# GLMs and Logistic Regression

Jonathan Pillow

Mathematical Tools for Neuroscience (NEU 314)
Fall, 2021

lecture 21

# warm-up problems

## Regression

1. Write down the formula for the least-squares regression solution for weights *w* given a design matrix X and output vector Y

## KL divergence

2. Write down the formula for KL(P,Q), KL divergence between P & Q.

3. When is KL(P,Q) zero?

4. When is KL(P,Q) infinite?

5. Compute the KL(P,Q) for distributions:

   P = [0.5, 0.5, 0 ]<br>
   Q = [0.25, 0.25, 0.5]

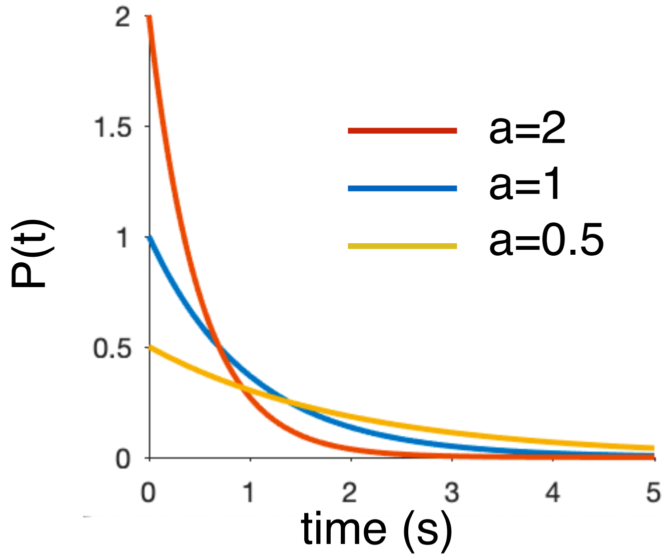
6. Can you describe what this means in terms of yes/no questions?

# warm-up problem: maximum likelihood

The **exponential distribution** describes interspike intervals in a Poisson process (which is famously “memoryless”, meaning that how long you’ve been waiting provides no information about the next spike time).

distribution (PDF)

$$P(t\,|\,a) = ae^{-at}$$



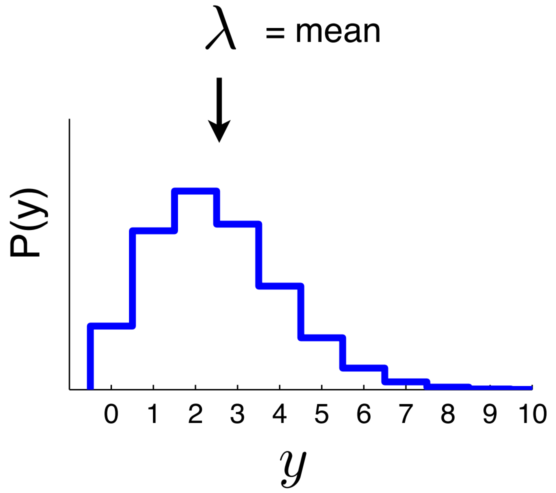
**problem**: Compute the maximum likelihood estimator for the parameter ‘*a’* of an given a set of N observed interspike intervals: $\{t_1, t_2, \ldots, t_n\}$.

# Example 1: linear Poisson neuron

spike count $y \sim Poiss(\lambda)$

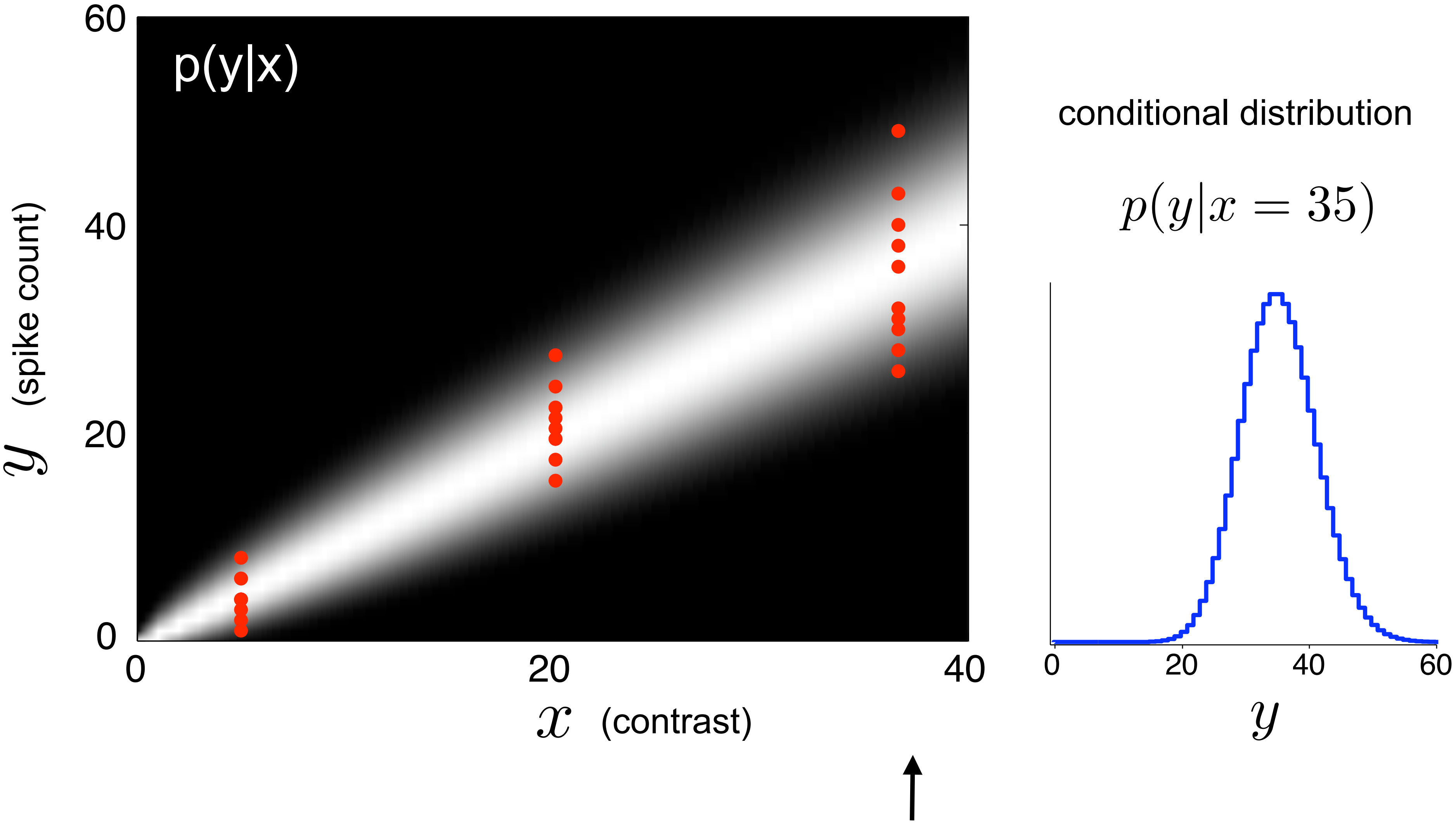
spike rate $\lambda = \theta x$

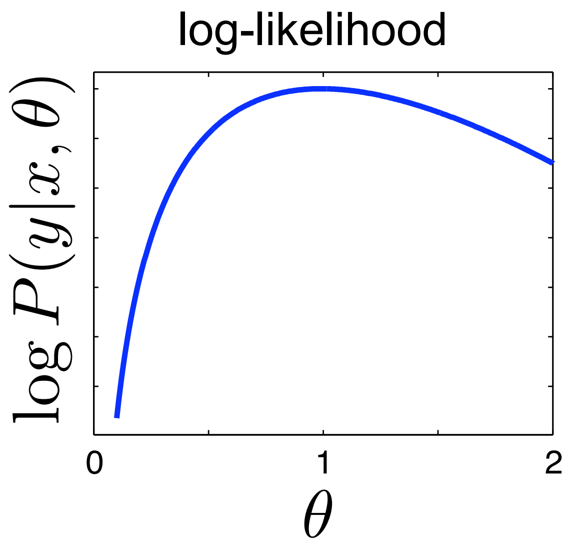
parameter stimulus



encoding model:
$$\begin{aligned} P(y|x,\theta) &= \frac{1}{y!}\lambda^y e^{-\lambda} \\ &= \frac{1}{y!}(\theta x)^y e^{-(\theta x)} \end{aligned}$$

$$\mathrm{mean}(y) = \theta x$$
$$\mathrm{var}(y) = \theta x$$





$$\log P(Y|X,\theta) = \textstyle\sum_i \log P(y_i|x_i,\theta)$$
$$= \sum y_i \log\theta - \theta x_i + c$$
$$= \log\theta(\sum y_i) - \theta(\sum x_i)$$

- Closed-form solution:

$$\tfrac{d}{d\theta} \log P(Y|X,\theta) = \tfrac{1}{\theta} \sum y_i - \sum x_i = 0$$
$$\Longrightarrow \quad \hat{\theta}_{ML} = \frac{\sum y_i}{\sum x_i}$$

(let’s notice: this is kind of a weird result!)

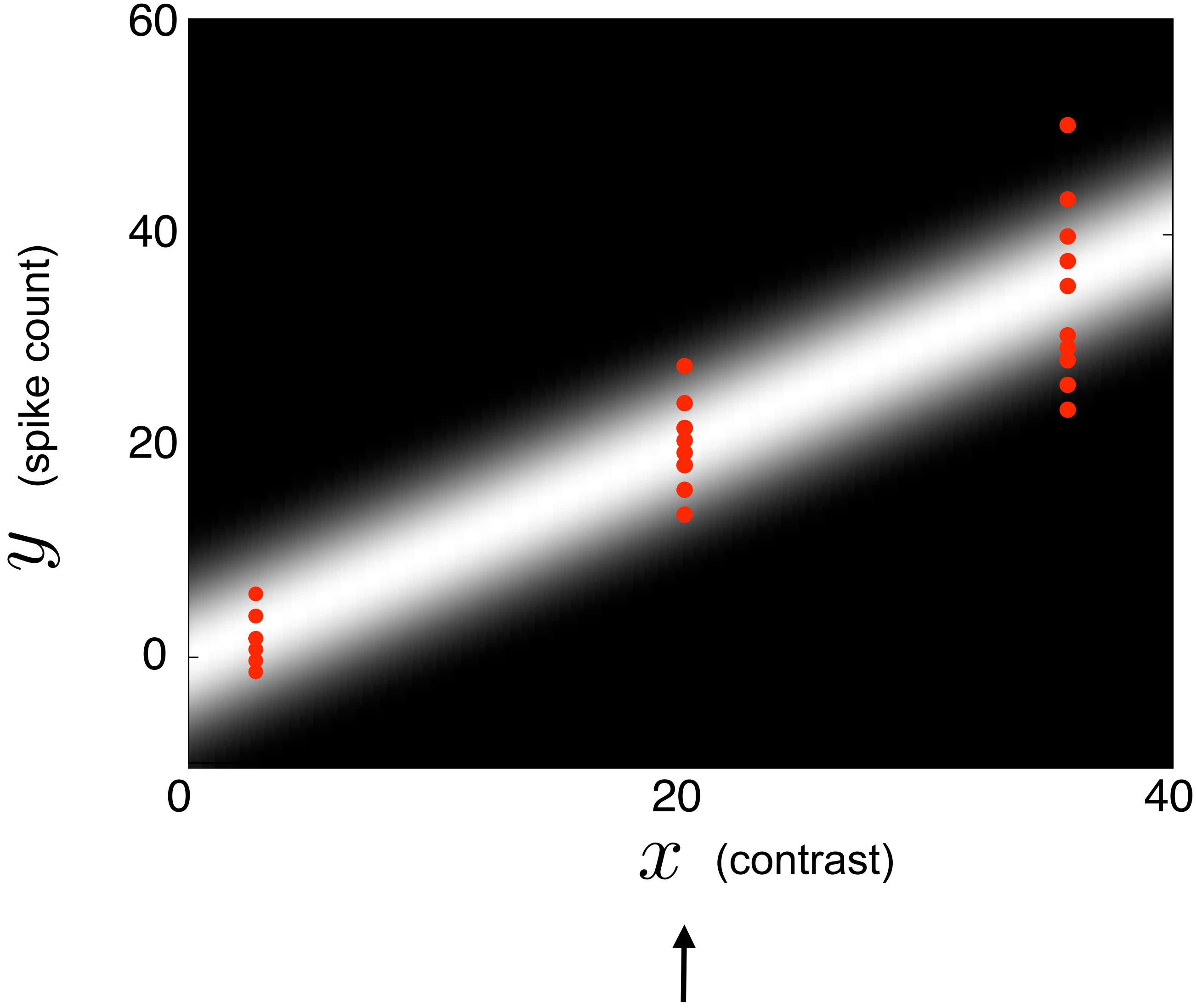
# Example 2: linear Gaussian neuron

spike count $y \sim \mathcal{N}(\mu, \sigma^2)$

spike rate $$\mu = \underset{\substack{\uparrow \\ \text{parameter}}}{\theta}\,\underset{\substack{\nwarrow \\ \text{stimulus}}}{x}$$

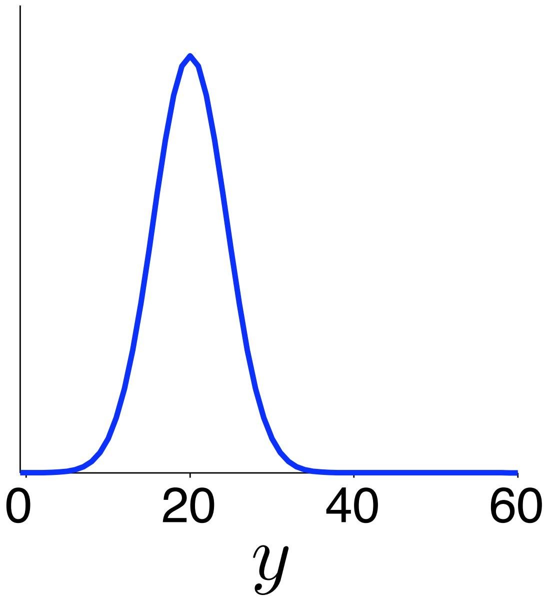
encoding model: $$P(y|x,\theta) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{\frac{-(y-\theta x)^2}{2\sigma^2}}$$

$$\mathrm{mean}(y) = \theta x$$
$$\mathrm{var}(y) = \sigma^2$$



encoding distribution

$p(y|x = 20)$



All slices have same width

$$\boxed{P(y|x,\theta) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{\frac{-(y-\theta x)^2}{2\sigma^2}}}$$

Log-Likelihood $\log P(Y|X,\theta) = -\sum \frac{(y_i - \theta x_i)^2}{2\sigma^2} + c$

Do it: differentiate, set to zero, and solve for $\theta$.

$$\boxed{P(y|x,\theta) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{\frac{-(y-\theta x)^2}{2\sigma^2}}}$$

Log-Likelihood $$\log P(Y|X,\theta) = -\sum \frac{(y_i - \theta x_i)^2}{2\sigma^2} + c$$

$$\frac{d}{d\theta} \log P(Y|X,\theta) = -\sum \frac{(y_i - \theta x_i)x_i}{\sigma^2} = 0$$

$$\sum y_i x_i - \sum \theta x_i^2 = 0$$

$$\theta \sum x_i^2 = \sum y_i x_i$$

Maximum-Likelihood Estimator: $$\hat{\theta}_{ML} = \frac{\sum y_i x_i}{\sum x_i^2}$$

("Least squares regression" solution)

(Recall that for Poisson, $\hat{\theta}_{ML} = \frac{\sum y_i}{\sum x_i}$ )

$$\boxed{P(y|x,\theta) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{\frac{-(y-\theta x)^2}{2\sigma^2}}}$$

Log-Likelihood $\log P(Y|X,\theta) = -\sum \frac{(y_i - \theta x_i)^2}{2\sigma^2} + c$

$$\frac{d}{d\theta} \log P(Y|X,\theta) = -\sum \frac{(y_i - \theta x_i)x_i}{\sigma^2} = 0$$

$$\sum y_i x_i - \sum \theta x_i^2 = 0$$

$$\theta \sum x_i^2 = \sum y_i x_i$$

Maximum-Likelihood Estimator: $\hat{\theta}_{ML} = \frac{\sum y_i x_i}{\sum x_i^2}$

(“Least squares regression” solution)

Matrix version: $\hat{\theta}_{ML} = (X^T X)^{-1} X^T Y$

(this is just least-squares regression!)

# least-squares revisited

(switching $\theta$ to $\vec{k}$)

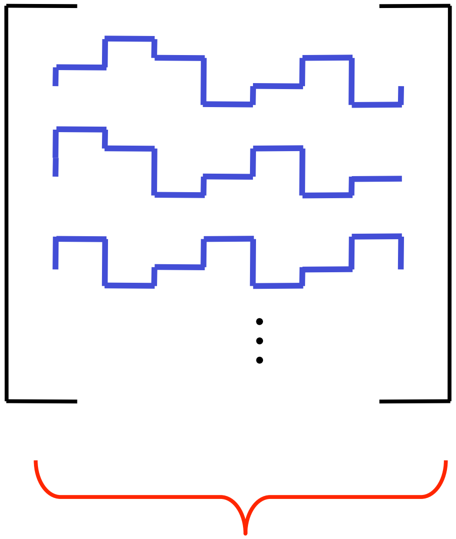
model: $$y_t = \vec{k} \cdot \vec{x}_t \; + \; \epsilon_t$$

$N(0, \sigma^2)$

Guassian noise with variance $\sigma^2$

$$Y \quad = \quad X\vec{k} \quad + \quad noise$$

$$\begin{bmatrix} 0 \\ 0 \\ 1 \\ \vdots \end{bmatrix} \quad =$$



$$\begin{bmatrix} \vec{k} \end{bmatrix} \quad + \quad noise$$

**design matrix**

# least-squares revisited

$N(0,\sigma^2)$

Guassian noise with variance $\sigma^2$

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: $$p(Y|X,\vec{k}) = \prod_{t=1}^{T} p(y_t|\vec{x}_t,\vec{k})$$

(independence across time bins)

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

$$\log P(Y|X,\vec{k}) = -\textstyle\sum_{t=1}^{T}\frac{(y_t-\vec{x}_t\cdot\vec{k})^2}{2\sigma^2} \; + \; const$$

log-likelihood

# least-squares revisited

model: $$y_t = \vec{k} \cdot \vec{x}_t \; + \; \epsilon_t$$

$N(0, \sigma^2)$

Guassian noise with variance $\sigma^2$

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

or

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

**General points**:

1. minimizing a sum of squares is *always* equivalent to maximizing likelihood under a Gaussian noise model!

2. solution doesn't depend on the noise variance $\sigma^2$

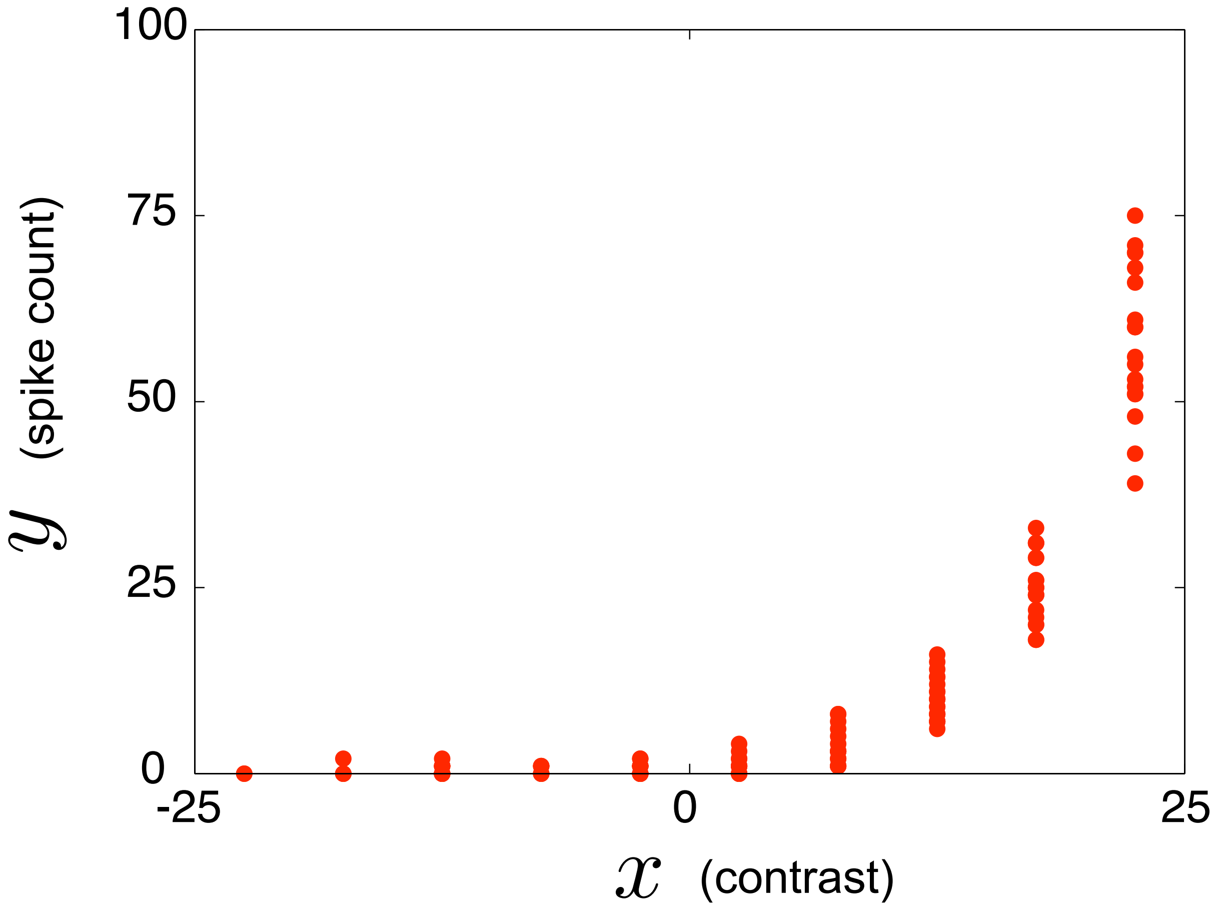
(independence across time bins)

$$\frac{(y_t - \vec{x}_t \cdot \vec{k})^2}{2\sigma^2})$$

$$\log P(Y | X, \vec{k}) = -\sum_{t=1}^{T} \frac{(y_t - \vec{x}_t \cdot \vec{k})^2}{2\sigma^2} \; + \; const$$

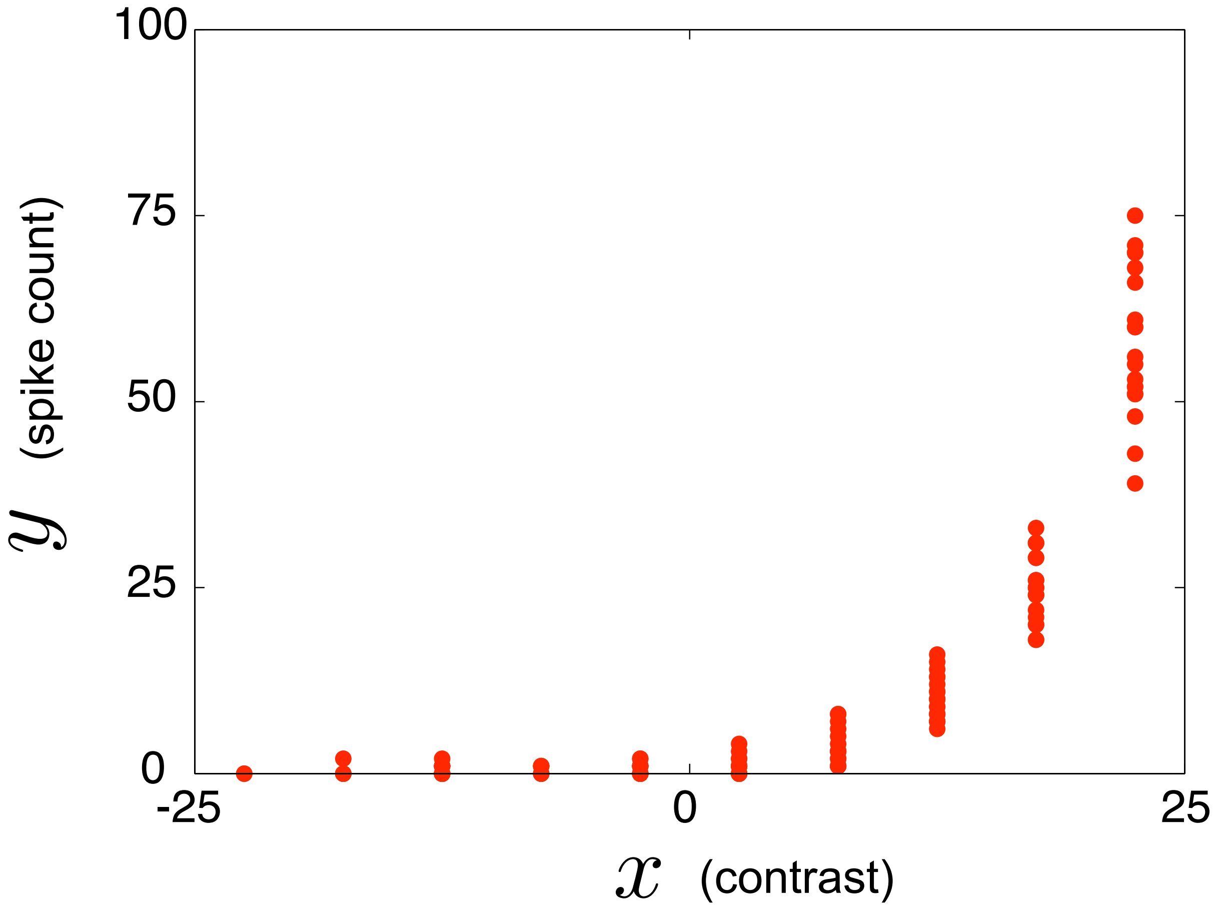
log-likelihood

# Example 3: unknown neuron



Be the computational neuroscientist: what model would you use?

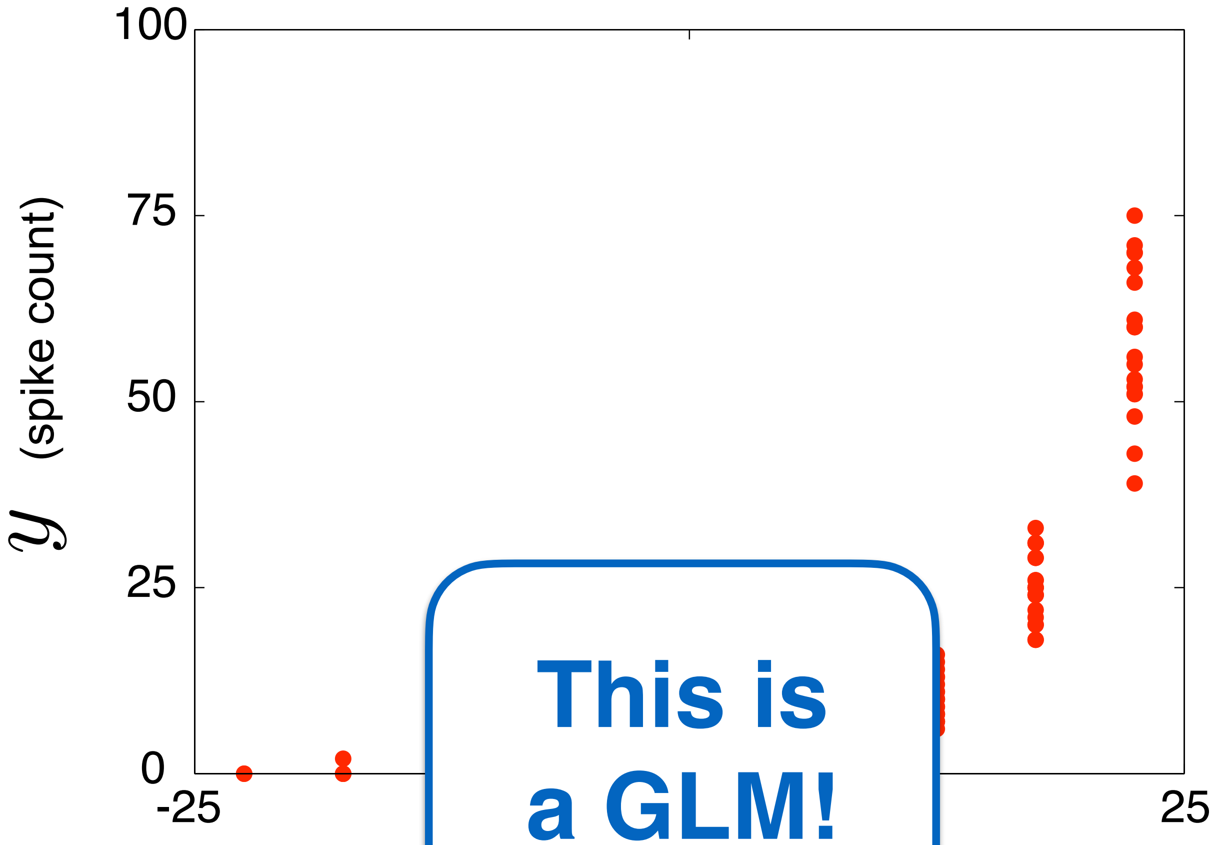
# Example 3: unknown neuron



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

$\lambda = f(\theta x)$, for some nonlinear function $f$

# Example 3: unknown neuron



**This is a GLM!**

More general setup:

$$y \sim Poiss(\lambda)$$

$$\lambda = f(\theta x)$$ , for some nonlinear function $f$

# Note on GLMs

- Be careful about terminology:

GLM ≠ GLM

General Linear Model

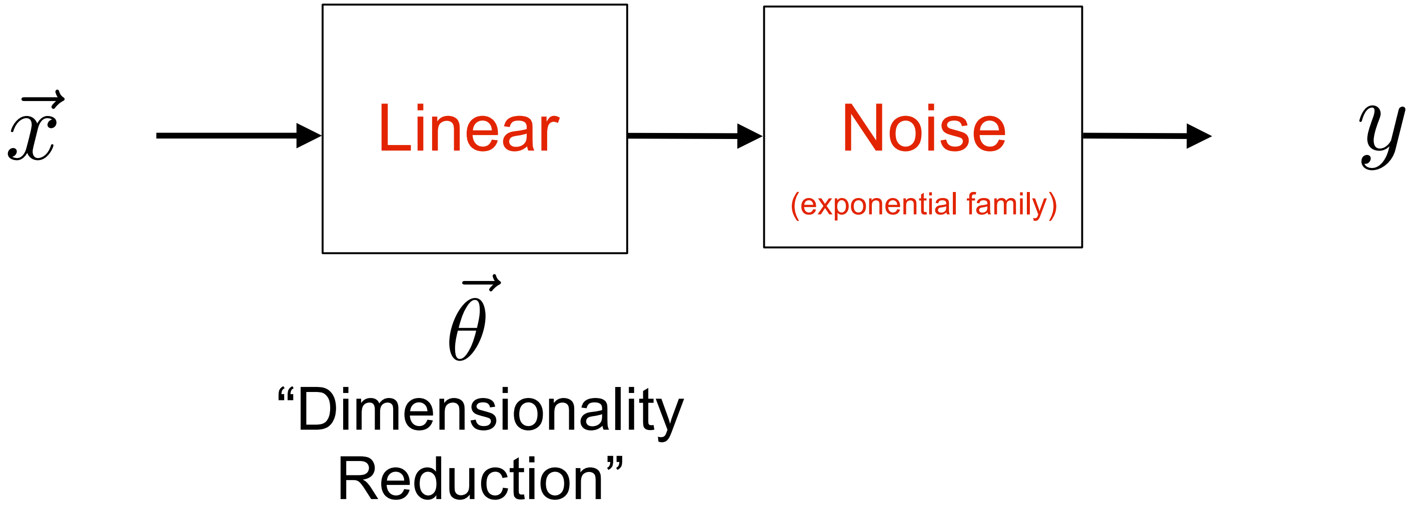
Linear

Generalized Linear Model

(Nelder 1972)

~~Linear~~

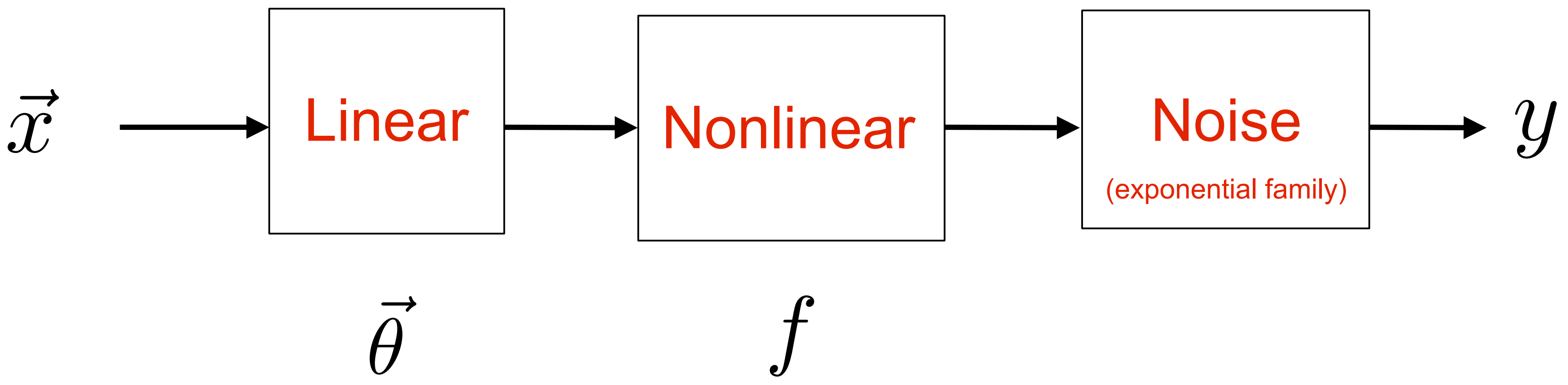
# 1. General Linear Model



Examples: 1. Gaussian $y = \vec{\theta} \cdot \vec{x} + \epsilon$

2. Poisson $y \sim \mathrm{Poiss}(\vec{\theta} \cdot \vec{x})$

# 2. Generalized Linear Model



Examples:

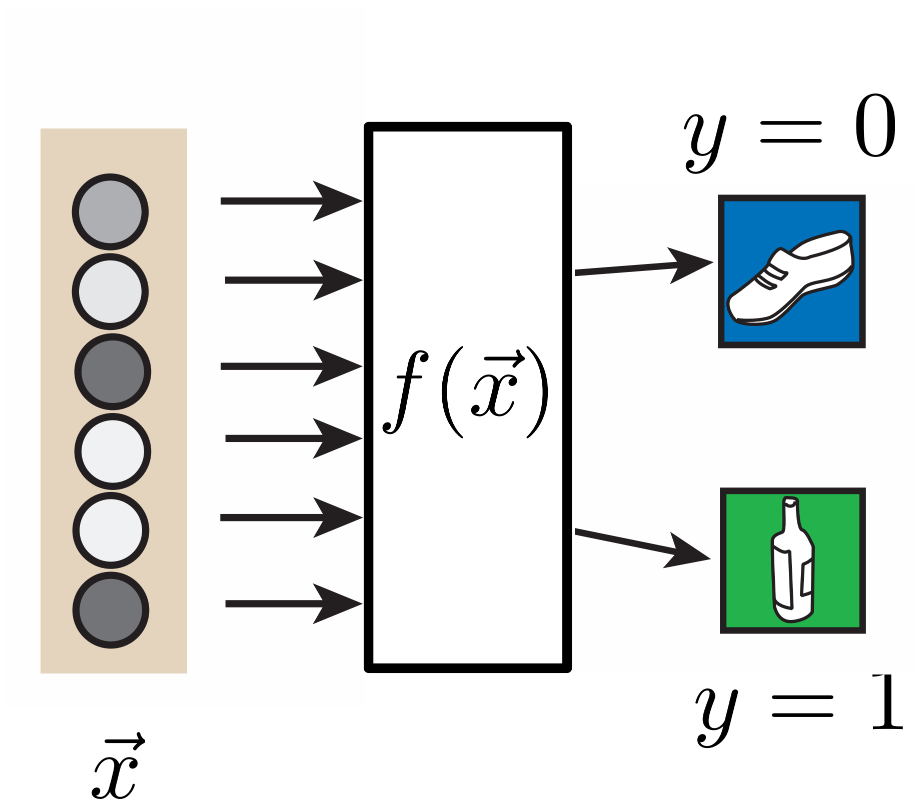
1. Gaussian $y = f(\vec{\theta} \cdot \vec{x}) + \epsilon$

2. Poisson $y \sim \mathrm{Poiss}(f(\vec{\theta} \cdot \vec{x}))$

# aside:
# Regression vs Classification

# Classification

- mapping from vector input to discrete category

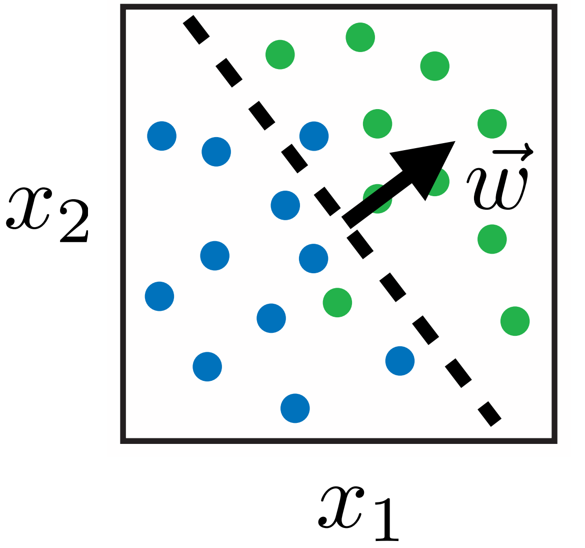


(voxel activity)

(spike counts)

linear classifier

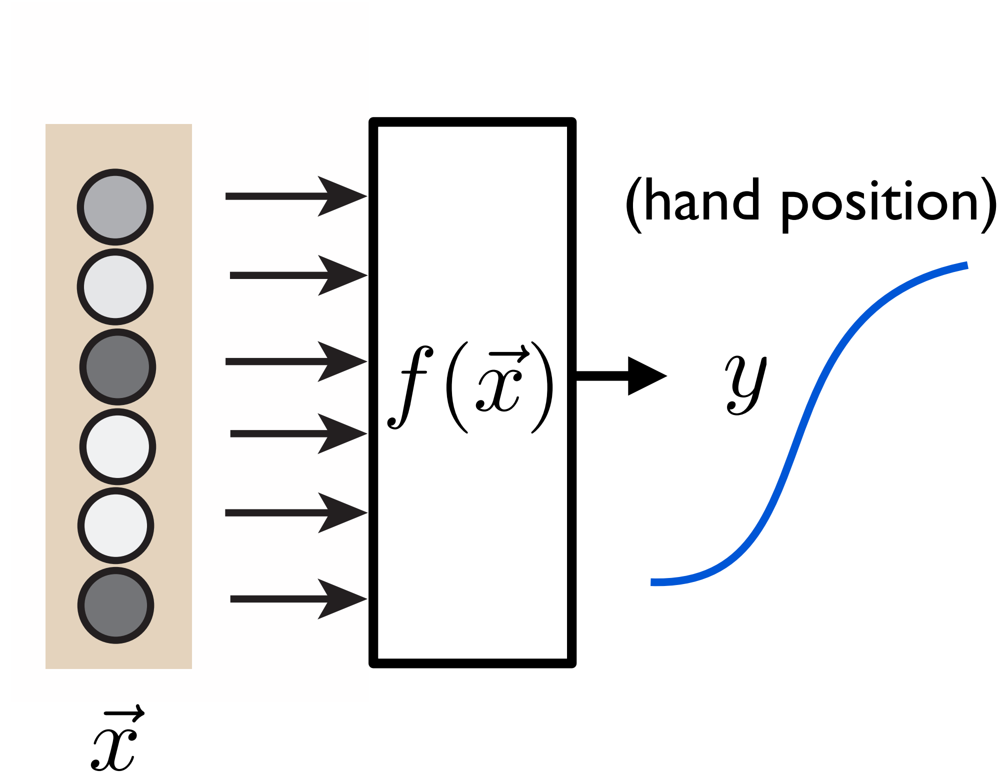
$$\vec{x} \cdot \vec{w} - b > 0$$



- linear perceptron
- Fisher linear discriminant
- support vector machine (SVM)

# Regression

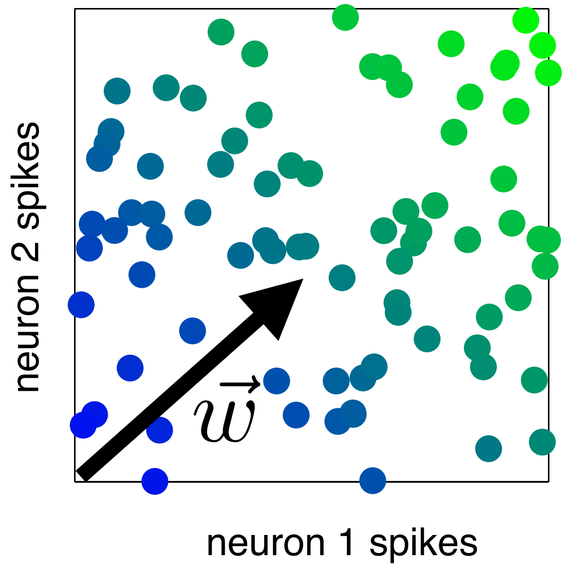
- output continuous instead of discrete



(hand position)

(neural activity)

$$y = \vec{x} \cdot \vec{w}$$

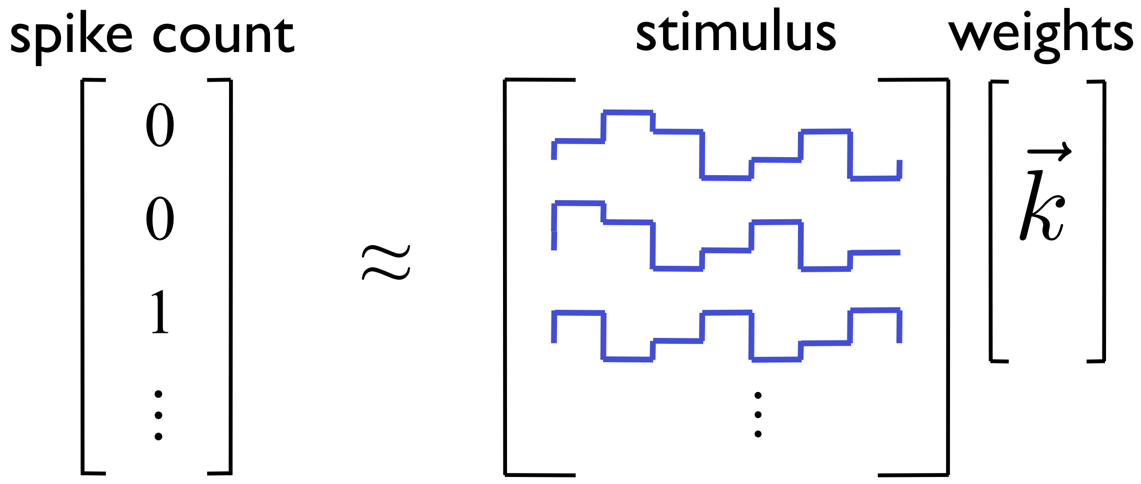


- can transform classification problems into regression problems (“logistic regression”):

probability of being in category

$$p(y=1) = f(\vec{x})$$

# GLM for binary responses



**Bernoulli GLM:**

(coin flipping model)

probability of spike at bin t

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

nonlinearity

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

$$p(y_t = 0|\vec{x}_t) = 1 - p_t$$

$$\boxed{p(y_t|\vec{x}_t) = p_t^{y_t}(1 - p_t)^{1-y_t}}$$

# GLM for binary responses

probability of spike at bin t

nonlinearity

**Bernoulli GLM:**

(coin flipping model)

$$p(y_t = 1|\vec{x}_t) = p_t \qquad\qquad p_t = f(\vec{x}_t \cdot \vec{k})$$

$$p(y_t = 0|\vec{x}_t) = 1 - p_t$$

$$\boxed{p(y_t|\vec{x}_t) = p_t^{y_t}(1-p_t)^{1-y_t}}$$

Equivalent ways of writing: $y_t|\vec{x}_t, \vec{k} \;\sim\; \mathrm{Ber}(f(\vec{x}_t \cdot \vec{k}))$

or $p(y_t|\vec{x}_t, \vec{k}) = f(\vec{x}_t \cdot \vec{k})^{y_t} \left(1 - f(\vec{x}_t \cdot \vec{k})\right)^{1-y_t}$

log-likelihood: $\mathcal{L} = \sum_{t=1}^{T} \Big( y_t \log f(\vec{x}_t \cdot \vec{k}) + (1-y_t)\log(1 - f(\vec{x}_t \cdot \vec{k})) \Big)$

in python:

```
L = np.sum( Y*np.log(f(X@k)) + (1-Y)*np.log(1-f(X@k)) )
```

# Logistic regression

**Bernoulli GLM:**

(coin flipping model)

probability of spike at bin t

nonlinearity

$$p(y_t = 1|\vec{x}_t) = p_t \qquad p_t = f(\vec{x}_t \cdot \vec{k})$$

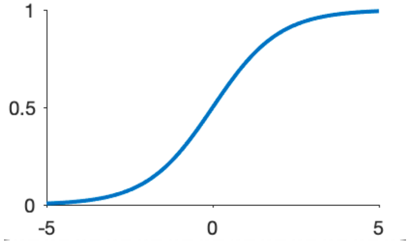
$$p(y_t = 0|\vec{x}_t) = 1 - p_t$$

$$\boxed{p(y_t|\vec{x}_t) = p_t^{y_t}(1-p_t)^{1-y_t}}$$

**Logistic regression:**

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

logistic function



- so logistic regression is a special case of a Bernoulli GLM, where the nonlinearity f(x) is a logistic function!

# Summary (last 3 lectures)

- Estimation
- Bias
- Variance
- Maximum Likelihood estimator
- MAP estimation: accounts for slow-speed bias in motion perception (Weiss, Simoncelli & Adelson 2002)
- General and Generalized Linear Models (GLMs)
- Bernoulli GLM / Logistic regression