Modeling Your Spiking Data
with
Generalized Linear Models

Outline

I. Part One: Specifying models

a. The generalized linear model form

b. Fitting parameters

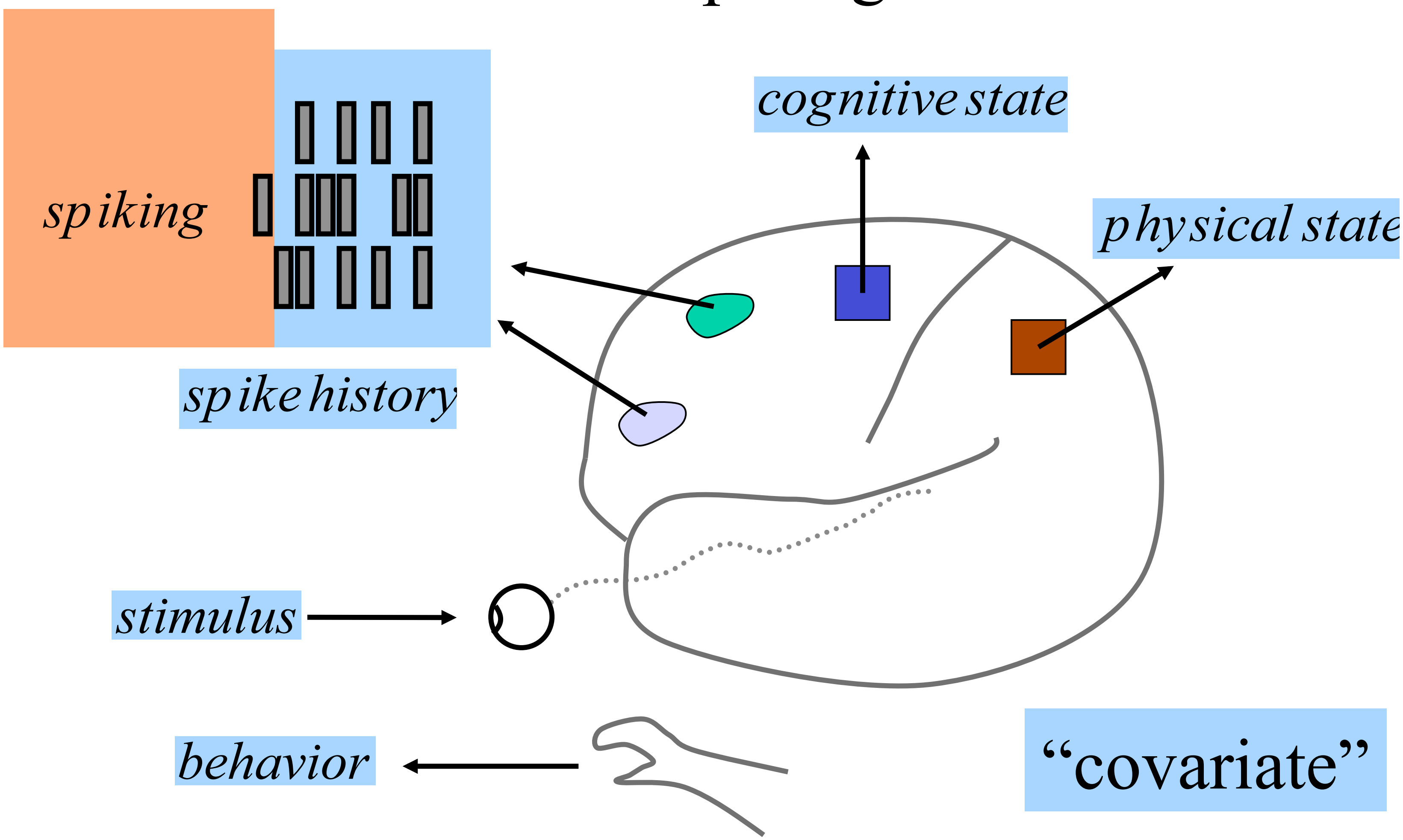
II. Part Two: Evaluating model quality

a. Parameter Uncertainty

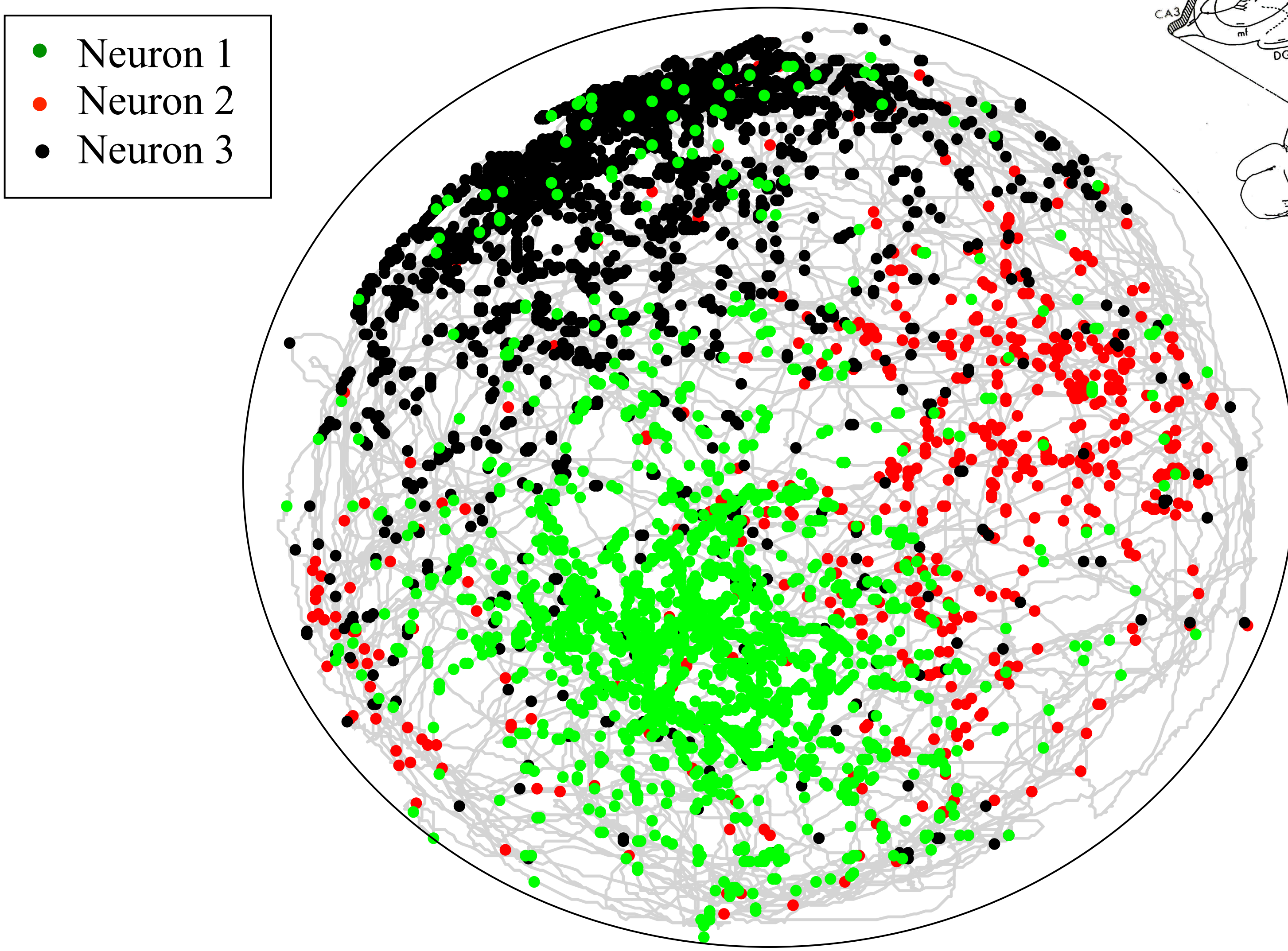
b. Relative: Akaike info. criterion (AIC)

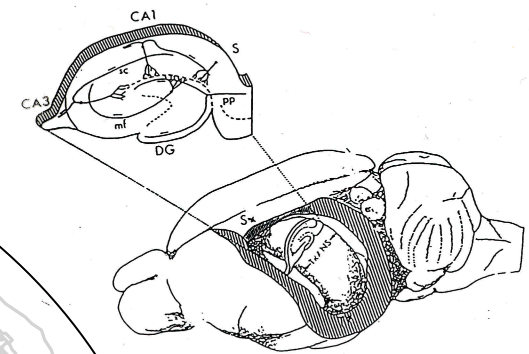
c. Absolute: Time rescaling theorem

What are Spiking Models?



Today's Case Study:
CA1 Hippocampal Place Cells





Challenges in Modeling Spikes

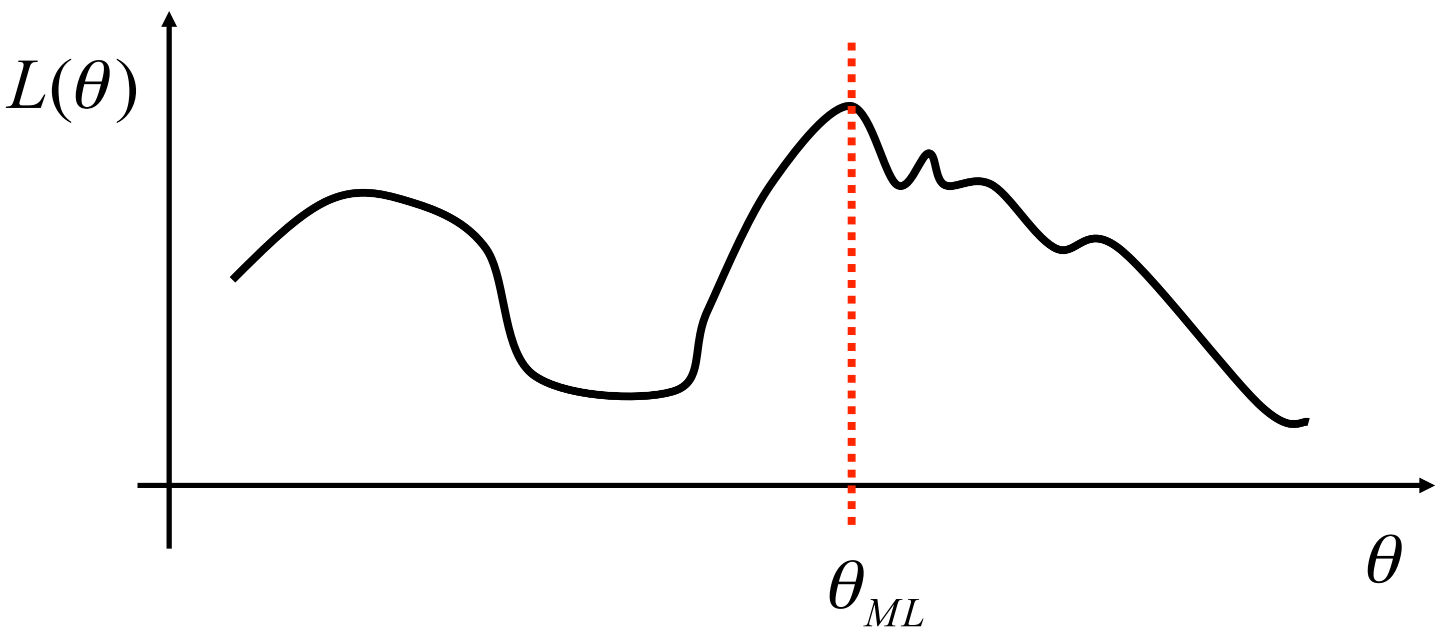
- Your neurons are in a vast network
- Each neuron has many biophysical parameters
- Biological noise
- Memory distributed across the network

Solution: Stochastic Models

$$p(\textit{ensemble spiking} \mid \textit{covariates}; \theta)$$

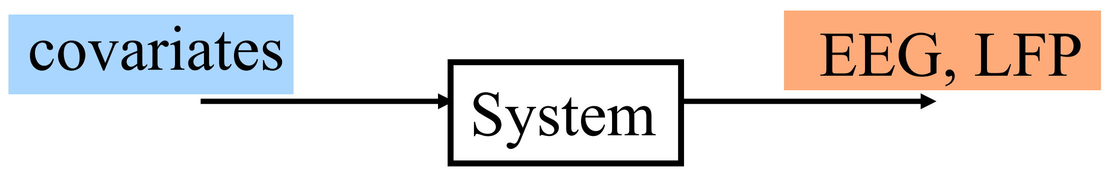
Data Likelihood &
Maximum Likelihood (ML) Parameter Fitting

$$L(\theta) = f(ensemble\ spiking \mid covariates; \theta)$$



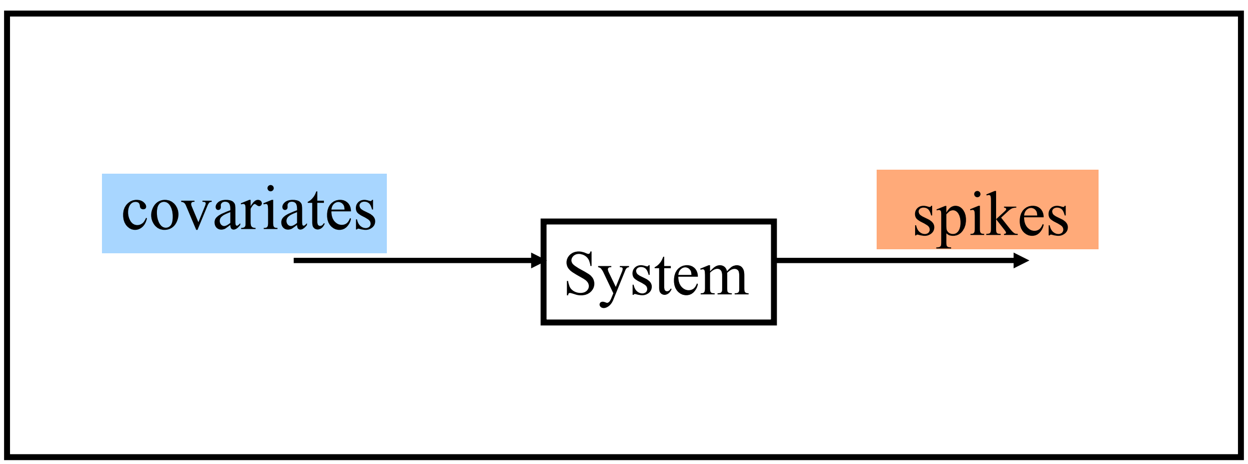
Introducing Generalized Linear Models

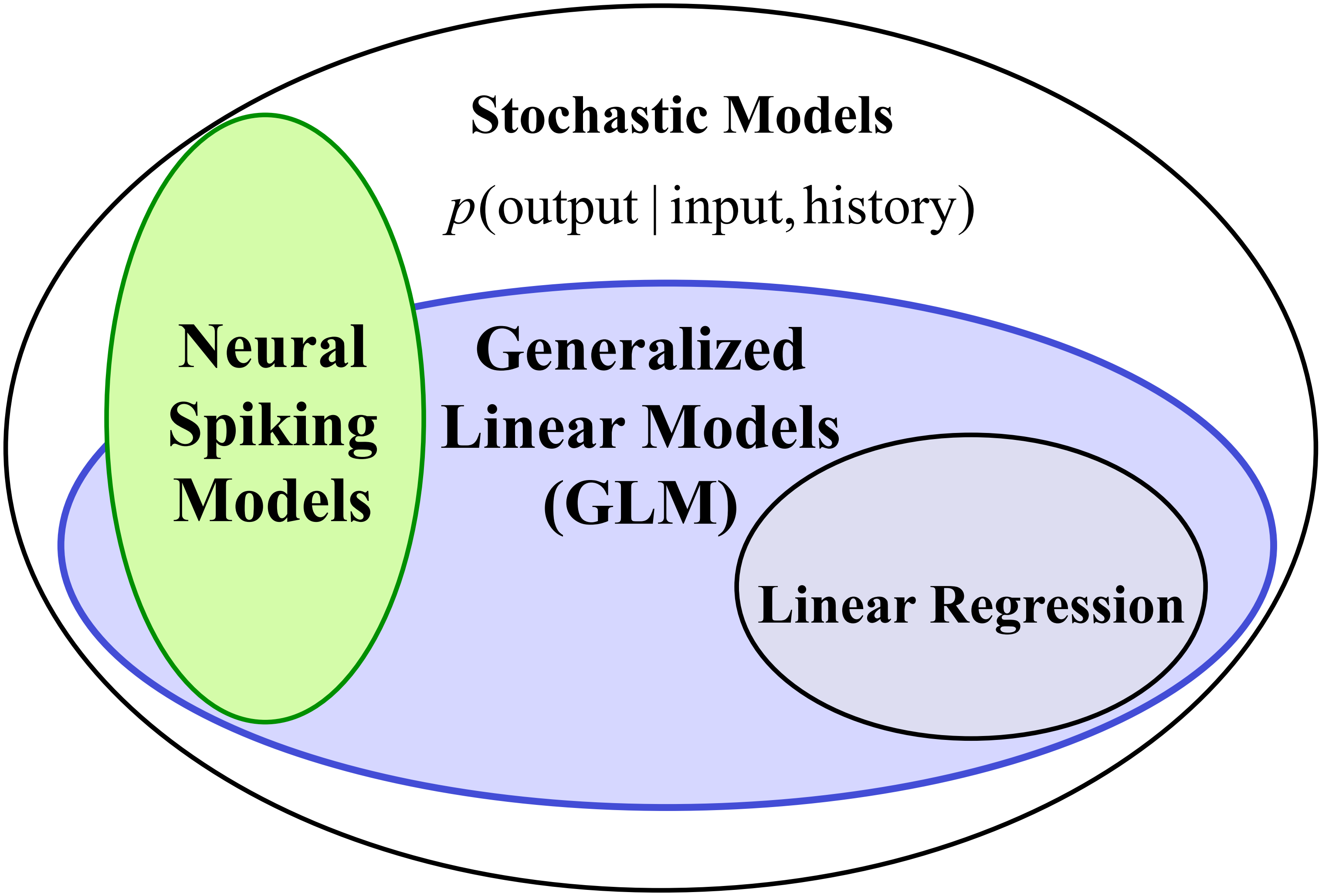
$$p(\textit{neural activity} \mid \textit{covariates})$$



Linear Regression
(Gaussian Model of Variability)

$$Y = X\beta + \varepsilon$$





Properties of GLM:

- Convex likelihood surface
- Estimators asymptotically have minimum MSE

Point Process Data Likelihood

The Conditional Intensity Function:

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

In Discrete Time:

Intensity Model: $\lambda_k = \lambda(t_k \mid H_{t_k})$

Observed Spike Data: $\Delta N_k = N(t_k + \Delta t) - N(t_k)$

Data Likelihood, one neuron:

$$L(\text{Spike Train} \mid \theta) = \exp\left(\sum_{k=1}^{T} \log(\lambda_k \Delta t)\Delta N_k - \lambda_k \Delta t\right)$$

The Generalized Linear Model

The Exponential Family of Distributions

$$L(\theta) = f(y \mid \theta) = \prod_{k=1}^{K} \exp\{T(y_k)C(\theta) + H(y_k) + D(\theta)\}$$

To establish a <u>Generalized Linear Model</u> set the natural parameter $C(\theta)$ **to be a linear function of the covariates**

$$C(\theta) = \theta_0 + \sum_{j=1}^{J} \theta_j x_j$$

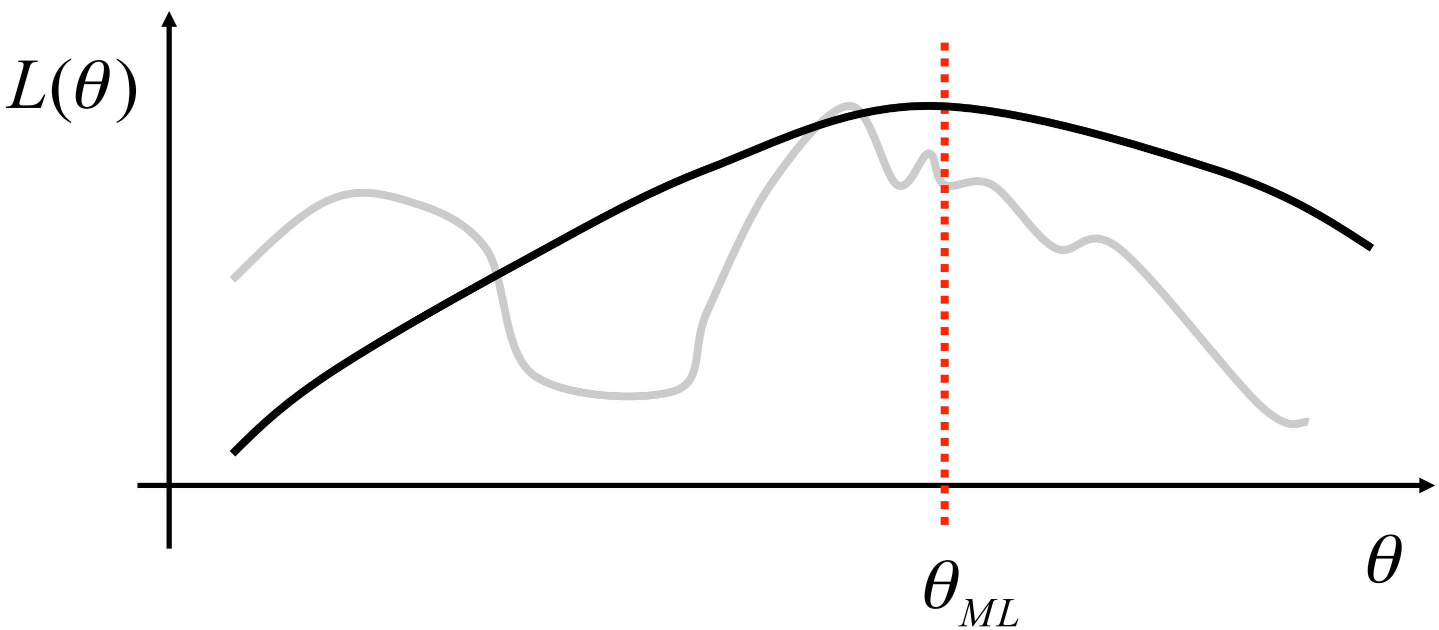
In this case:

$$\log\left(\lambda(t_k \mid H_{t_k}\right) = \theta_0 + \sum_{j=1}^{J} \theta_j x_j(t_k)$$

This model form makes ML parameter estimation easy.

ML estimation is easier with this GLM: Convex Likelihood

$$L(\theta) = f(\textit{ensemble spiking} \mid \textit{covariates}; \theta)$$



Convex Likelihood

$$L(\theta) = \exp\left[\sum_{k=1}^{T}\left(\log(\lambda_k \Delta t) n_k - \lambda_k \Delta t\right)\right]$$

$$l = \log(L) = \sum_{k=1}^{T}\left[\underline{x}_k^{/}\theta n_k + \log\left(\Delta\right) n_k - \lambda_k \Delta t\right]$$

$$\frac{\partial l}{\partial \theta} = \sum_{k=1}^{T}\left[x_k n_k - x_k \exp\left(\underline{x}_k^{/}\theta\right)\Delta\right]$$

$$\frac{\partial^2 l}{\partial^2 \theta} = \sum_{k=1}^{T}\Delta\left[-x_k \exp\left(\underline{x}_k^{/}\theta\right) x_k^{/}\right] = -\sum_{k=1}^{T}\lambda_k \Delta\, x_k x_k^{/} < 0$$

Intensity Model

$$\log \lambda_k(\theta) = \underline{x}_k^{'}\theta$$

Fitting GLM

- In general, no closed form ML estimator
- Use numerical optimization, such as MATLAB's `glmfit`

Summary: GLM, Part I

1. extends the concept of linear regression
2. allows for modeling spikes
3. easy to fit model parameters

1: Plot raw data

```
>> load glm_data.mat
>> who
```

Your variables are:

spiketimes	**Time stamps on spike events**
x_at_spiketimes	**Animal position (x) at corresponding times stamps**
y_at_spiketimes	**Animal position (y) at corresponding times stamps**
T	**Time**
spikes_binned	**spike data in 33 ms bins**
xN	**Normalized position (x)**
yN	**Normalized position (y)**
vxN	**Normalized velocity (v_x)**
vyN	**Normalized velocity (v_y)**
r	**Movement speed = sqrt(v_x^2+ v_y^2)**
phi	**Movement direction = atan2(v_y,v_x)**

Example call to **`glmfit`**

```
b = glmfit([xN yN],spikes_binned,'poisson');
```

Visualizing model from **`glmfit`**

```
lambda = exp(b(1) + b(2)*x_new + b(3)*y_new);
```

Quadratic model with **glmfit**

```
b = glmfit([xN yN xN.^2 yN.^2 xN.*yN],…
            spikes_binned,'poisson');
```

Visualizing this model from **glmfit**

```
lambda = exp(b(1) + b(2)*x_new …
          + b(3)*y_new + b(4)*x_new.^2 …
          + b(5)*y_new.^2 …
          + b(6)*x_new.*y_new);
```

Fitting other covariates with **glmfit**

```
b=glmfit([vxN vxN.^2],spikes_binned,'poisson');
```

Visualizing this model from **glmfit**

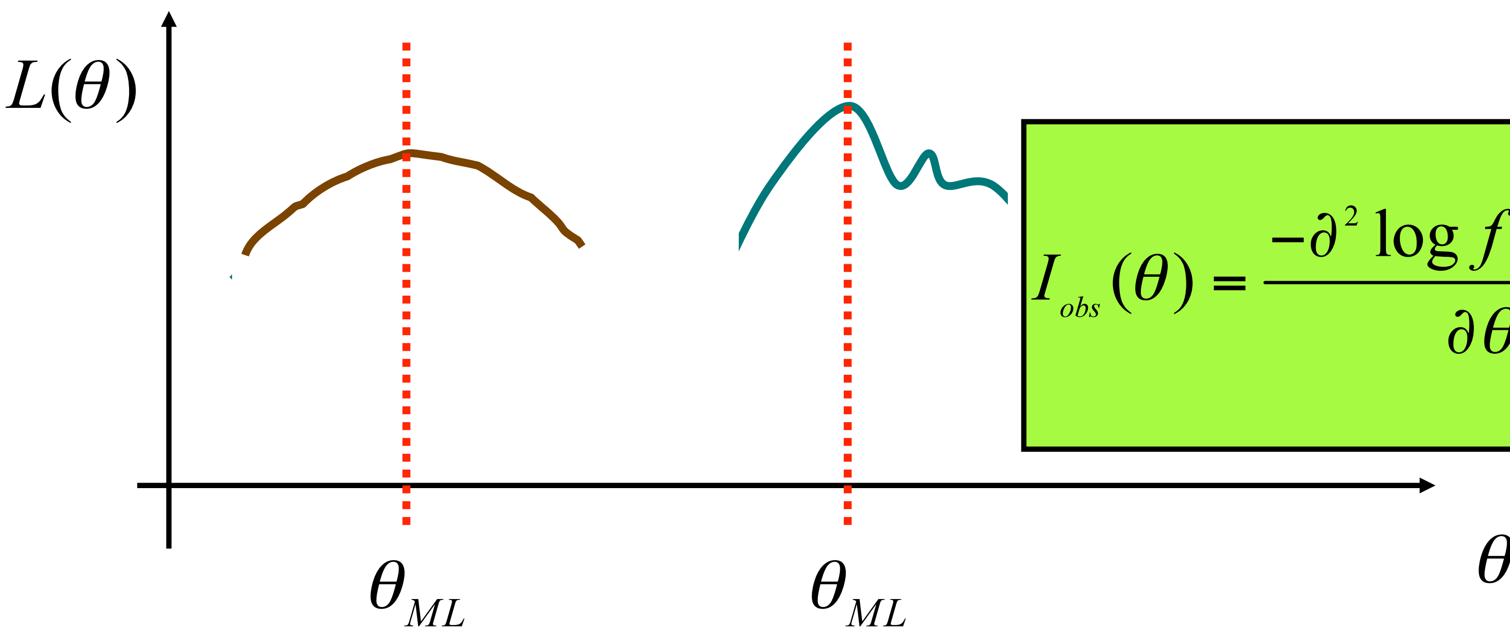
```
plot(velocities,exp(b(1)+b(2)*velocities…
                   +b(3)*velocities.^2),'r');
```

Part II: Model Quality

1. Parameter uncertainty
2. Relative: Akaike Info. Criterion (AIC)
3. Absolute: Time rescaling theorem

ML Parameter Uncertainty

$$L(\theta) = f(ensemble\ spiking \mid covariates; \theta)$$



$$I_{obs}(\theta) = \left. \frac{-\partial^2 \log f(x_{obs} \mid \theta)}{\partial \theta^2} \right|_{\hat{\theta}_{ML}}$$

ML Parameter Uncertainty

```
>> load glm_data.mat

>>  [b,dev,stats] = glmfit( put inputs here);
```

`stats.se` – **standard error on parameter estimates**

`stats.p` – **significance of parameter estimates**

```
errorbar(1:length(b), b , 2*stats.se );
```

Akaike Info. Criterion

- Approximates relative Kullback-Leibler divergence between real distribution and model

$$E_{g(Y)}\left[\log \frac{g(Y)}{f\left(Y;\theta\right)}\right]$$

- Formula: $AIC = -2\log\left[\, p(data|\,\hat{\theta}_{ML})\right] + 2\,P$

Computing AIC from **`glmfit`**

```
>> [b,dev,stats] = glmfit( put inputs here);

>> AIC = dev+2*length(b)
```

5. Time Rescaling Theorem

Let's return to the probability distribution for a single ISI:

$$p\big(t \mid x(t)\big) = \lambda(t \mid x(t)) \cdot \exp\left[-\int_0^t \lambda\big(u \mid x(u)\big)du\right]$$

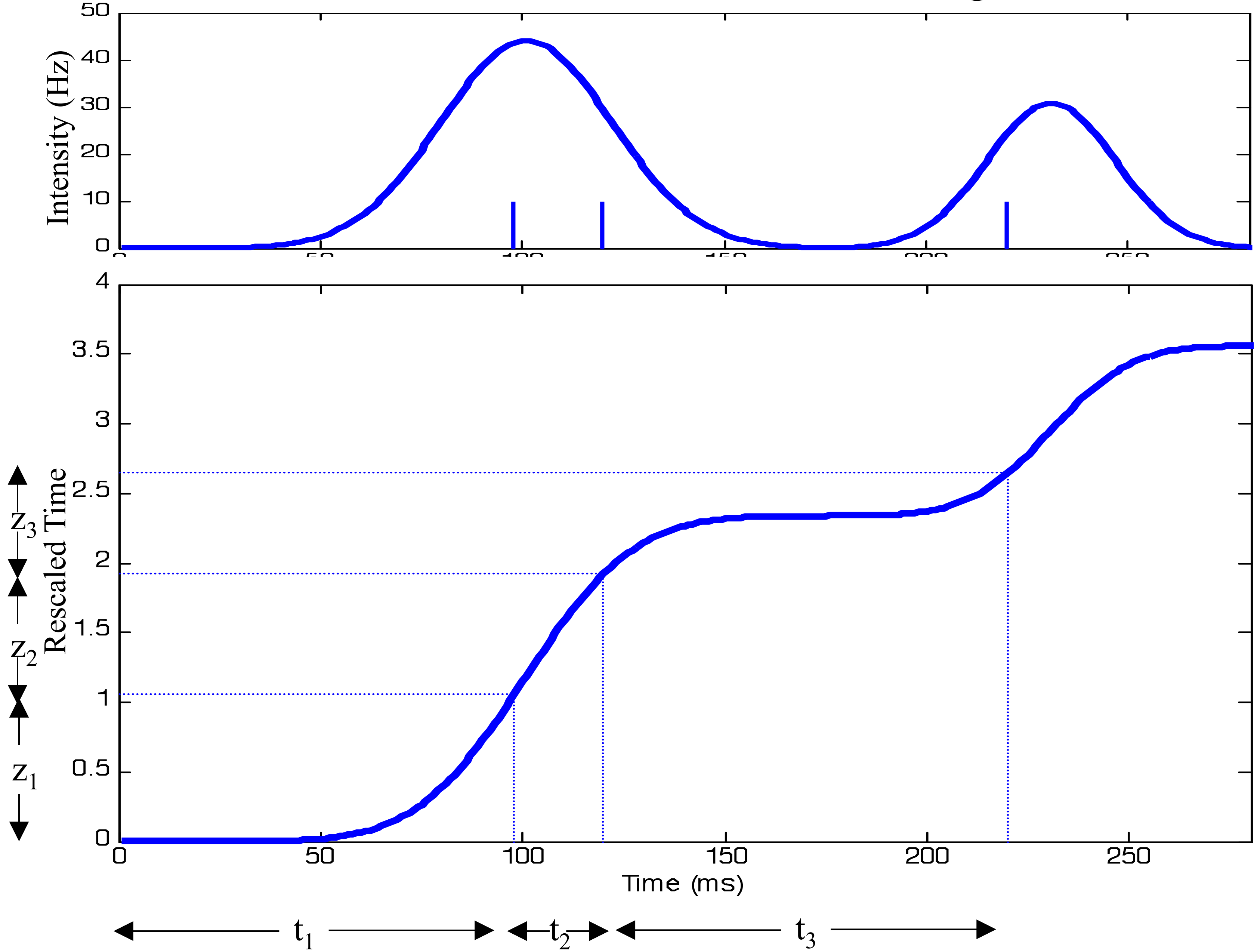
and make a change of variables,

$$z = \int_0^t \lambda\big(u \mid x(u)\big)du$$

Then the distribution $p(z) = p\big(t \mid x(t)\big)\left|\frac{dt}{dz}\right| = \exp(-z)$

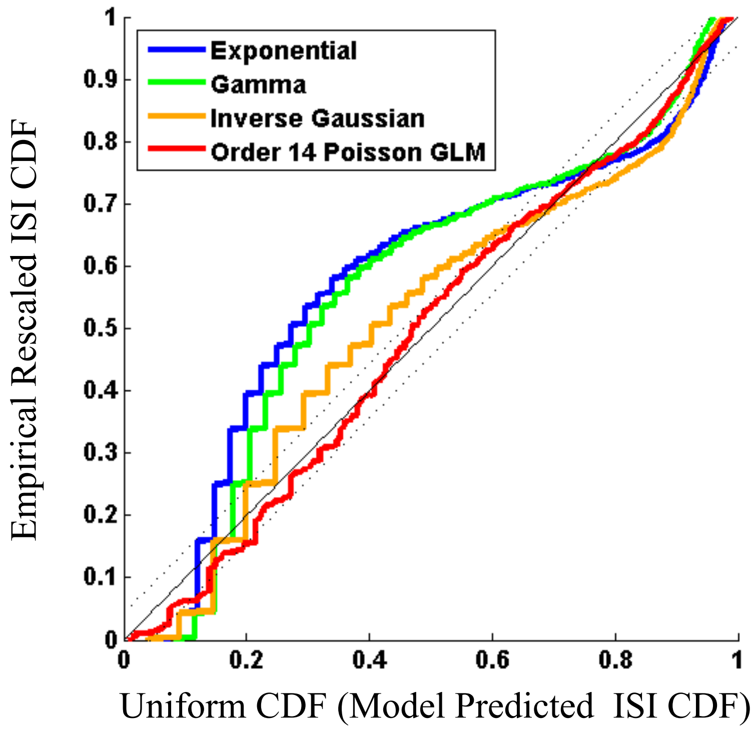
is exponential with parameter $\lambda = 1$.

Illustration: Time Rescaling



Time Rescaling Statistic: KS Plots

Graphical measure of goodness-of-fit, based on time rescaling, comparing an empirical and model cumulative distribution function. If the model is correct, then the rescaled ISIs are independent, identically distributed random variables whose ordered quantiles should produce a 45° line [*Ogata, 1988*].



Construct KS Plot for Linear Model

```
>> glm_ks
```

Construct KS Plot for Quadratic Model

Modify **`glm_ks.m`**

```
lambdaEst = exp( b_quad(1) + b_quad(2)*xN
      + b_quad(3)*yN + b_quad(4)*xN.^2
      + b_quad(5)*yN.^2 + b_quad(6)*xN.*yN );
```

Questions to Ask by Modeling Spikes

- What stimulus features drive the spiking?
- How does the spiking vary with a covariate (stimulus, behavioral measure, cognitive state, or spike history,)?
- How well does the spiking represent a covariate ?
- How do neurons or brain regions cooperate in spiking?
- What physical processes determine the spiking?

Summary: GLM, Part II

1. extends the concept of linear regression
2. allows for modeling spikes
3. easy to fit model parameters
4. Fisher information and parameter uncertainty
5. Relative model quality: AIC
6. Absolute model quality: Time Rescaling