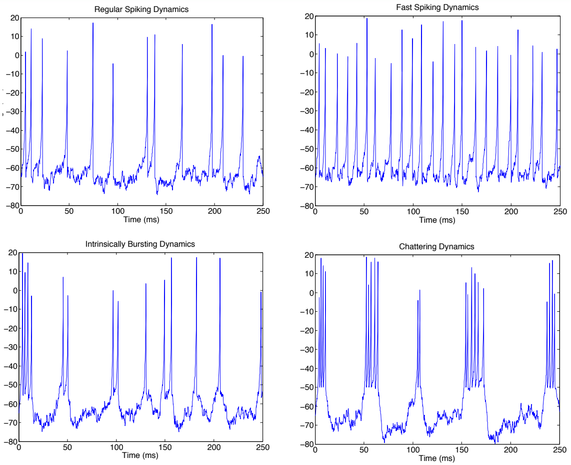
The inverse of $I(\beta)$ converges to the covariance matrix of β. We can use the diagonal entries σ_{ss} of $I(\beta)$ as variances of β for the purpose of confidence intervals, e.g.

$$[\beta_s - \sigma_{ss} N((1-\gamma)/2), \beta_s + \sigma_{ss} N((1-\gamma)/2)]$$

for the γ% quantile. Here N denotes the standard normal cdf.

We declare a link between the target and source s with γ% confidence level $\iff$ 0 is not contained in the confidence interval of β_s.

Izhikevich neurons



$$
\begin{aligned}
\dot{u} &= 5u - p_3 a + p_2 u^2/10 + I \\
\dot{a} &= 0.03(p_1 u - a)
\end{aligned}
$$

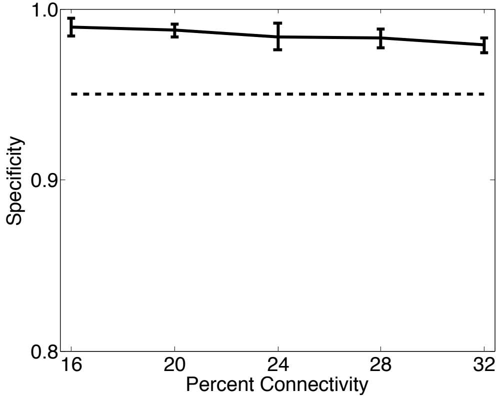
$p_1 = 0.4$

$p_2 = 0.2$

$p_3 = 1.0$

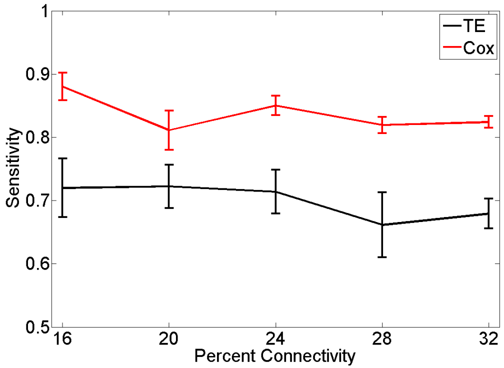
when u exceeds threshold Θ, then $u \to c$ and $a \to a + d$.

Statistics of Cox Method



Running Cox at 95% confidence level, 500 spikes per node.
10-node heterogeneous Izhikevich network.
Specificity = TN/(TN+FP) should be 95%

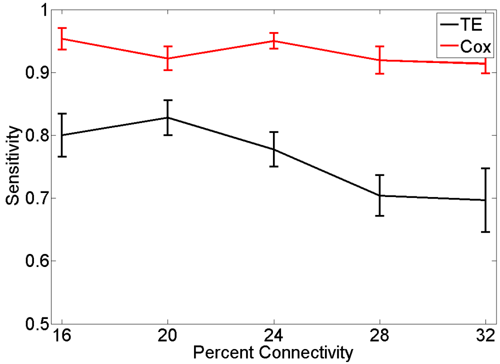
Comparison on Izhikevich networks



Comparison of Cox with Transfer Entropy order 3, using 500 spikes per node.

10-node heterogeneous Izhikevich network.
Sensitivity = TP/(TP+FN)

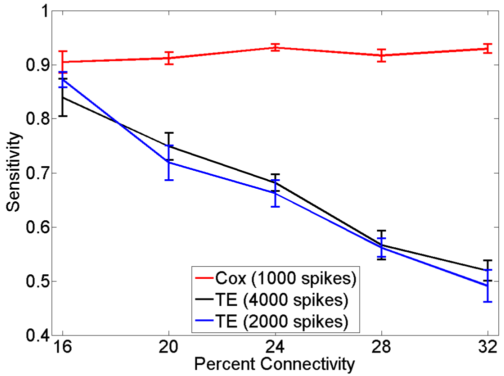
Comparison on Izhikevich networks



Comparison of Cox with Transfer Entropy order 3, using 1000 spikes per node.

10-node heterogeneous Izhikevich network.
Sensitivity = TP/(TP+FN)

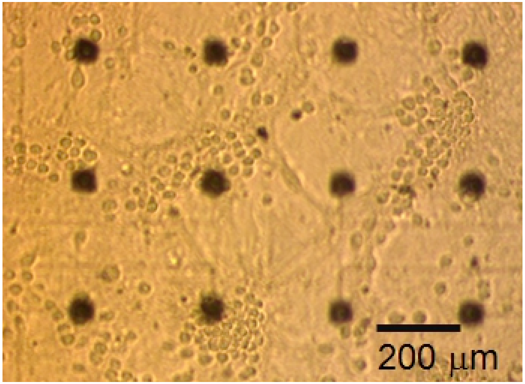
Comparison on Izhikevich networks



Comparison of Cox with Transfer Entropy order 3.

20-node heterogeneous Izhikevich network.
Sensitivity = TP/(TP+FN)

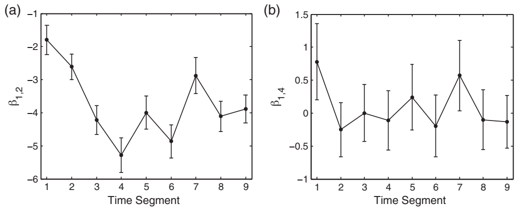
Neural cultures



Dissociated neurons deposited on Multiple Electrode Array.

After a few weeks they grow connections and spiking activity near electrodes can be monitored.

Neural cultures



Cultured neurons grown on MEA

95% confidence intervals on 9 successive two-minute time intervals. When the confidence intervals avoid 0, we say there is a connection with 95% confidence.

Advantages of Cox method

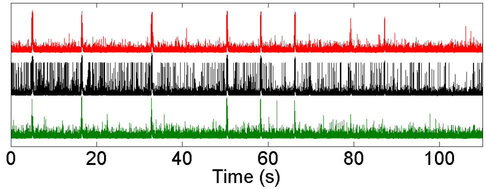
1. Although “semi-parametric”, the promised FP rate appears to hold for typical neuron models.

2. The TP rate compares favorably with alternative methods.

3. It is “statistical”, meaning that a choice of threshold from ROC curve is not necessary.

-1. It is an offline method.

Network structure from time series



Many methods exist.

We looked for methods that were “statistical”, i.e. where user choice of ROC threshold is not needed.

One idea: Use ability of data assimilation techniques to fit parameters (consider connection strengths of network to be parameters).

In addition to being statistical, this method is realtime and adaptive, unlike Cox.

Kalman Filter

Assume a model

$$\begin{aligned}\dot{x} &= f(x,p) + \omega_t \\ y &= h(x) + \nu_t.\end{aligned}$$

Observe y_t^o.

Possible goals:

1. Reconstruct state x from noisy observations
2. Reconstruct state x from noisy, incomplete observations

For example,

$$h(x) = h(x_1, \ldots, x_n) = x_1$$

Want to reconstruct $x_2(t), \ldots, x_n(t)$ as well as $x_1(t)$.

Ensemble Kalman Filter

Kalman Filter

Model state and state covariances, assuming Gaussian noise.

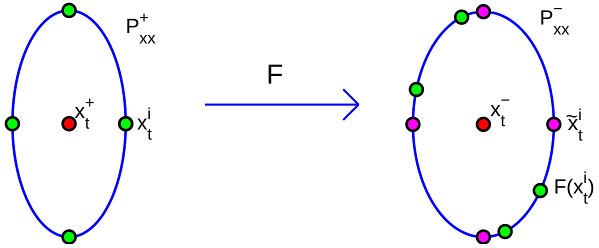
On each assimilation step, use observations to solve for maximum likelihood solution of current x_t and covariance matrix Σ_t.

Ensemble Kalman Filter (EnKF)

At each assimilation step, use current state x_t and covariance matrix Σ_t to generate a set of states, which are run in parallel according to system model f.

Results are combined to find maximum likelihood x_{t+1} and Σ_{t+1}.

Ensemble Kalman Filter (EnKF)



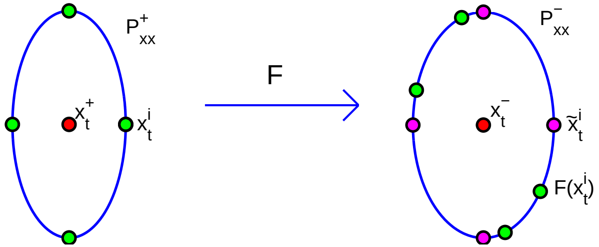
$$
\begin{aligned}
x_t^- &= \langle F(x_t^i) \rangle \\
P_{xx}^- &= \frac{1}{2n-1} \sum (F(x_t^i) - x_t^-)(F(x_t^i) - x_t^-)^T + Q
\end{aligned}
$$

where Q is dynamical noise covariance.

$$\tilde{x}_t^i = \text{“sigma points” on semimajor axes}$$

(use Cholesky or matrix square root)

Ensemble Kalman Filter (EnKF)



Calculate $\tilde{y}_t^i = H(\tilde{x}_t^i)$. Set $\hat{y}_t = \langle \tilde{y}_t^i \rangle$.

$$\begin{aligned}
P_{yy} &= \frac{1}{2n-1} \sum (\tilde{y}_t^i - \hat{y}_t)(\tilde{y}_t^i - \hat{y}_t)^T + R \\
P_{xy} &= \frac{1}{2n-1} \sum (\tilde{x}_t^i - x_t^-)(\tilde{y}_t^i - \hat{y}_t)^T \\
K &= P_{xy} P_{yy}^{-1} \text{ and } P_{xx}^+ = P_{xx}^- - K P_{yy} K^T \\
x_{t+1}^+ &= x_t^- + K(y_t - \hat{y}_t)
\end{aligned}$$

Kalman Filter

Assume a model

$$\begin{aligned} \dot{x} &= f(x,p) + \omega_t \\ y &= h(x) + \nu_t. \end{aligned}$$

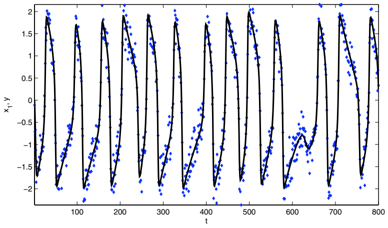
Observe y_t^o.

Extra trick:

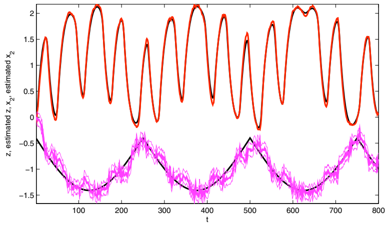
Can view parameters p as fixed variables, and ask the filter to estimate them.

$$\begin{aligned} \dot{x}_1 &= f_1(x) \\ \dot{x}_2 &= f_2(x) \\ \dot{x}_3 &= f_3(x) \\ \dot{p}_1 &= 0 \\ \dot{p}_2 &= 0 \end{aligned}$$

Ensemble Kalman Filter (EnKF) for neurons



(a)



(b)

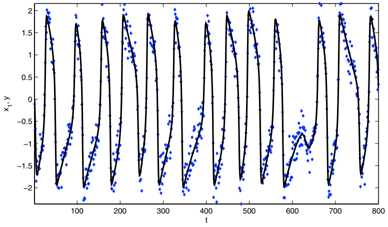
Voss, Timmer, Kurths (2004)

Spiking Fitzhugh-Nagumo neuron with added noise

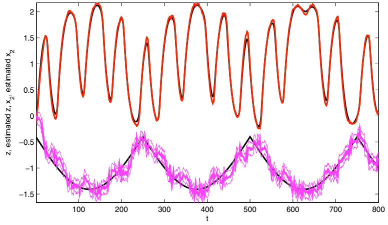
Time series of voltage was observed and (hidden) recovery variable was reconstructed by EnKF.

Further, a parameter was estimated and tracked using the same idea.

Ensemble Kalman Filter (EnKF) for neurons



(a)



(b)

Voss, Timmer, Kurths (2004)

Many further applications followed.

- Control
  Schiff, Sauer (J. Neural Eng. 2008)
- Networks
  Schiff, Sauer (PRE 2009)
- Variety of neural models
  Ullah, Schiff (Assimilating seizure dynamics, PLoS 2010)

Network structure from time series

Data assimilation requires a model. A good model?

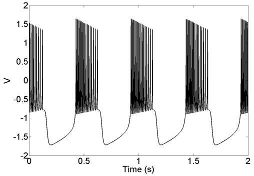
We consider connection strengths in an n-node network as $n(n-1)$ parameters, and ask EnKF to fit the parameters from the voltage measurements only.

Our strategy was to pick a generic spiking model (Hindmarsh-Rose) and try to reconstruct networks of

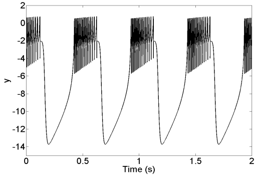
1. HR neurons, with correct and incorrect parameter settings
2. Other neuron models (HH, FN, etc.)
3. MEA data

What makes the approach statistical is that the entire covariance matrix of all variables is carried along and refined during the run. We can use the covariance of the parameters generated by EnKF to generate confidence intervals for the connection strengths.

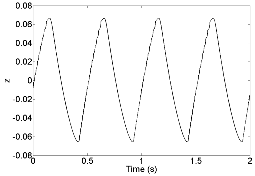
Hindmarsh-Rose Model



V



y

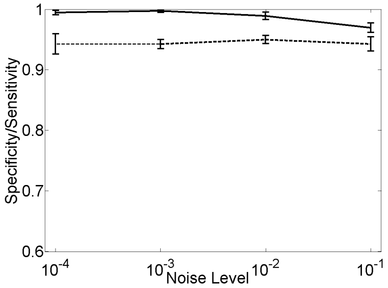


z

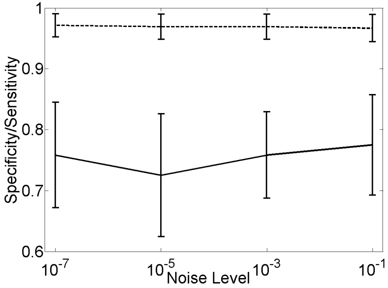
$$\begin{aligned}
\dot{V} &= y - V^3 + 3V^2 - z + I \\
\dot{y} &= 1 - bV^2 - y \\
\dot{z} &= \tau(s(V+1) - z)
\end{aligned}$$

Typical bursting dynamics with parameter values $b = 5$, $k = 4$, $\tau = 0.001$ and $s = 4$.

Network structure from time series



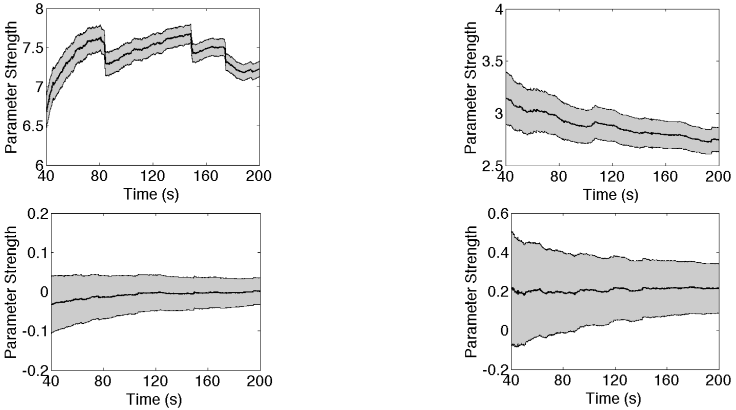
10-node Hindmarsh-Rose network, Hindmarsh-Rose EnKF model



10-node Hodgkin-Huxley network, Hindmarsh-Rose EnKF model

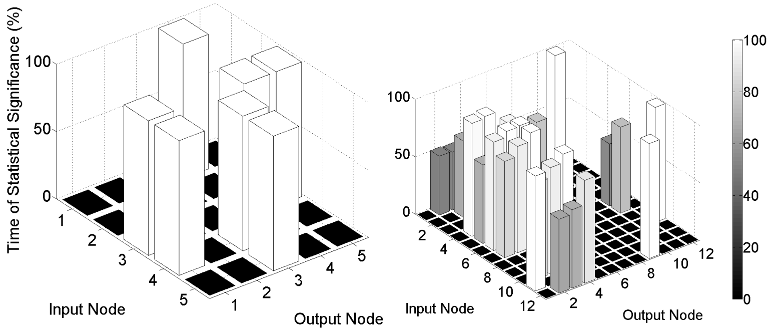
Network structure from time series

Four network connections from MEA data tracked over 160 of time series.

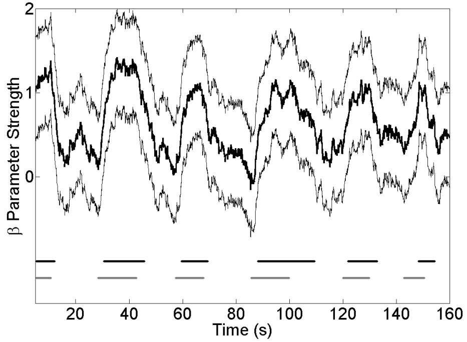


Shaded area denotes 95% confidence region.

Results of two MEA experiments



This method is realtime and adaptive



Hodgkin-Huxley network data assimilated using Hindmarsh-Rose models.

A connection was turned on/off randomly. Bottom traces show the actual "on" time (grey bar) and the filter-estimated "on" time of the connection (black bar). In the upper trace, the 95% confidence region is shown.

Outline

1. Network structure from spike trains
   - Cox method from survival analysis
   - Levenberg-Marquardt
   - Statistics
2. Network structure from time series
   - Data assimilation approach
   - Simulation and results
   - Neural cultures
3. Learning unmodeled variables
   - Model error
   - Hodgkin-Huxley
   - Potassium from neural culture

Kalman Filter

Assume a model

$$\begin{aligned} \dot{x} &= f(x, p) + \omega_t \\ y &= h(x) + \nu_t. \end{aligned}$$

Observe y_t^o.

Possible goals:

1. Reconstruct state x from noisy observations
2. Reconstruct state x from noisy, incomplete observations

For example,

$$h(x) = h(x_1, \ldots, x_n) = x_1$$

Want to reconstruct $x_2(t), \ldots, x_n(t)$ as well as $x_1(t)$.

Multiple Model Ensemble Kalman Filter

$$\begin{aligned} \dot{w} &= F(w) + \omega_t \\ \begin{bmatrix} y \\ \vdots \\ y \\ z \end{bmatrix} &= H(w) + \nu_t \end{aligned}$$

where

$$w = \begin{bmatrix} x^1 \\ \vdots \\ x^m \\ c_1 \\ \vdots \\ c_m \\ d \end{bmatrix}, \qquad F = \begin{bmatrix} f(x^1, p_1) \\ \vdots \\ f(x^m, p_m) \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix}, \qquad H(w) = \begin{bmatrix} h(x^1) \\ \vdots \\ h(x^m) \\ \sum_{j=1}^m c_i^T x^i + d \end{bmatrix}.$$

During training phase, observe y and unmodeled variable z ;
During prediction phase, observe y only and predict z as $\sum_{j=1}^m c_i^T x^i + d$.

Multiple Model Ensemble Kalman Filter

$$\begin{aligned} \dot{w} &= F(w) + \omega_t \\ \begin{bmatrix} y \\ \vdots \\ y \\ \cancel{z} \end{bmatrix} &= H(w) + \nu_t \end{aligned}$$

where

$$w = \begin{bmatrix} x^1 \\ \vdots \\ x^m \\ \cancel{\epsilon_1} \\ \vdots \\ \cancel{\epsilon_m} \\ \cancel{d} \end{bmatrix}, \qquad F = \begin{bmatrix} f(x^1, p_1) \\ \vdots \\ f(x^m, p_m) \\ \cancel{0} \\ \vdots \\ \cancel{0} \\ \cancel{0} \end{bmatrix}, \qquad H(w) = \begin{bmatrix} h(x^1) \\ \vdots \\ h(x^m) \\ \cancel{\sum_{j=1}^m c_i^T x^i + d} \end{bmatrix}.$$

During training phase, observe y and unmodeled variable z ;
During prediction phase, observe y only and predict z as $\sum_{j=1}^m c_i^T x^i + d$.

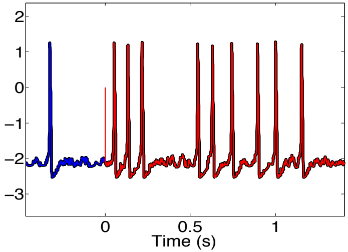
Reconstructing unmodeled variables

Example. Can we use the EnKF with multiple versions of the Hindmarsh-Rose neuron to build a predictor for one of the Hodgkin-Huxley gating variables (m, n, h)?

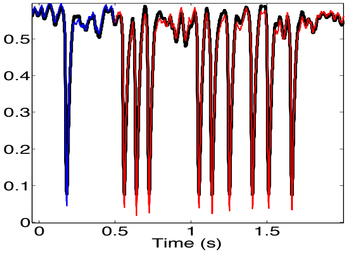
Both HR and HH models have a voltage variable, but HR has no variable that resembles the HH gating variables.

Why it might work: Using versions of HR with different time constants may allow a reasonable fit.

Reconstructing unmodeled variable H



V



H

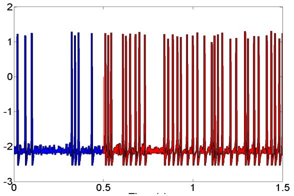
During training phase, parameters c_i are estimated, and used to predict during prediction phase. Exact data is shown in black.

Red curve on right side is

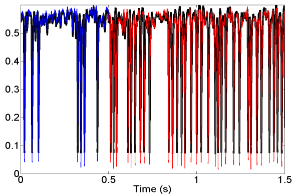
$$h(t) = -0.42y_1 + 0.48y_2 + 0.33z_1 - 0.16z_2 - 0.83$$

where y_1, z_1 and y_2, z_2 are output from filters using two slightly different versions of Hindmarsh-Rose.

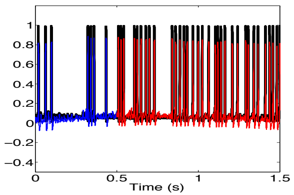
Reconstructing Hodgkin-Huxley gating variables



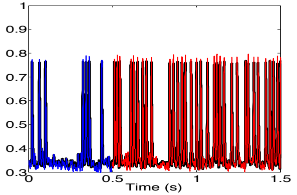
V



H



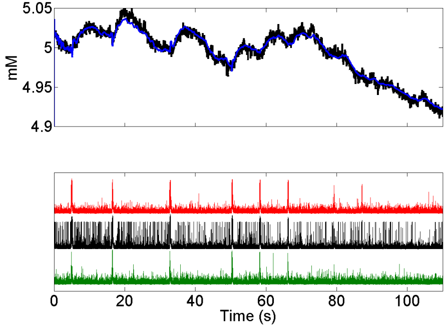
M



N

Hodgkin-Huxley gating variables H, M, and N are reconstructed from V using two versions of HR. Blue = training phase. Red = prediction phase.

Recovering potassium from voltages

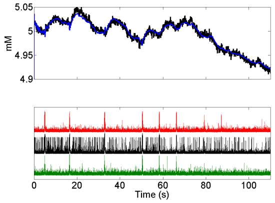


Top trace (black) is measurement of potassium concentration at particular point in dish.

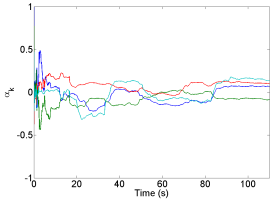
Lower traces are voltages from three nearby electrodes.

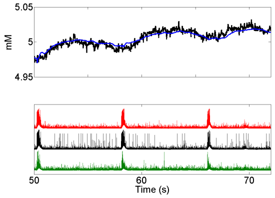
We used Multiple Model DA with the Hindmarsh-Rose model to learn the unmodeled variable (potassium). Training phase is shown here. Blue curve is the fit of the potassium variable.

Recovering potassium from voltages

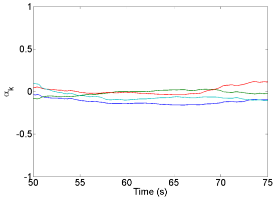


Training phase





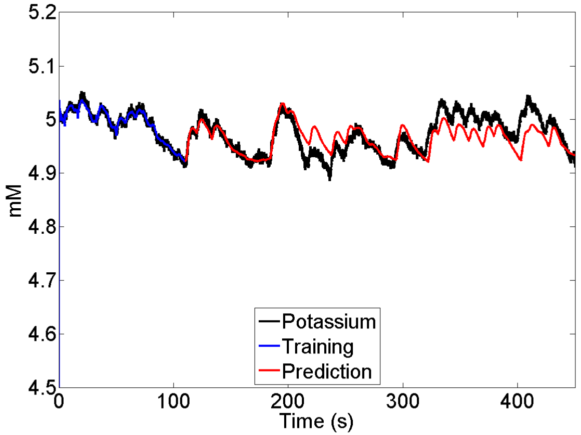
Detail



Potassium and voltages

Four of the largest coefficients during training phase.

Recovering potassium from voltages



Summary

1. The Cox method is a slow but efficient statistical technique for network inference from spike trains, with high specificity (control of false positives) and sensitivity (control of false negatives).

2. Standard data assimilation methods can be used to build a realtime, statistical algorithm for tracking network links from time series.

3. Multiple model data assimilation can be used along with training data to track unmodeled variables.