Machine Learning Methods for Neural Data Analysis

Lecture 2: Introduction to probabilistic modeling

Scott Linderman

STATS 220/320 (NBIO220, CS339N). Winter 2023.

Probabilistic models

- This course is about probabilistic models — specifically probabilistic ***generative*** **models** —for neural data.
- The generative process is determined by model **parameters.**
- Our goal is to **estimate** or **infer** those parameters so that we can draw insight from them.
- More complex data needs more sophisticated models, but as we will see:
    - Rich models can be composed of simple building blocks.
    - The principles of estimation and inference remain the same.

Survey analysis

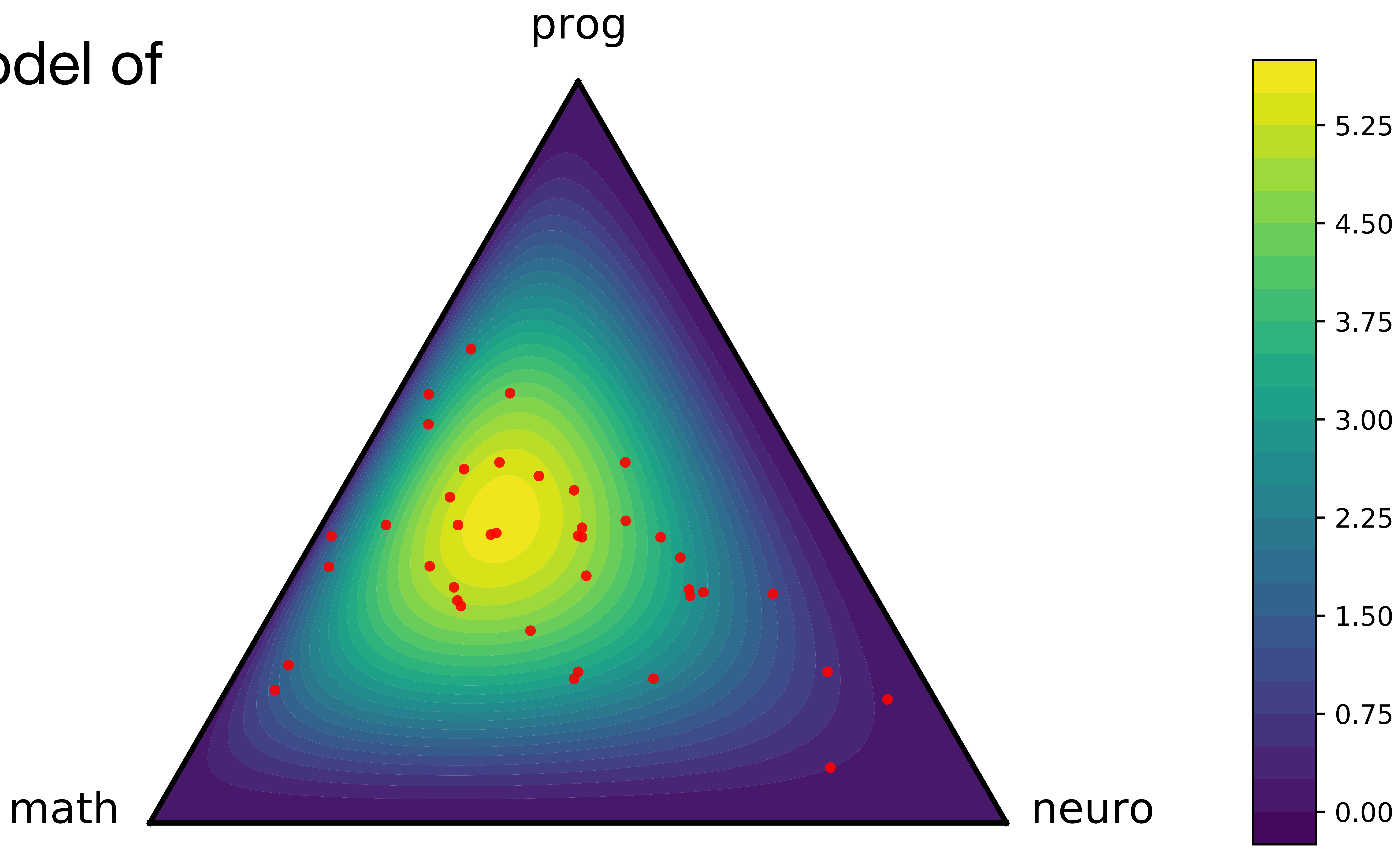
I constructed a probabilistic model of your survey responses. Let,

$$\boldsymbol{\pi}_n = \frac{1}{Z_n}\left[s_{n,1}, s_{n,2}, s_{n,3}\right]$$

$$Z_n = s_{n,1} + s_{n,2} + s_{n,3}$$

where $s_{n,k}$ is the your self-reported “skill” in subject k. It looks like a decent model is,

$$\boldsymbol{\pi}_n \sim \mathrm{Dir}([1.9, 2.6, 2.7])$$

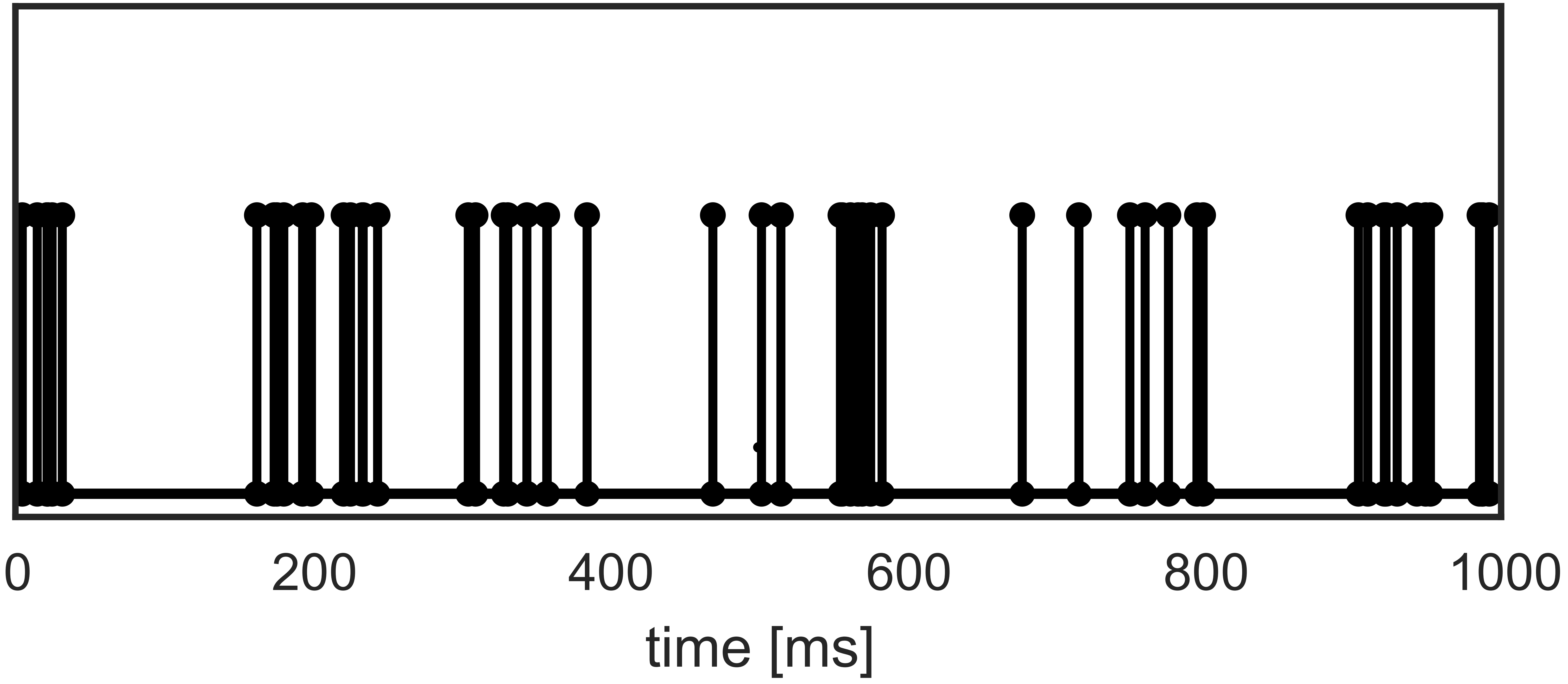


Survey analysis

- I went to high school where Ferris Bueller's Day Off was filmed.
- I can lick my elbow
- I enjoy trying out different kinds of European accents!
- I can't burp!
- Me and my sister would practice reciting the alphabet backwards because we heard it was a common DUI test
- I boulder V14 outdoor.
- I'd never traveled west of Toronto before starting Fall quarter at Stanford
- *And so many more! (I did not try to build a generative model for these responses.)*

Motivating example

- Consider the following 1s snippet of a (simulated) spike train. Do you see any interesting features? How might you start to model it?



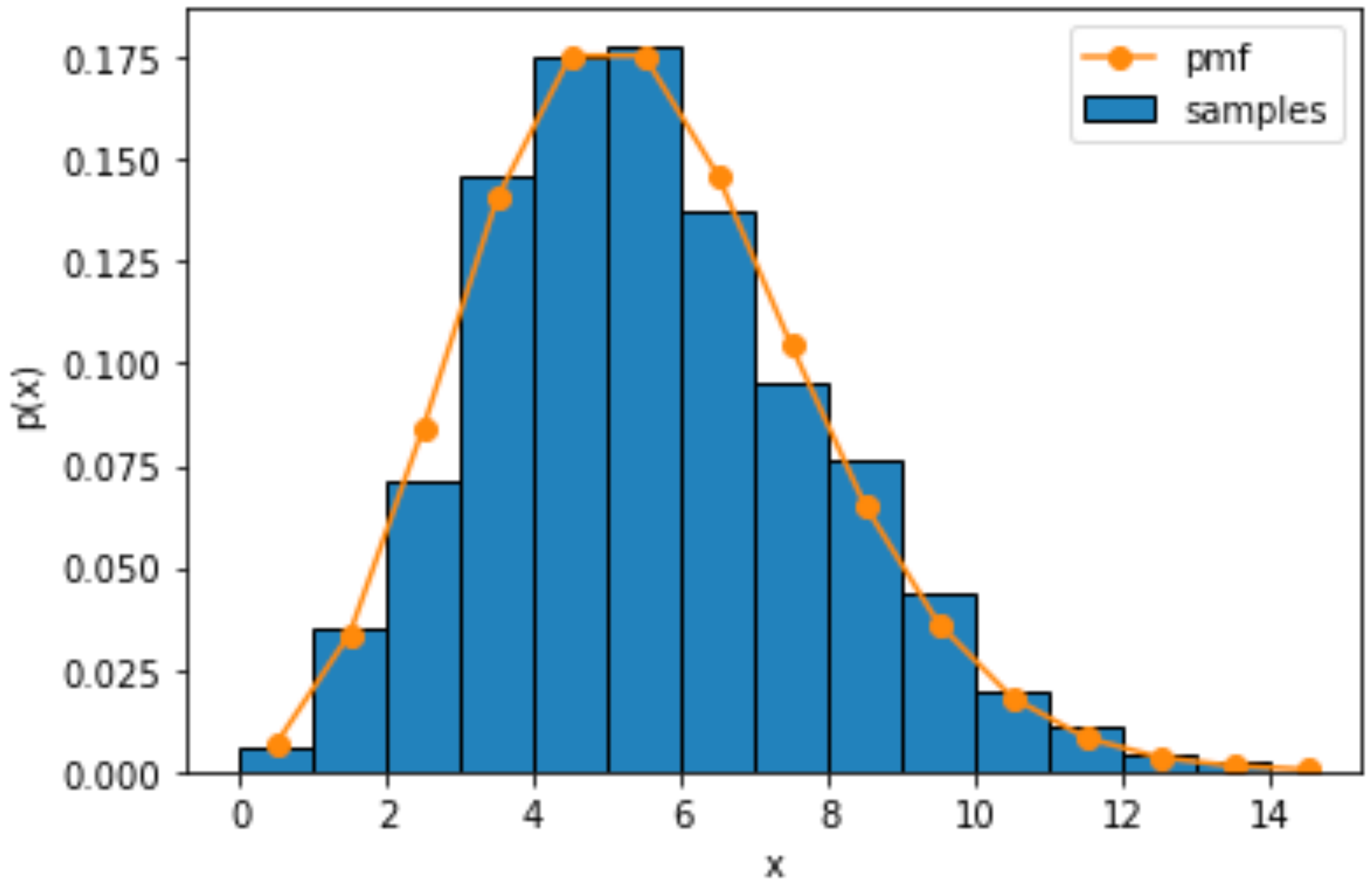
Simple model

Sampling a Poisson distribution

```
import torch
from torch.distributions import Poisson
import matplotlib.pyplot as plt

# Construct a Poisson distribution with rate 5.0 and draw 1000 samples
rate = 5.0
pois = Poisson(rate)
xs = pois.sample(sample_shape=(1000,))

# Plot a histogram of the samples and overlay the pmf
bins = torch.arange(15)
plt.hist(xs, bins, density=True, edgecolor='k', label='samples')
plt.plot(bins + .5, torch.exp(pois.log_prob(bins)), '-o', label='pmf')
plt.xlabel("x")
plt.ylabel("p(x)")
_ = plt.legend()
```



(This code is available on the course website.)

Fitting a Poisson distribution

Solving for the MLE

Adding a prior distribution on the rate

For example, this may not be the first neuron you've ever encountered. Maybe, based on your experience, you have a sense for the distribution of neural firing rates. That knowledge can be encoded in a **prior distribution**.

One common choice of prior on rates is the **gamma distribution**,

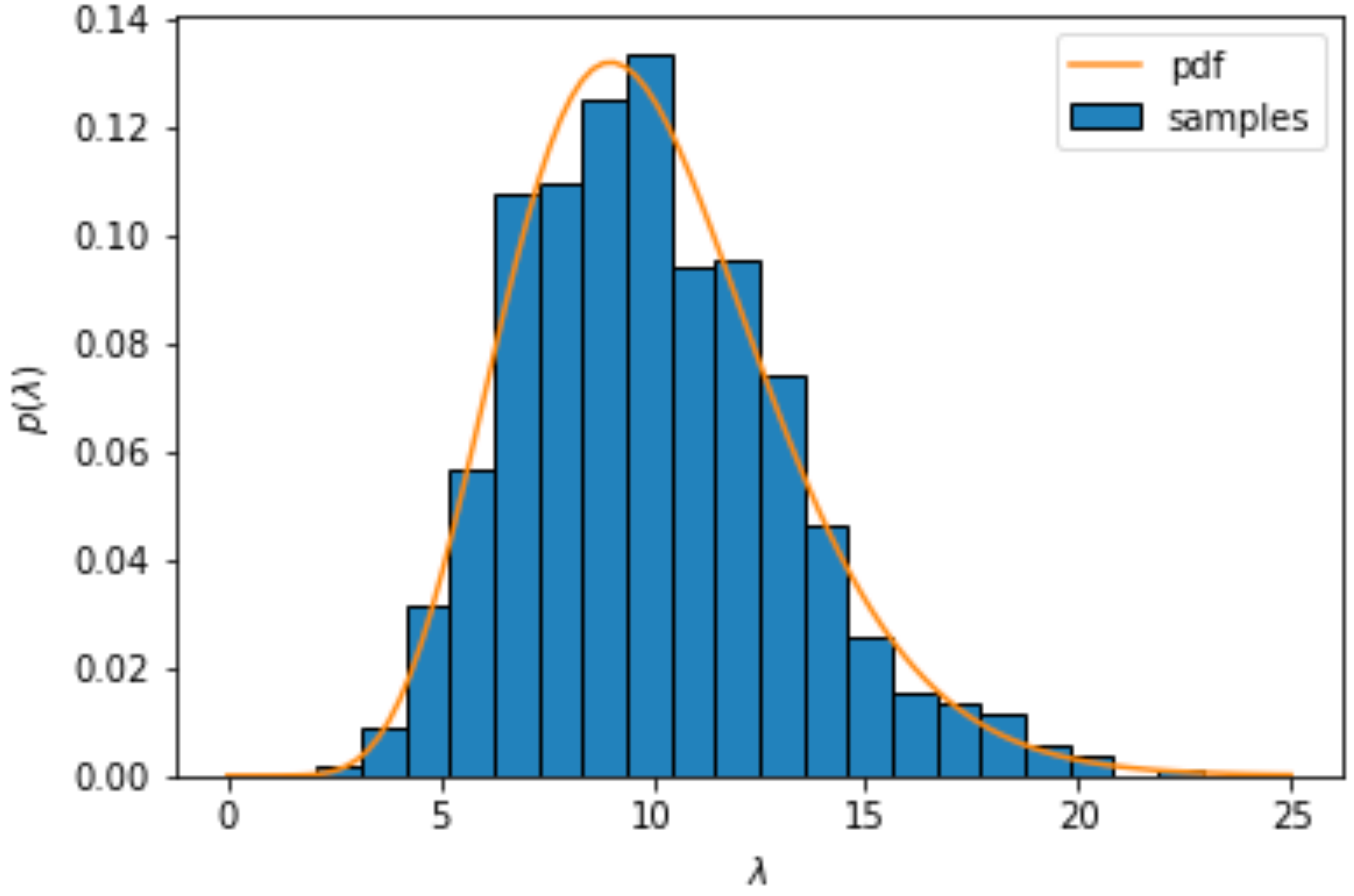
$$\lambda \sim \mathrm{Ga}(\alpha, \beta).$$

The gamma distribution has **support** for $\lambda \in \mathbb{R}_+$, and it is governed by two parameters:

- α, the **shape** or **concentration** parameter, and
- β, the **inverse scale** or **rate** parameter.

It's **probability density function (pdf)** is,

$$\mathrm{Ga}(\lambda; \alpha, \beta) = \frac{\beta^\alpha}{\Gamma(\alpha)} \lambda^{\alpha-1} e^{-\beta\lambda}.$$



“Fitting” the model with the prior

When we add in the prior distribution on λ, it becomes a random variable too. Now we have to consider the **joint distribution** of $\mathbf{x}$ and λ,

$$\begin{aligned} p(\mathbf{x}, \lambda) &= p(\mathbf{x} \mid \lambda)\, p(\lambda) \\ &= \left[\prod_{t=1}^{T} \mathrm{Pois}(x_t \mid \lambda)\right] \mathrm{Ga}(\lambda; \alpha, \beta) \end{aligned}$$

“Fitting” the model with the prior

When we add in the prior distribution on λ, it becomes a random variable too. Now we have to consider the **joint distribution** of $\mathbf{x}$ and λ,

$$
\begin{aligned}
p(\mathbf{x}, \lambda) &= p(\mathbf{x} \mid \lambda)\, p(\lambda) \\
&= \left[\prod_{t=1}^{T} \mathrm{Pois}(x_t \mid \lambda)\right] \mathrm{Ga}(\lambda; \alpha, \beta)
\end{aligned}
$$

The Product Rule, the Sum Rule, and Bayes’ Rule

In the first line we applied the **product rule** of probability, which says that we can rewrite a joint distribution as a product of a **marginal distribution** and a **conditional distribution**

$$p(x, y) = p(x)\, p(y \mid x).$$

The order doesn’t matter; we could alternatively write,

$$p(x, y) = p(y) p(x \mid y).$$

Product, Sum, and Bayes' Rule (continued)

The marginal distributions $p(x)$ and $p(y)$ are obtained via the **sum rule**,

$$p(x) = \sum_{y \in \mathcal{Y}} p(x, y)$$

where $\mathcal{Y}$ is the support of the random variable y.

Finally, putting both together, we obtain **Bayes' rule**,

$$p(x \mid y) = \frac{p(x, y)}{p(y)} = \frac{p(y \mid x)\, p(x)}{p(y)}.$$

Bayesian inference

We want to compute the **posterior distribution** of the rate λ *given* the observed spike counts $\mathbf{x}$ (and the prior parameters α and β, which are assumed fixed),

$$p(\lambda \mid \mathbf{x}).$$

Bayesian inference

We want to compute the **posterior distribution** of the rate λ *given* the observed spike counts $\mathbf{x}$ (and the prior parameters α and β, which are assumed fixed),

$$p(\lambda \mid \mathbf{x}).$$

By Bayes' rule (see box above), the posterior distribution is equal to the ratio of the **joint distribution** over the **marginal distribution**,

$$p(\lambda \mid \mathbf{x}) = \frac{p(\mathbf{x}, \lambda)}{p(\mathbf{x})}.$$

Bayesian inference

We want to compute the **posterior distribution** of the rate λ *given* the observed spike counts $\mathbf{x}$ (and the prior parameters α and β, which are assumed fixed),

$$p(\lambda \mid \mathbf{x}).$$

By Bayes' rule (see box above), the posterior distribution is equal to the ratio of the **joint distribution** over the **marginal distribution**,

$$p(\lambda \mid \mathbf{x}) = \frac{p(\mathbf{x}, \lambda)}{p(\mathbf{x})}.$$

Note that the denominator (the marginal distribution) does not depend on λ, so the posterior is proportional to the joint,

$$p(\lambda \mid \mathbf{x}) \propto p(\mathbf{x}, \lambda).$$

Maximum *a posteriori* (MAP) estimation

A simple summary of the posterior distribution is its **mode** — the point(s) where the pdf is maximized,

$$\lambda_{\mathsf{MAP}} = \arg\max p(\lambda \mid \mathbf{x}).$$

or equivalently,

$$\lambda_{\mathsf{MAP}} = \arg\max p(\lambda, \mathbf{x}).$$

since the posterior is proportional to the joint.

> **Warning**
>
> True Bayesians cringe at MAP estimation! *How can a single point (the mode) summarize an entire distribution!?* It can't, but we'll use it for now and be better Bayesians later in the course.

Conjugate priors

Now let's go back and expand the Poisson pmf and the gamma pdf in the joint distribution,

$$p(\lambda \mid \mathbf{x}) \propto p(\mathbf{x}, \lambda)$$

Solving for the MAP estimate

How do we solve for the MAP estimate, $\lambda_{\text{MAP}} = \arg\max p(\lambda \mid \mathbf{x})$?

Now that you know the posterior is a gamma distribution, you can expand its pdf, take the log, take the derivative wrt λ, set it to zero and solve.

Or you can just go to the Wikipedia page on the gamma distribution and see that its mode is

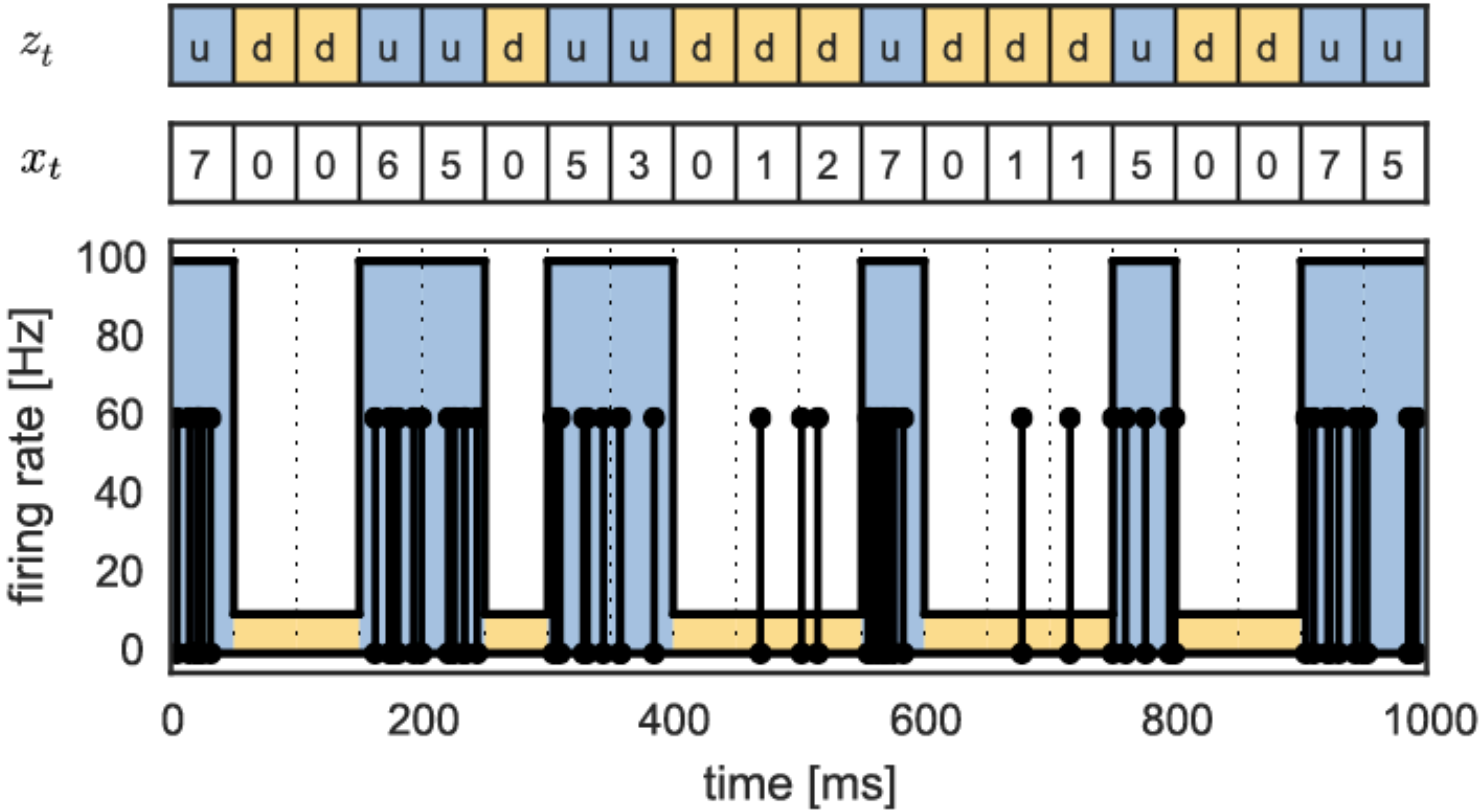
$$\lambda_{\text{MAP}} = \frac{\alpha' - 1}{\beta'} = \frac{\alpha - 1 + \sum_{t=1}^{T} x_t}{\beta + T}$$

for $\alpha' \geq 1$, and 0 otherwise.

Uninformative priors: under what prior is the MAP estimate the same as the MLE?

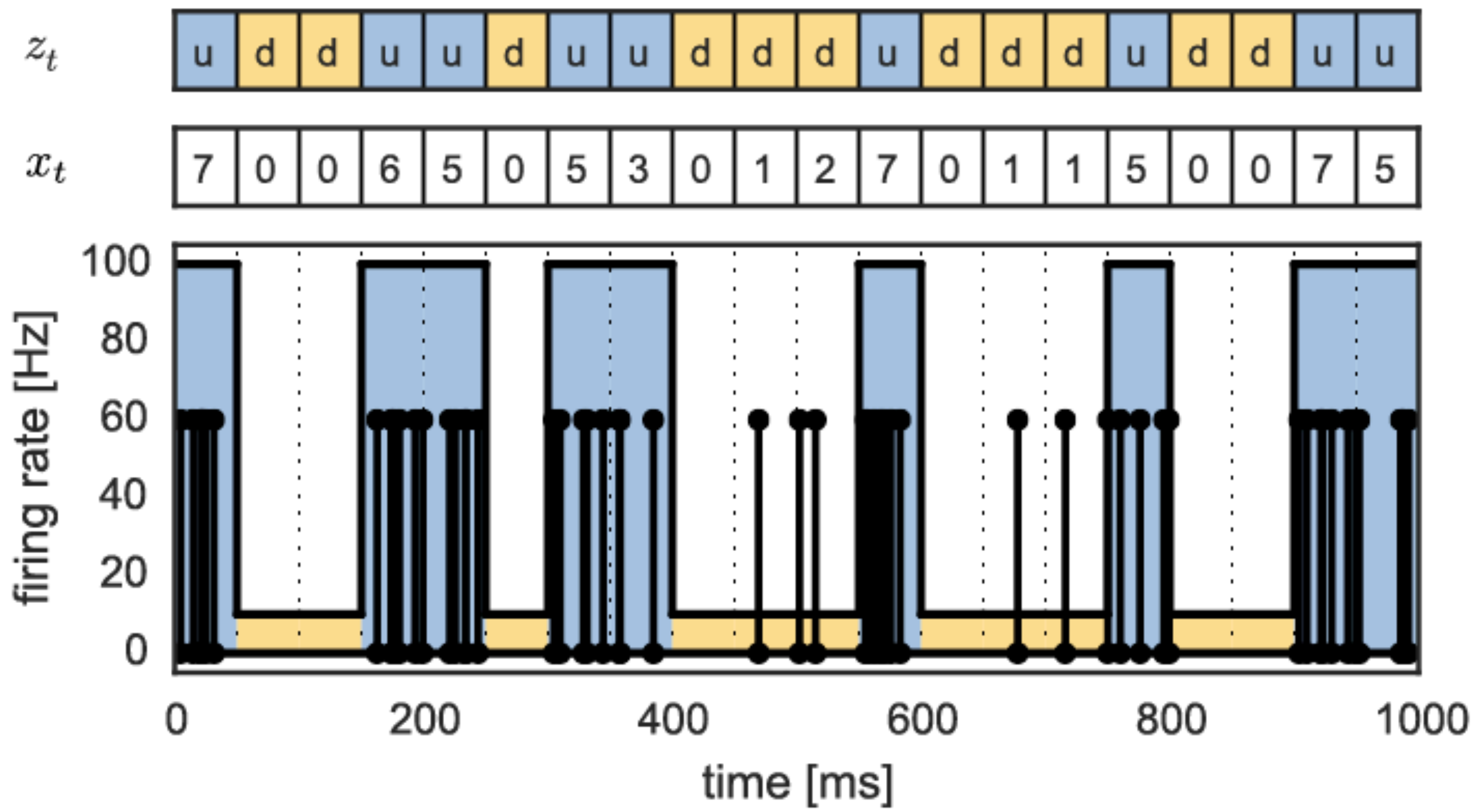
Mixture models and latent variables

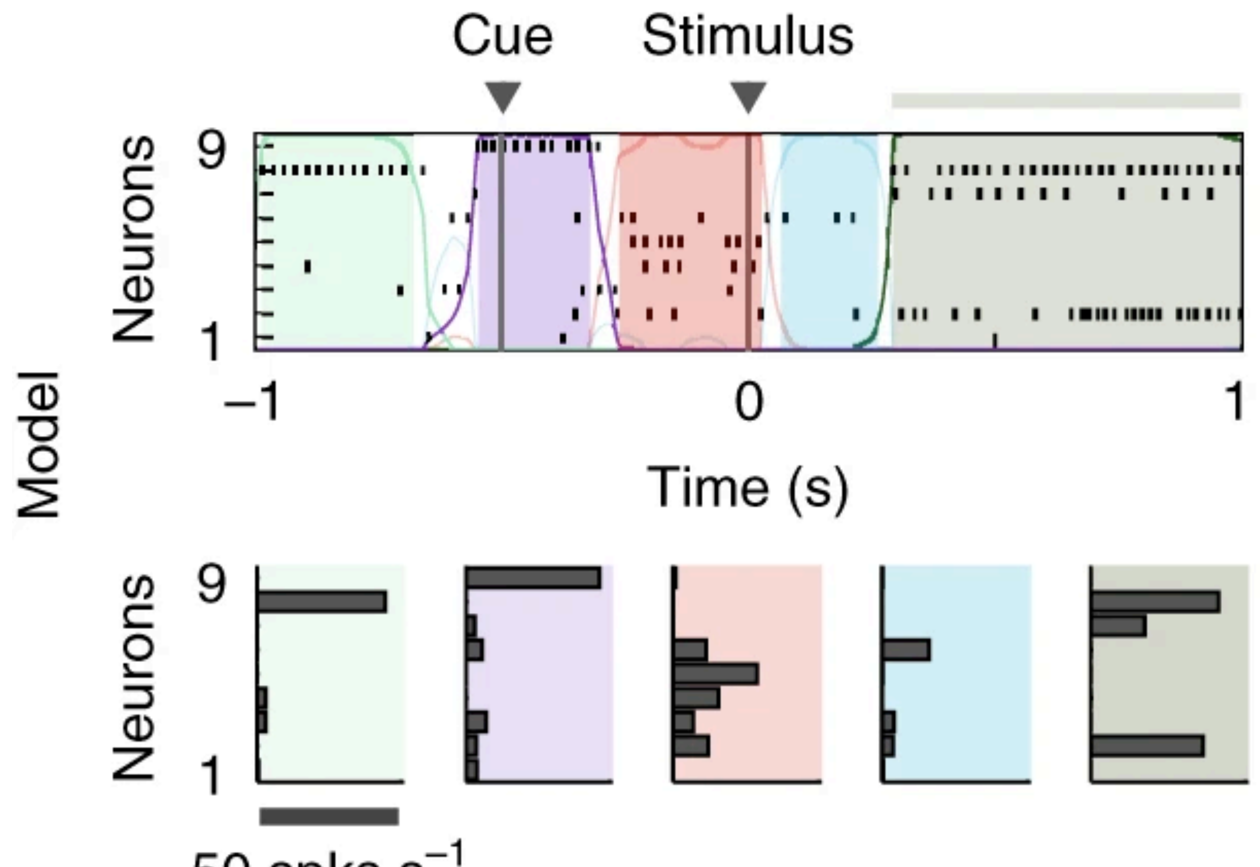
Real data is rarely so simple! One way to build richer models is via **latent variables**.



Mixture models and latent variables

- Let $z_t \in \{0,1\}$ be the *latent state:*
  - E.g. high firing ("up") and low firing ("down") states.
  - Sequences of "coding states" in gustatory cortex.
- Each state has its own firing rate.
- Our goal is to infer these states given only the spike trains.





Mazucatto et al (Nat. Neuro., 2019)

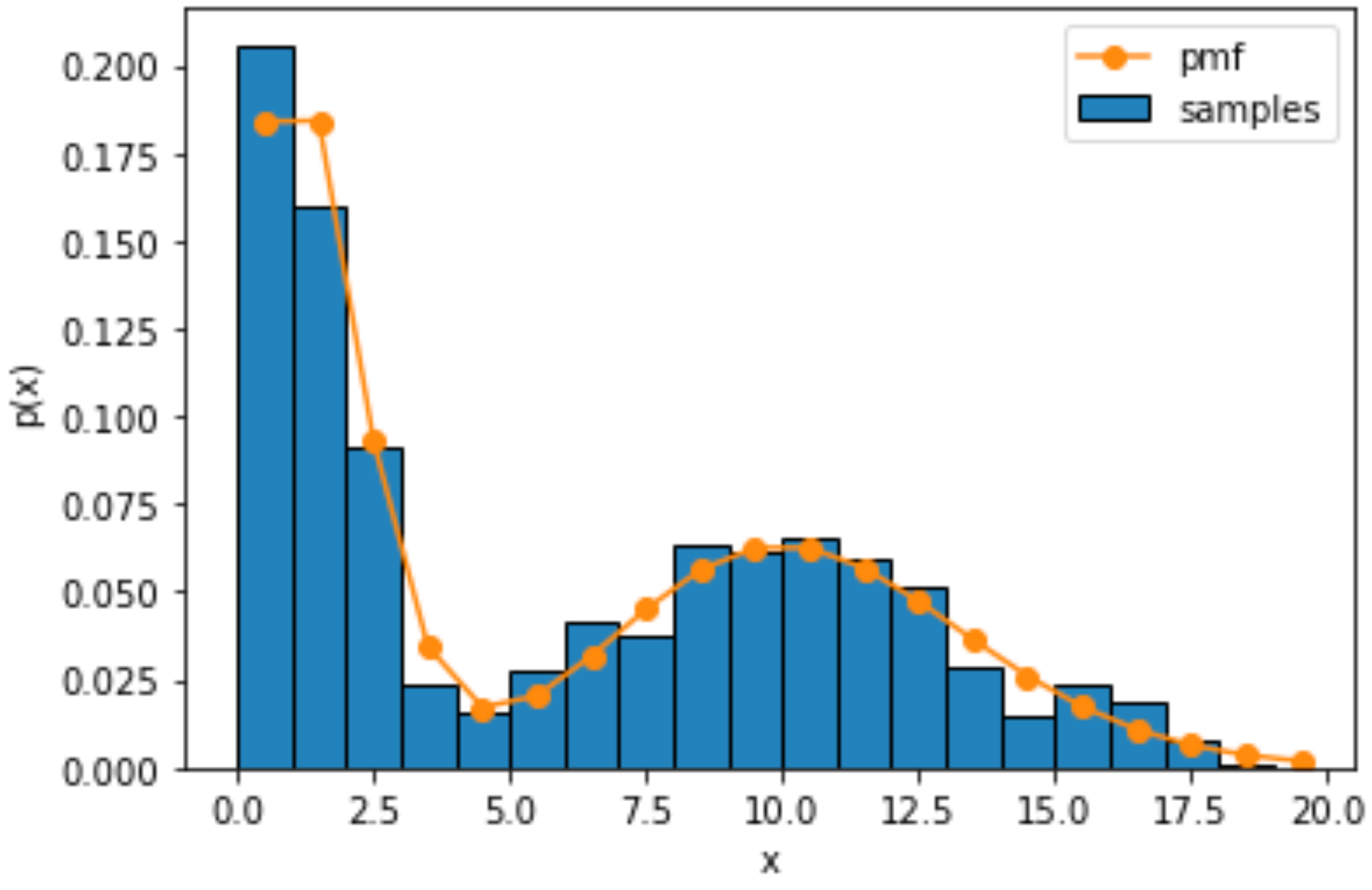
A Poisson mixture model

Finally, assume that the latent variables are equally probable and independent across time. Formally, we can write that as a **categorical distribution** with equal probabilities for both states,

$$z_t \sim \mathrm{Cat}([\tfrac{1}{2}, \tfrac{1}{2}]).$$

The resulting model is called a **mixture model** because the marginal distribution, $p(x_t \mid \boldsymbol{\lambda})$ where $\boldsymbol{\lambda} = (\lambda_0, \lambda_1)$, is a mixture of two Poisson distributions,

$$\begin{aligned} p(x_t \mid \boldsymbol{\lambda}) &= \sum_{z_t \in \{0,1\}} p(x_t, z_t \mid \boldsymbol{\lambda}) \\ &= \sum_{z_t \in \{0,1\}} p(x_t \mid z_t, \boldsymbol{\lambda})\, p(z_t) \\ &= \frac{1}{2}\mathrm{Pois}(x_t \mid \lambda_0) + \frac{1}{2}\mathrm{Pois}(x_t \mid \lambda_1) \end{aligned}$$



Fitting a mixture model by coordinate ascent

Conceptually, fitting the mixture model is no different than fitting the the simple Poisson model above.

We will perform MAP estimation to find,

$$\mathbf{z}_{\mathsf{MAP}}, \boldsymbol{\lambda}_{\mathsf{MAP}} = \arg\max p(\mathbf{z}, \boldsymbol{\lambda} \mid \mathbf{x})$$

where $\mathbf{z} = (z_1, \ldots, z_T)$. Again, this is equivalent to maximizing the joint probability.

Fitting a mixture model by coordinate ascent

Conceptually, fitting the mixture model is no different than fitting the the simple Poisson model above.

We will perform MAP estimation to find,

$$\mathbf{z}_{\mathsf{MAP}}, \boldsymbol{\lambda}_{\mathsf{MAP}} = \arg\max p(\mathbf{z}, \boldsymbol{\lambda} \mid \mathbf{x})$$

where $\mathbf{z} = (z_1, \ldots, z_T)$. Again, this is equivalent to maximizing the joint probability.

Expanding the joint distribution over spike counts, latent variables, and rates,

$$\begin{aligned} p(\mathbf{x}, \mathbf{z}, \boldsymbol{\lambda}) &= \left[\prod_{t=1}^{T} p(x_t \mid z_t, \boldsymbol{\lambda})\, p(z_t)\right] p(\boldsymbol{\lambda}) \\ &= \left[\prod_{t=1}^{T} \mathrm{Pois}(x_t \mid \lambda_{z_t}) \times \frac{1}{2}\right] \mathrm{Ga}(\lambda_0; \alpha, \beta)\, \mathrm{Ga}(\lambda_1; \alpha, \beta) \end{aligned}$$

Fitting a mixture model by coordinate ascent

Fixing the rates, the most likely state at time t is,

$$z_t = \begin{cases} 1 & \text{if } \; \mathrm{Pois}(x_t \mid \lambda_1) \geq \mathrm{Pois}(x_t \mid \lambda_0) \\ 0 & \text{otherwise} \end{cases}$$

Fitting a mixture model by coordinate ascent

Conclusion

This chapter introduced the basics of probabilistic modeling:

- We encountered 3 common distributions: **Poisson, gamma, and categorical.**

- We learned how to construct **joint distributions** using the **product rule**, how to compute **marginal distributions** with the **sum rule**, and how to find the **posterior distribution** with **Bayes' rule**.

- We learned about **maximum likelihood estimation (MLE)** and **maximum *a posteriori* (MAP)** estimation.

- We encountered **conjugate priors** where the posterior distribution is in the same family, making calculations particularly simple.

- Finally, we learned how to construct **more flexible models** by introducing **latent variables**, and how to perform MAP estimation in those models using **coordinate ascent**.

Further Reading

There are many great references on probabilistic modeling. I like:

- Ch 2.1 and 2.2 of [Murphy, 2023]
- Ch 1.2 of [Bishop, 2006]
- See references on the course website.
- **Next time: Basic Neurobio and Simple Spike Sorting!**