Generalized Linear Models (GLMs)
Example 3: unknown neuron

Be the computational neuroscientist: what model would you use?
Example 3: unknown neuron

More general setup: \( y \sim Poiss(\lambda) \)

\[ \lambda = f(\theta x) \]

for some nonlinear function \( f \)
Quick Quiz:

The distribution $P(y|x, \theta)$ can be considered as a function of $y$, $x$, or $\theta$.

What is $P(y|x, \theta)$:

1. as a function of $y$?
   Answer: **encoding distribution** - probability distribution over spike counts

2. as a function of $\theta$?
   Answer: **likelihood function** - the probability of the data given model params

3. as a function of $x$?
   Answer: **stimulus likelihood function** - useful for ML stimulus decoding!
What is this?

Stimulus likelihood function
(for decoding)

\[ P(y = 20 | x, \theta) \]

\[ \hat{x}_{ML} \]
GLMs

• Be careful about terminology:

GLM ≠ GLM

General Linear Model ≠ Generalized Linear Model (Nelder 1972)

Linear ≠ Linear
Stephen Senn: I must confess to having some confusion when I was a young statistician between general linear models and generalized linear models. Do you regret the terminology?

John Nelder: I think probably I do. I suspect we should have found some more fancy name for it that would have stuck and not been confused with the general linear model, although general and generalized are not quite the same. I can see why it might have been better to have thought of something else.
Moral:
Be careful when naming your model!
1. General Linear Model

Examples:

1. Gaussian
   \[ y = \theta \cdot \vec{x} + \epsilon \]

2. Poisson
   \[ y \sim \text{Pois}ss(\theta \cdot \vec{x}) \]
2. Generalized Linear Model

Examples:
1. Gaussian \[ y = f(\theta \cdot \vec{x}) + \epsilon \]
2. Poisson \[ y \sim \text{Poisson}(f(\theta \cdot \vec{x})) \]
2. Generalized Linear Model

Terminology:

- \( \vec{x} \) → Linear
- Nonlinear
- Noise (exponential family) → \( y \)

\[ f^{-1} \] = “link function”

- “distribution function”

- “parameter”

\( \vec{\theta} \)
From spike counts to spike trains:

\[ y_t = \hat{k} \cdot \vec{x}_t + \text{noise} \]

First idea: linear-Gaussian model!

Response at time \( t \)

\[ y_t = \hat{k} \cdot \vec{x}_t + \epsilon_t \]

Stimulus

Response

Time
The response at time $t$ is given by:

$$y_t = \vec{k} \cdot \vec{x}_t + \text{noise}$$

where $\vec{k}$ is the linear filter, $\vec{x}_t$ is the vector stimulus at time $t$, and $N(0, \sigma^2)$ represents the noise distribution.

The stimulus is shown in the graph, with a time bin at a time $t = 1$.

The response is also shown, with spikes at the time bins.

The data is walked through one time bin at a time.
\[ y_t = \vec{k} \cdot \vec{x}_t + \text{noise} \]

### Walk through the data
- One time bin at a time
- Stimulus
- Response
- Time

**Linear Filter**

**Vector Stimulus at Time** $t$

\[ N(0, \sigma^2) \]
response at time $t$

$$y_t = \vec{k} \cdot \vec{x}_t + \text{noise}$$

linear filter

vector stimulus at time $t$

$N(0, \sigma^2)$

walk through the data one time bin at a time

$t = 3$

stimulus

response

time $y_t$
response at time $t$

$$y_t = \vec{k} \cdot \vec{x}_t + \text{noise}$$

linear filter

vector stimulus at time $t$

$N(0, \sigma^2)$

walk through the data one time bin at a time

$t = 4$

stimulus

response

time $\rightarrow$ $y_t$
response at time $t$

$$y_t = \vec{k} \cdot \vec{x}_t + \text{noise}$$

linear filter

vector stimulus at time $t$

walk through the data
one time bin at a time

$t = 5$

stimulus

response

time $\rightarrow$ $y_t$

$N(0, \sigma^2)$
response at time $t$

\[ y_t = \vec{k} \cdot \vec{x}_t \ + \ \text{noise} \]

linear filter

vector stimulus at time $t$

walk through the data one time bin at a time

$t = 6$

stimulus

response

time  \rightarrow \quad y_t \quad N(0, \sigma^2)
Build up to following matrix version:

\[ Y = X \vec{k} + \text{noise} \]

\[
\begin{bmatrix}
0 \\
1 \\
0 \\
\vdots \\
\end{bmatrix}
= \begin{bmatrix}
\begin{array}{c}
\vdots \\
\end{array}
\end{bmatrix}
\begin{bmatrix}
\vec{k}
\end{bmatrix}
\]

**design matrix**
Build up to following matrix version:

\[
Y = X \kappa + \text{noise}
\]

least squares solution:

\[
\hat{\kappa} = (X^T X)^{-1} X^T Y
\]

(maximum likelihood estimate for “Linear-Gaussian” GLM)
Formal treatment: scalar version

model: \[ y_t = \vec{k} \cdot \vec{x}_t + \epsilon_t \]

equivalent to writing: \[ y_t | \vec{x}_t, \vec{k} \sim \mathcal{N}(\vec{x}_t \cdot \vec{k}, \sigma^2) \]

or\[ p(y_t | \vec{x}_t, \vec{k}) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(y_t - \vec{x}_t \cdot \vec{k})^2}{2\sigma^2}} \]

For entire dataset: \[ p(Y | X, \vec{k}) = \prod_{t=1}^{T} p(y_t | \vec{x}_t, \vec{k}) \quad \text{(independence across time bins)} \]

\[ = (2\pi\sigma^2)^{-\frac{T}{2}} \exp(-\sum_{t=1}^{T} \frac{(y_t - \vec{x}_t \cdot \vec{k})^2}{2\sigma^2}) \]

\[ \log P(Y | X, \vec{k}) = -\sum_{t=1}^{T} \frac{(y_t - \vec{x}_t \cdot \vec{k})^2}{2\sigma^2} + \text{const} \quad \text{log-likelihood} \]
Formal treatment: vector version

\[ Y = X \tilde{k} + \tilde{\epsilon} \]

\[ \begin{bmatrix} 0 \\ 1 \\ 0 \\ \vdots \end{bmatrix} = \begin{bmatrix} \vdots \\ \vdots \end{bmatrix} \begin{bmatrix} \tilde{k} \\ \vdots \end{bmatrix} + \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \epsilon_3 \\ \vdots \end{bmatrix} \]

equivalent to writing:

\[ Y | X, \tilde{k} \sim N(X \tilde{k}, \sigma^2 I) \]

or

\[ P(Y | X, \tilde{k}) = \frac{1}{|2\pi \sigma^2 I|^T} \exp \left( - \frac{1}{2\sigma^2} (Y - X \tilde{k})^\top (Y - X \tilde{k}) \right) \]

Take log, differentiate and set to zero.
But noise is not Gaussian!

Bernoulli GLM:

\[ p_t = f(\mathbf{x}_t \cdot \mathbf{k}) \]

\[ p(y_t = 1|\mathbf{x}_t) = p_t \]

Equivalent ways of writing:

\[ y_t|\mathbf{x}_t, \mathbf{k} \sim \text{Ber}(f(\mathbf{x}_t \cdot \mathbf{k})) \]

or

\[ p(y_t|\mathbf{x}_t, \mathbf{k}) = f(\mathbf{x}_t \cdot \mathbf{k})^{y_t} \left(1 - f(\mathbf{x}_t \cdot \mathbf{k})\right)^{1-y_t} \]

log-likelihood:

\[ \mathcal{L} = \sum_{t=1}^{T} \left( y_t \log f(\mathbf{x}_t \cdot \mathbf{k}) + (1 - y_t) \log(1 - f(\mathbf{x}_t \cdot \mathbf{k})) \right) \]
Logistic regression:

\[ f(x) = \frac{1}{1 + e^{-x}} \]

• so logistic regression is a special case of a Bernoulli GLM

Bernoulli GLM:

\( p_t = f(\vec{x}_t \cdot \vec{k}) \)

\( p(y_t = 1|\vec{x}_t) = p_t \)

(coin flipping model, \( y = 0 \) or \( 1 \))
Poisson regression

Poisson GLM:

\[
\lambda_t = f(\vec{x}_t \cdot \vec{k}) \quad \text{(firing rate)}
\]

\[
y_t | \vec{x}_t, \vec{k} \sim \text{Poiss}(\Delta \lambda_t) \quad \text{(integer } y \geq 0 \text{ )}
\]

encoding distribution:

\[
p(y_t | \vec{x}_t, \vec{k}) = \frac{(\Delta \lambda_t)^{y_t}}{y_t!} e^{-\Delta \lambda_t}
\]

log-likelihood:

\[
\mathcal{L} = \log p(Y | X, \vec{k}) = \sum_t \left( y_t \log f(\vec{x}_t \cdot \vec{k}) - f(\vec{x}_t \cdot \vec{k}) \right) + \text{const}
\]

\[
= Y^\top \log f(X \vec{k}) - 1^\top f(X \vec{k}) + \text{const}
\]
Summary:

1. “Linear-Gaussian” GLM: \[ Y \mid X, \bar{k} \sim \mathcal{N}(X \bar{k}, \sigma^2 I) \]
   \[ \hat{k} = (X^T X)^{-1} X^T Y \]

2. Bernoulli GLM: \[ y_t \mid \bar{x}_t, \bar{k} \sim \text{Ber}(f(\bar{x}_t \cdot \bar{k})) \]
   \[ \mathcal{L} = Y^\top \log f(X \bar{k}) - (1 - Y)^\top \log(1 - f(X \bar{k})) \]

3. Poisson GLM: \[ y_t \mid \bar{x}_t, \bar{k} \sim \text{Poiss}(\Delta \lambda_t) \]
   \[ \mathcal{L} = Y^\top \log f(X \bar{k}) - 1^\top f(X \bar{k}) \]