Machine Learning Methods for Neural Data Analysis

Lecture 1: Introduction

Scott Linderman

STATS 220/320 (NBIO220, CS339N). Winter 2021.

Welcome!

Tell us about yourself:

https://tinyurl.com/stats320

Introductions

- About me:
  - Asst. Prof. of Statistics and Computer Science (by cou
  - Institute Scholar, Wu Tsai Neurosciences Institute
  - Pandemic hobby: building and refinishing tables
- TA's:
  - Han Wu (3rd year PhD student in Statistics)
  - Jaime Roquero Gimenez (5th year PhD student in Statistics)

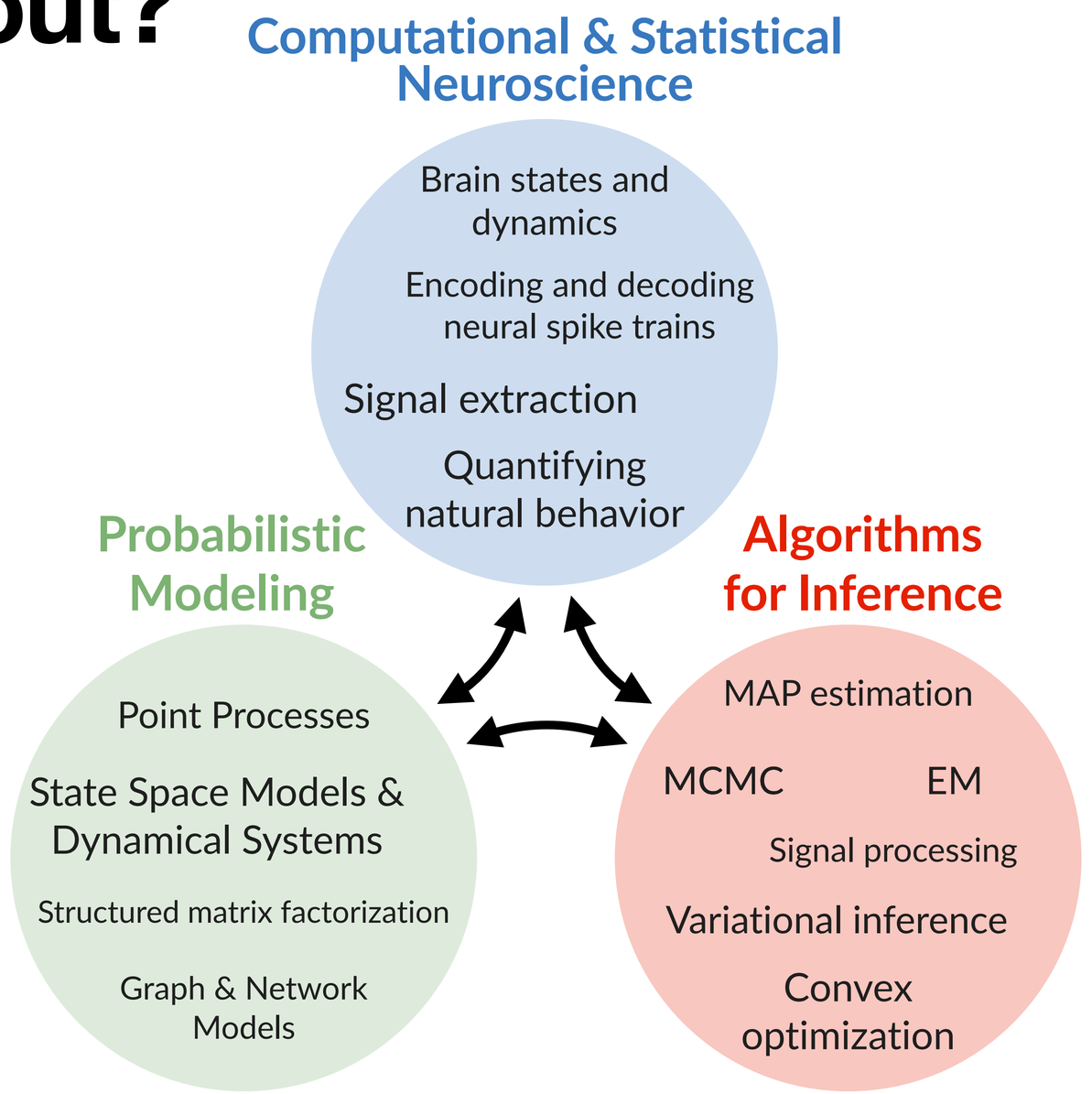
Please take this survey: https://tinyurl.com/stats320

Agenda

1. What is this course about?
2. Logistics
3. Technical primer

What is this course about?

- Understanding key challenges in computational neuroscience and neural data analysis.
- Formalizing these challenges with probabilistic models.
- Developing algorithms for statistical inference in these models.
- Implementing these algorithms in Python and applying them to data.



Please take this survey: https://tinyurl.com/stats320

Course Outline

- Unit 1: Extracting signals of interest from raw data
- Unit 2: Encoding and decoding neural spike trains
- Unit 3: Latent variable models of neural and behavioral data

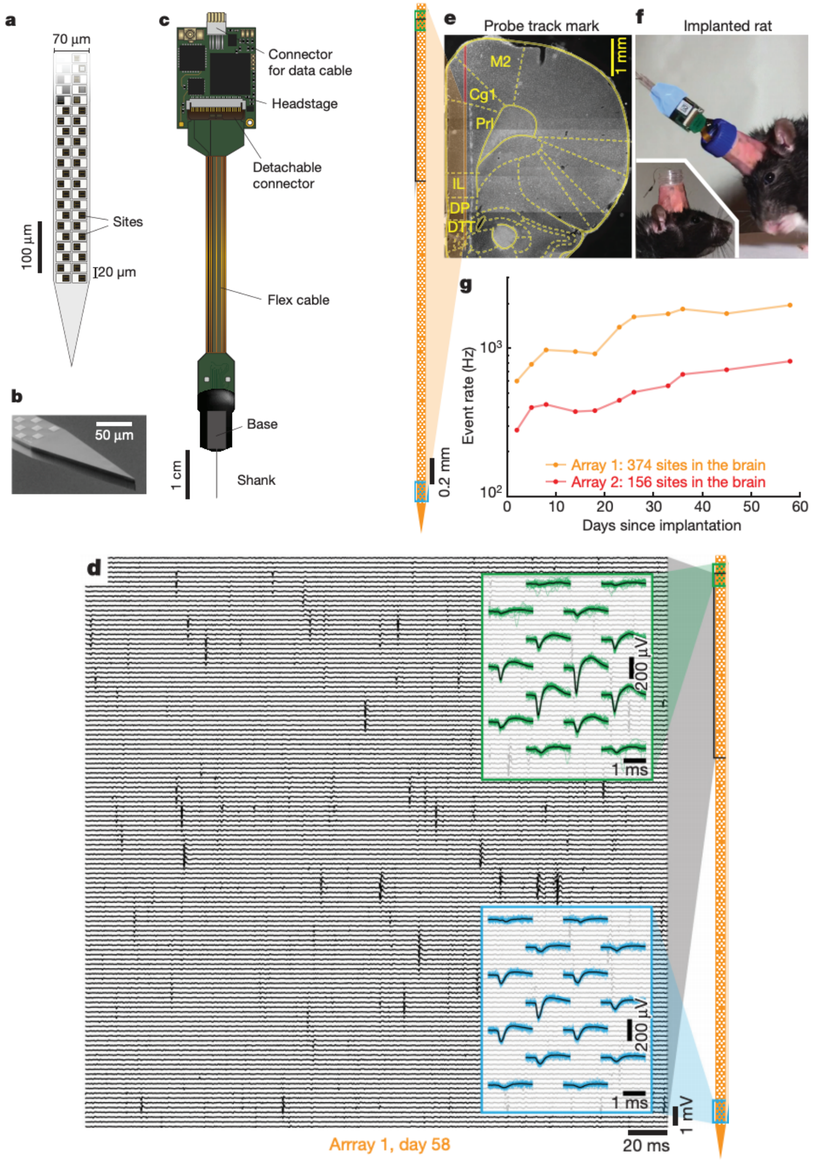
- Syllabus on Canvas (and open for shopping)

Please take this survey: https://tinyurl.com/stats320

Unit 1: Signal Extraction

Spike Sorting

- Modern recording probes like **Neuropixels** measure the electrical activity of **hundreds of cells** across **multiple brain regions** simultaneously.
- The raw data is a **multidimensional time series** of **voltage measurements**, one for each recording site on the probe.
- When neurons near the probe fire an **action potential**, it registers a **spike in the voltage** on nearby channels.
- Our goal is to **find the spikes** in this time series and **assign a neuron label** based on its waveform.



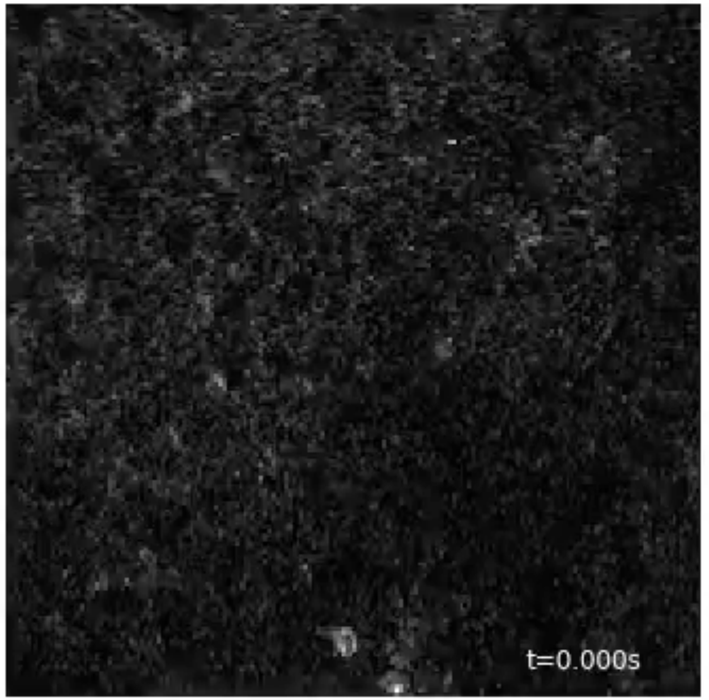
Jun et al, 2017.

Please take this survey: https://tinyurl.com/stats320

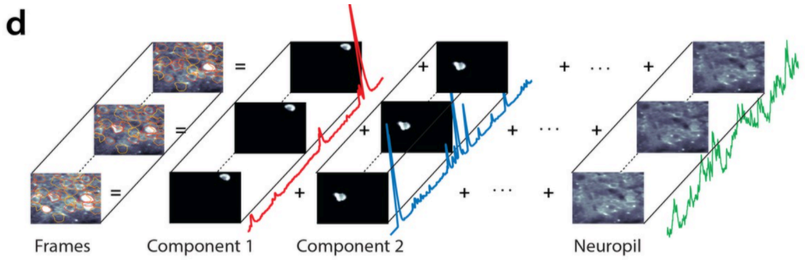
Unit 1: Signal Extraction

Demixing calcium imaging data

- When neurons spike, there's a large influx of **calcium ions (Ca^{2+})** into the cell.
- **Genetically encoded calcium indicators** (GECIs) bind to calcium ions, and when light is shone on them they fluoresce.
- Using these indicators, neuroscientists can **optically record** calcium concentrations, a good proxy for neural spiking.
- **Demixing videos to identify cells** and **deconvolving traces to identify spikes** is an area of active research.



Data from Sue Ann Koay and David Tank



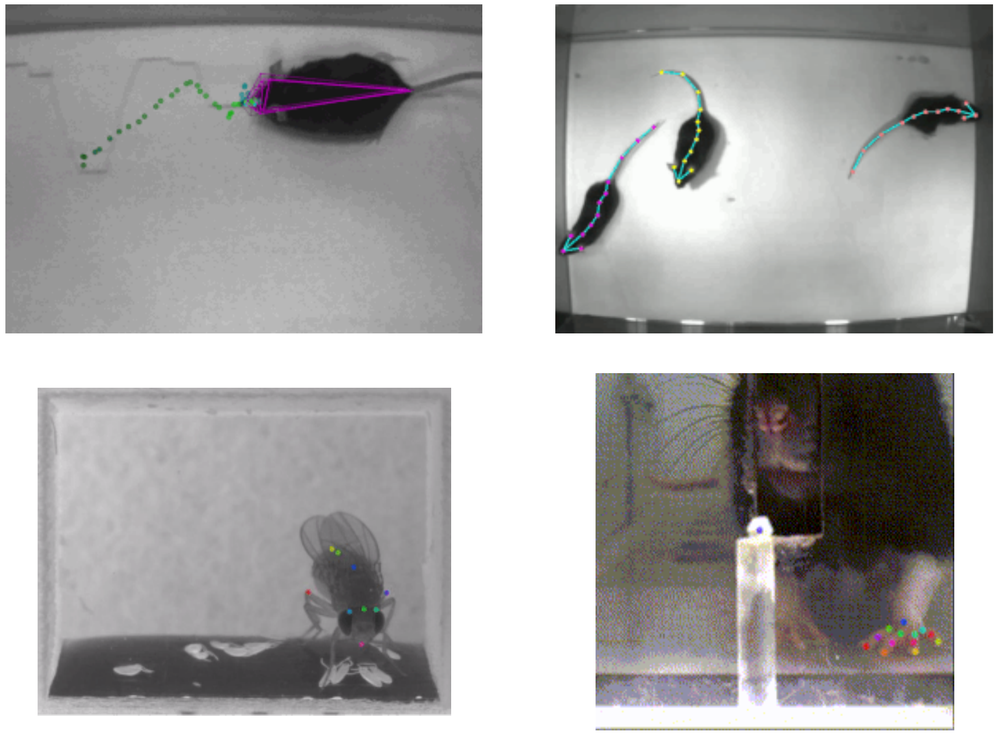
Giovanucci et al (eLife, 2019)

Please take this survey: https://tinyurl.com/stats320

Unit 1: Signal Extraction

Markerless pose tracking

- We want to understand how neural activity produces behavior.
- First, we need to **quantify motor outputs**, ideally in unconstrained animals.
- State of the art methods for **markerless pose tracking** use **deep convolutional neural networks (CNNs)** to find keypoints in videos.



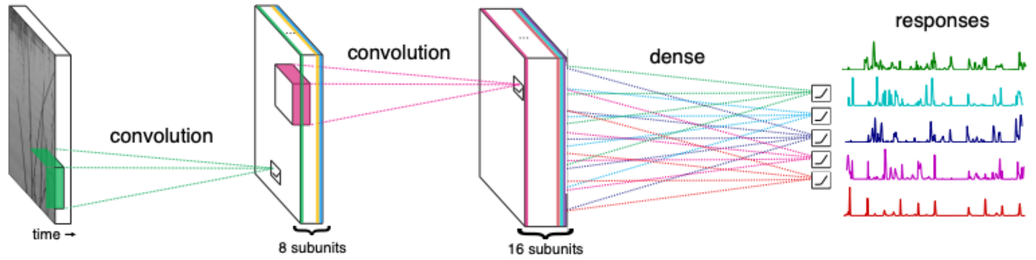
Mathis et al (Nat. Neuro., 2018)
https://github.com/DeepLabCut/DeepLabCut

Please take this survey: https://tinyurl.com/stats320

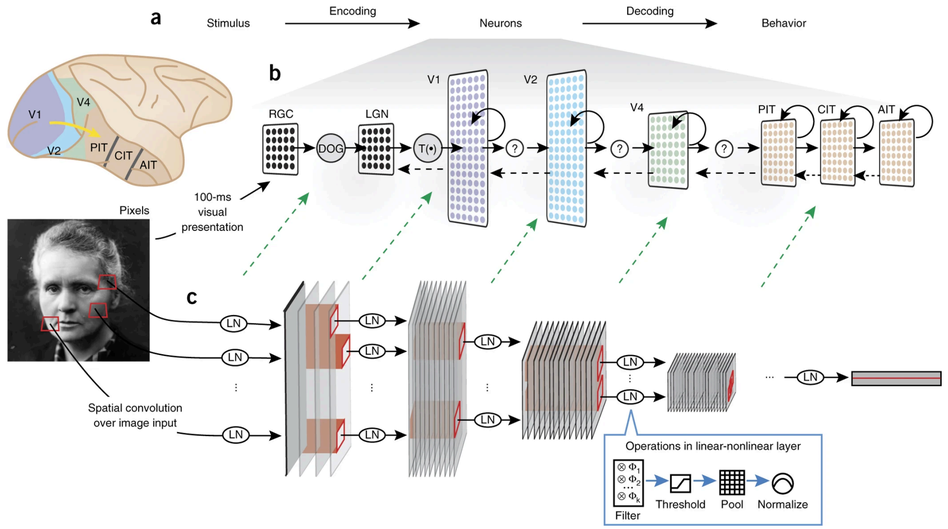
Unit 2: Encoding and Decoding Neural Spike Trains

Predicting neural responses to images

- CNNs aren't just useful for signal extraction, they're also our best models for how the **visual system encodes sensory inputs.**
- Of course, we see a constantly changing visual scene. We'll build models that **take in movies** and **output neural firing rates.**
- Neural spikes are modeled as draws from a **Poisson process** with these firing rates.



McIntosh et al (NeurIPS 2016)



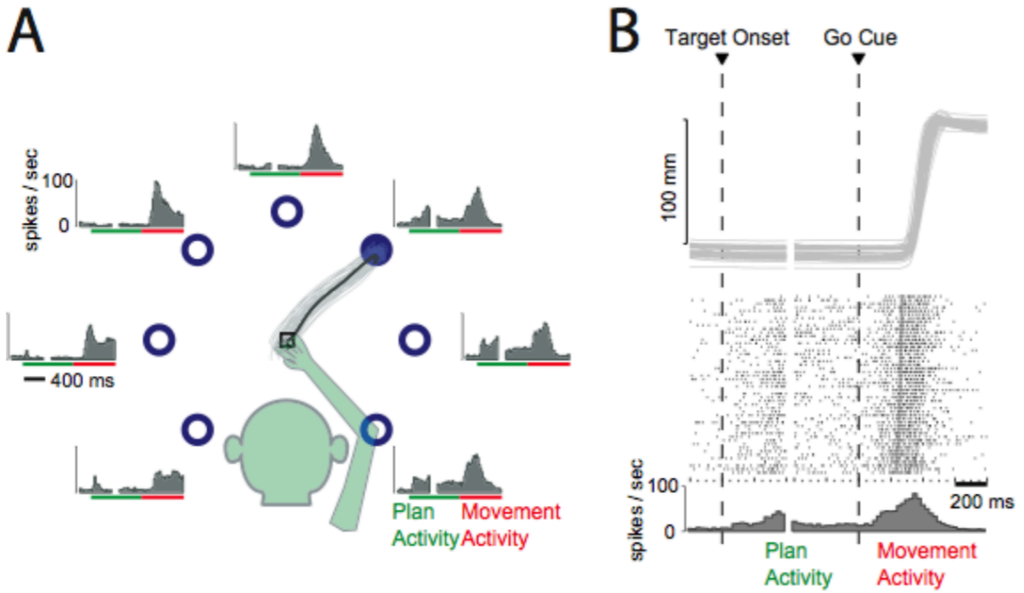
Yamins and DiCarlo (Nat. Neuro. 2016)

Please take this survey: https://tinyurl.com/stats320

Unit 2: Encoding and Decoding Neural Spike Trains

Decoding arm movements from neural data

- We also want to understand how to **decode motor outputs from neural activity**.
- This is a central challenge in **building neural prostheses**.
- Neurons in motor cortex, in particular, fire at different rates for different movements.
- We can leverage these differences to **infer movements** from **neural data.**



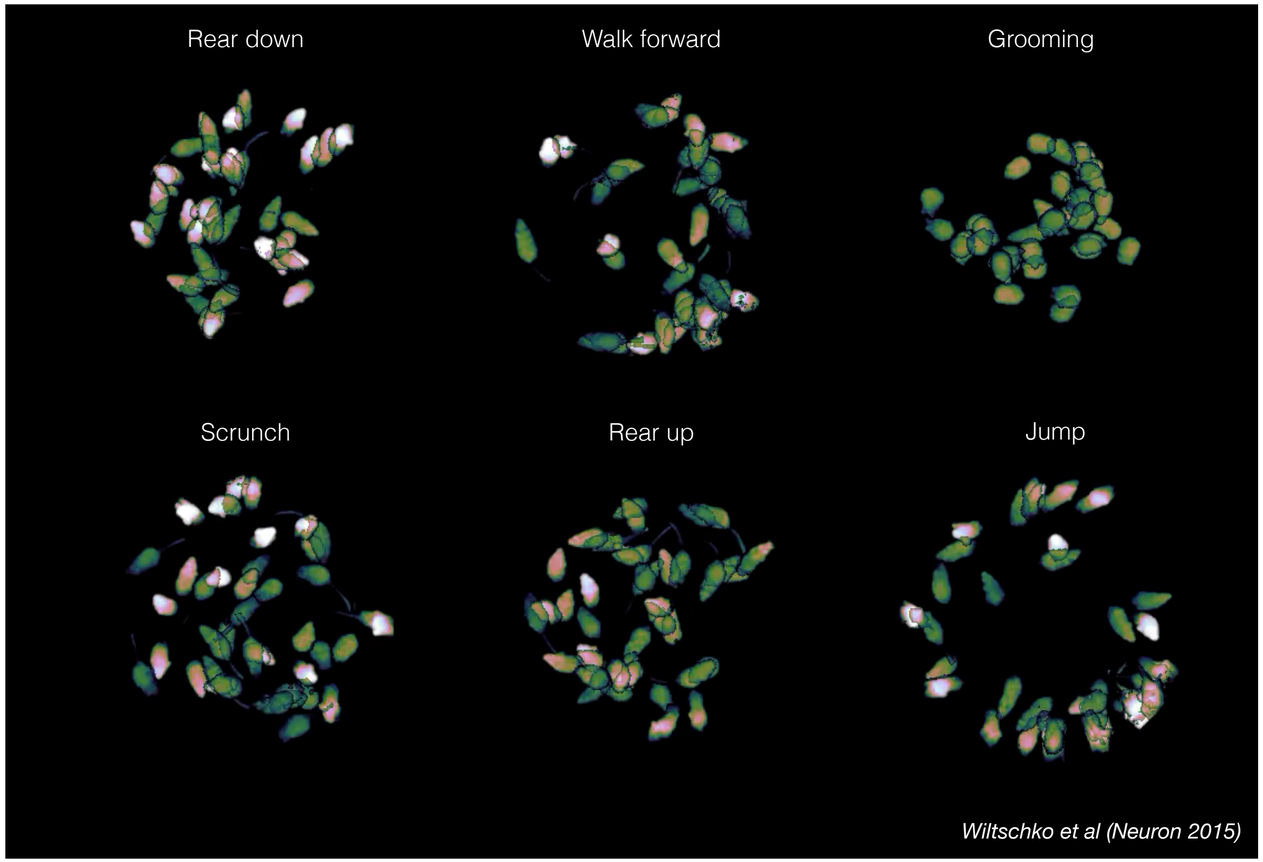
Prof. Krishna Shenoy, EE124

Please take this survey: https://tinyurl.com/stats320

Unit 3: Latent variable models of neural and behavioral data

Summarizing behavior with movement “syllables”

- We can learn a lot about the brain by understanding the **structure of its outputs**.
- Recently, there’s been a “**call to action**” to better characterize animal behavior.
  - Krakauer et al (Neuron, 2017); Datta et al. (Neuron, 2019)
- **Latent variable models** offer a compelling means of **summarizing behavior** in terms of **hidden states**, or “syllables,” of movement.
- We’ll build **autoregressive hidden Markov models** to extract such syllables from video data.

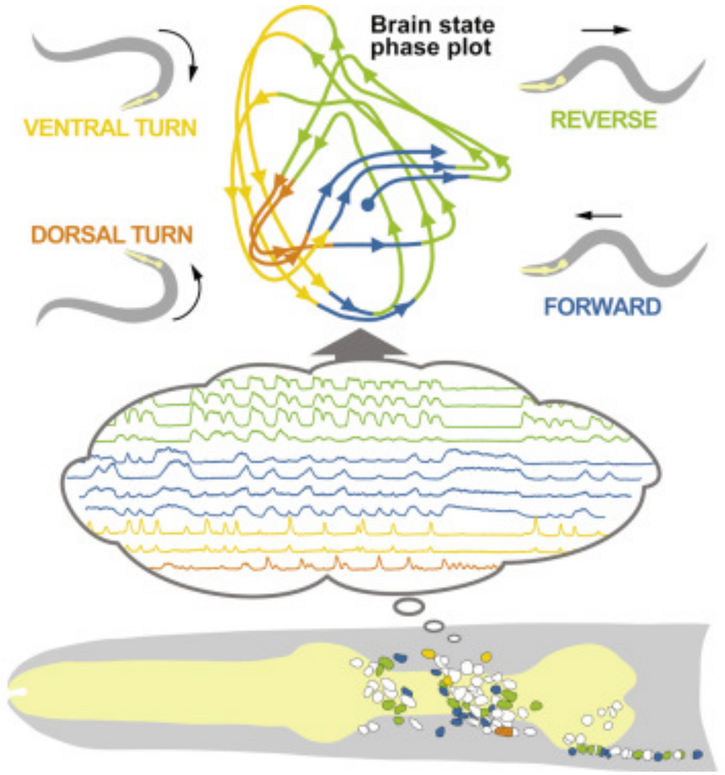


Please take this survey: https://tinyurl.com/stats320

Unit 3: Latent variable models of neural and behavioral data

Discovering dynamical states in whole-brain recordings

- A remarkable property of brain activity is that it is often **lower dimensional** than the sheer number of neurons.
- Moreover, the **dynamics** within this low dimensional space are often **indicative of** the animal's **behavior**.
- We will study **state space models** for characterizing these low dimensional dynamics.



Kato et al (Cell, 2015)

Please take this survey: https://tinyurl.com/stats320

Logistics

Lectures

- Lectures on **Monday/Wednesday, 11:30am-12:30pm PT** on Zoom.
- They will be **recorded** and available via Canvas links.
- **Slides** will be posted on Canvas after lecture.
- When possible, **long form notes** will also be made available.
- Each lecture will end with **further reading** for you to dig deeper, if you'd like.

Please take this survey: https://tinyurl.com/stats320

Labs

- Labs on **Friday, 11:30am-12:30pm PT** on Zoom.
- You'll work in **small, randomly assigned teams** to **implement a method** from lecture and **apply it to data.**
- Your **weekly assignment** will be to finish the lab with your teammates.
- The labs will **not be recorded** because we'll be in break-out rooms.
- TA's and I will bounce between rooms to **answer questions.**
- **Let me know** in advance if you have to miss a lab.
- For those in **other continents:** we'll try to pair you with one another and have you work with us during office hours (more info in a few slides).

Please take this survey: https://tinyurl.com/stats320

Final project

- The final project is an opportunity to **apply what you've learned** to a **problem of interest** to you.
- For example, you could:
  - **Implement a method** from a recent research paper and recapitulate its results on synthetic data.
  - Apply methods developed in class to **study a dataset of interest** to you.
  - **Propose and implement an extension** to an existing method that would address some of its limitations.
  - Perform a **theoretical analysis** of a method to study its statistical properties.
  - Perform a **literature review** of neural data analysis methods for a certain problem.
- You'll submit a proposal partway through the course and a final report + code at the end.
- You can work individually or in a small team (expectations will be scaled appropriately).

Please take this survey: https://tinyurl.com/stats320

Office hours

- Han: Tues morning 9–11am PT.
- Scott: Weds 1–2pm PT.
- Jaime: Thurs 5pm—7pm PT.
- Zoom links to be posted on Canvas.

Please take this survey: https://tinyurl.com/stats320

Grading

Labs (dropping lowest)	(8-1) x 10% each = 70% total
Project proposal	5%
Final project	25%

- There are eight labs total. I'll drop your lowest score.
- Labs will be due at midnight Thursday (i.e. 11:59pm PT).
- Please take up to 8 late days (max 3 per lab) throughout the quarter.

Please take this survey: https://tinyurl.com/stats320

Honor Code

1. The Honor Code is an undertaking of the students, individually and collectively:
    1. that they will not give or receive aid in examinations; that they will not give or receive unpermitted aid in class work, in the preparation of reports, or in any other work that is to be used by the instructor as the basis of grading;
    2. that they will do their share and take an active part in seeing to it that others as well as themselves uphold the spirit and letter of the Honor Code.
2. The faculty on its part manifests its confidence in the honor of its students by refraining from proctoring examinations and from taking unusual and unreasonable precautions to prevent the forms of dishonesty mentioned above. The faculty will also avoid, as far as practicable, academic procedures that create temptations to violate the Honor Code.
3. While the faculty alone has the right and obligation to set academic requirements, the students and faculty will work together to establish optimal conditions for honorable academic work.

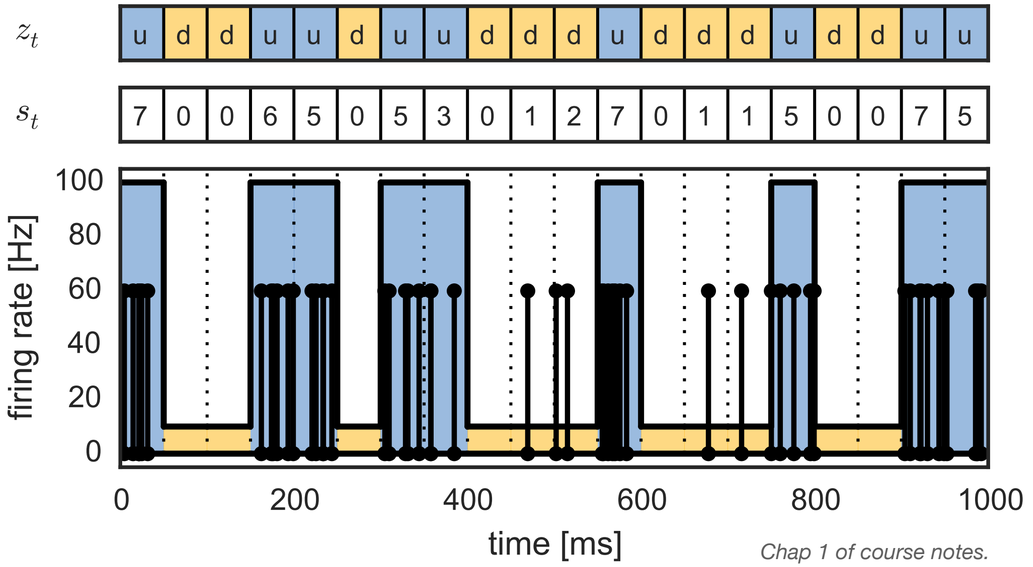
https://communitystandards.stanford.edu/policies-and-guidance/honor-code

Questions so far?

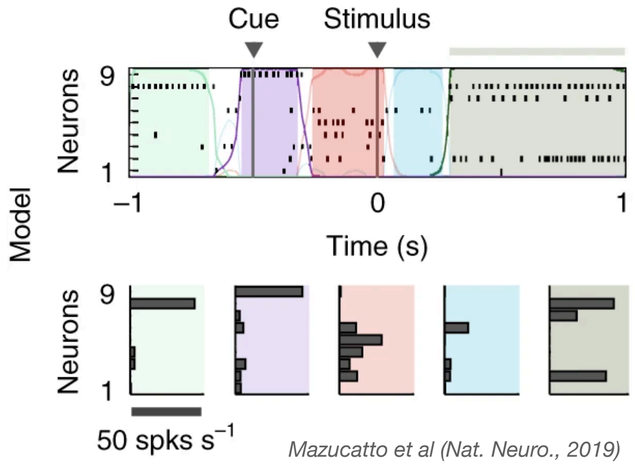
Technical Primer

A mixture model of neural spike trains

- Neural activity is sometimes characterized in terms of a progression of discrete states.
  - E.g. high firing (“up”) and low firing (“down”) states.
  - Sequences of “coding states” in gustatory cortex.
- Our goal is to infer these states given only the spike trains.



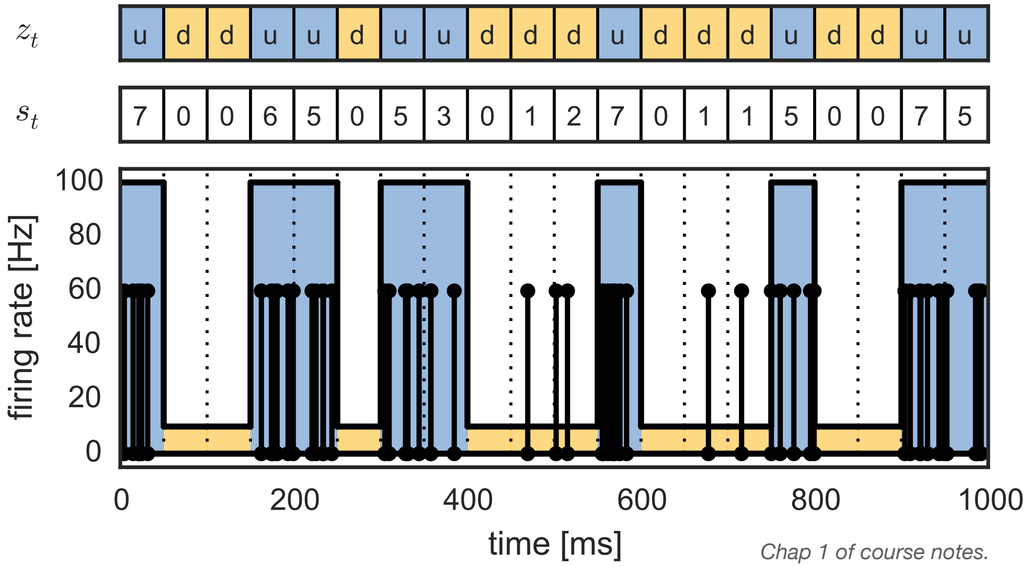
Chap 1 of course notes.



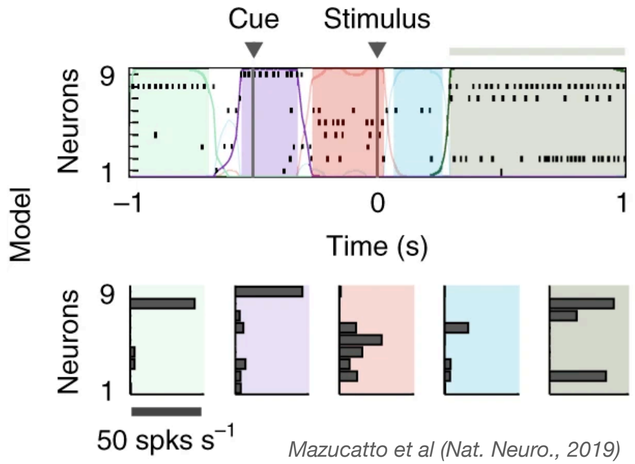
Mazucatto et al (Nat. Neuro., 2019)

Constants, Data, Latent Variables, and Parameters

- **Constants:** Let
  - T denote the number of time bins, each of width Δt [sec].
  - N denote the number of neurons.
  - K denote the number of discrete states.
- **Data**: Let
  - s_{tn} denote the number of spikes by neuron n in time bin t.
- **Latent variables**: Let
  - $z_t \in \{1, \ldots, K\}$ denote the discrete state during time bin t.
- **Parameters:** Let
  - $\lambda_{kn} \in \mathbb{R}_+$ denote the non-negative firing rate [spikes/sec] of neuron n in state k.
  - $\pi \in \Delta_K$ denote the prior probability of discrete states where Δ_K denotes the simplex of non-negative length-K vectors that sum to 1.



Chap 1 of course notes.



Mazucatto et al (Nat. Neuro., 2019)

Probabilistic Model

- Assume…
- Spike counts are conditionally **independent Poisson random variables**, given the current state:

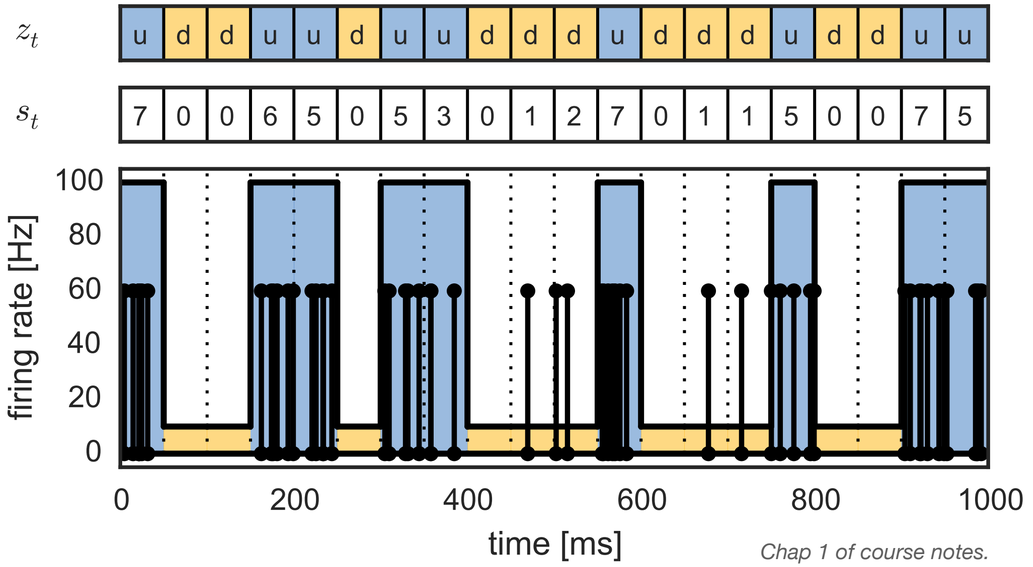
$$s_{tn} \sim \mathrm{Po}(\lambda_{z_t,n} \cdot \Delta t)$$

- Discrete states are independent across time:

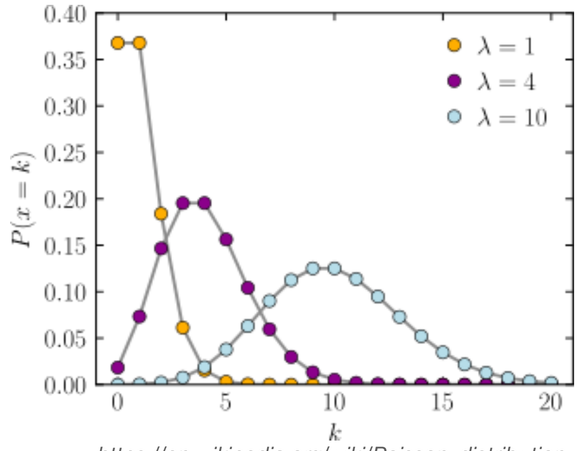
$$z_t \sim \pi$$

- Thus, the **likelihood** is:

$$p(s, z \mid \lambda, \pi) = \prod_{t=1}^{T} \left[\prod_{n=1}^{N} p(s_{tn} \mid z_t) \right] p(z_t \mid \pi)$$



Chap 1 of course notes.



https://en.wikipedia.org/wiki/Poisson_distribution

Probabilistic Model

- Assume…
- Spike counts are conditionally **independent Poisson random variables**, given the current state:

$$s_{tn} \sim \mathrm{Po}(\lambda_{z_t,n} \cdot \Delta t)$$

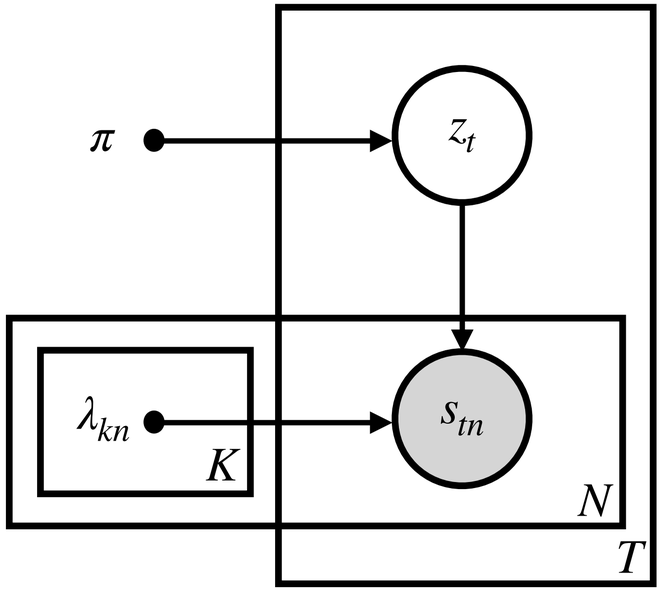
- Discrete states are independent across time:

$$z_t \sim \pi$$

- Thus, the **likelihood** is:

$$\begin{aligned}
p(s, z \mid \lambda, \pi) &= \prod_{t=1}^{T} \left[\prod_{n=1}^{N} p(s_{tn} \mid z_t) \right] p(z_t \mid \pi) \\
&= \prod_{t=1}^{T} \left[\prod_{n=1}^{N} \mathrm{Po}(s_{tn} \mid \lambda_{z_t,n} \Delta t) \right] \pi_{z_t} \\
&= \prod_{t=1}^{T} \prod_{k=1}^{K} \left(\left[\prod_{n=1}^{N} \mathrm{Po}(s_{tn} \mid \lambda_{kn} \Delta t) \right] \pi_k \right)^{\mathbb{I}[z_t=k]} \\
&= \prod_{t=1}^{T} \prod_{k=1}^{K} \left(\left[\prod_{n=1}^{N} \frac{1}{s_{tn}!} (\lambda_{kn} \Delta t)^{s_{tn}} e^{-\lambda_{kn} \Delta t} \right] \pi_k \right)^{\mathbb{I}[z_t=k]}
\end{aligned}$$

Probabilistic Graphical Model



Prior Distributions

- Assume…
- Firing rates are drawn from a gamma distribution with shape α and inverse-scale (aka “rate”) β:

$$\lambda_{kn} \sim \mathrm{Ga}(\alpha, \beta)$$

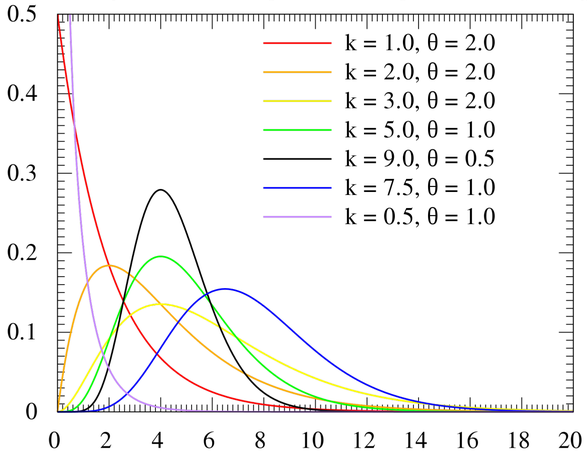
- The state probabilities are drawn from a symmetric Dirichlet prior distribution with concentration γ:

$$\pi \sim \mathrm{Dir}(\gamma 1_K) = \mathrm{Dir}([\gamma, \ldots, \gamma])$$

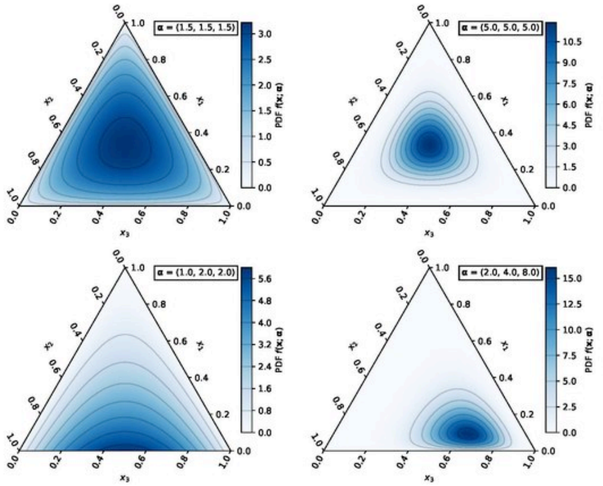
- Thus, the **prior density** is:

$$\begin{aligned} p(\lambda \mid \alpha, \beta) &= \prod_{k=1}^{K} \prod_{n=1}^{N} \mathrm{Ga}(\lambda_{kn} \mid \alpha, \beta) \\ &= \prod_{k=1}^{K} \prod_{n=1}^{N} \frac{\beta^{\alpha}}{\Gamma(\alpha)} \lambda_{kn}^{\alpha-1} e^{-\beta \lambda_{kn}} \\ p(\pi \mid \gamma) &= \mathrm{Dir}(\pi \mid \gamma 1_K) \\ &= \frac{\Gamma(K\gamma)}{\Gamma(\gamma)^K} \prod_{k=1}^{K} \pi_k^{\gamma-1} \end{aligned}$$

Gamma probability density function (pdf)



Dirichlet pdf



Putting it all together

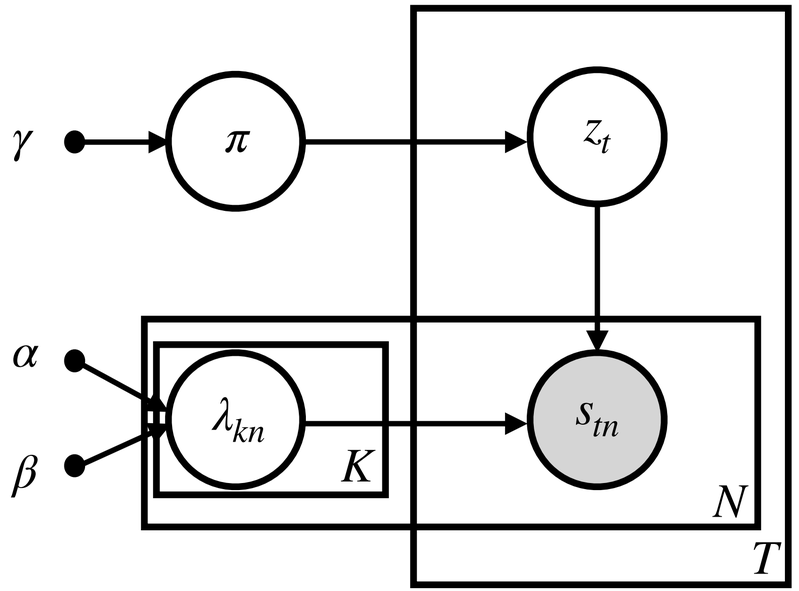
- Using the **chain rule** of probability, the **joint distribution** is:

$$p(s, z, \lambda, \pi \mid \alpha, \beta, \gamma) = p(s, z \mid \lambda, \pi)\, p(\lambda \mid \alpha, \beta)\, p(\pi \mid \gamma)$$

- Substituting our functional forms from the previous slides,

$$p(s, z, \lambda, \pi \mid \alpha, \beta, \gamma) = \left[\prod_{t=1}^{T} \left[\prod_{n=1}^{N} \mathrm{Po}(s_{tn} \mid \lambda_{z_t, n} \Delta t) \right] \pi_{z_t} \right] \times \left[\prod_{k=1}^{K} \prod_{n=1}^{N} \mathrm{Ga}(\lambda_{kn} \mid \alpha, \beta) \right] \mathrm{Dir}(\pi \mid \gamma 1_K)$$

Probabilistic Graphical Model



Maximum a posteriori estimation

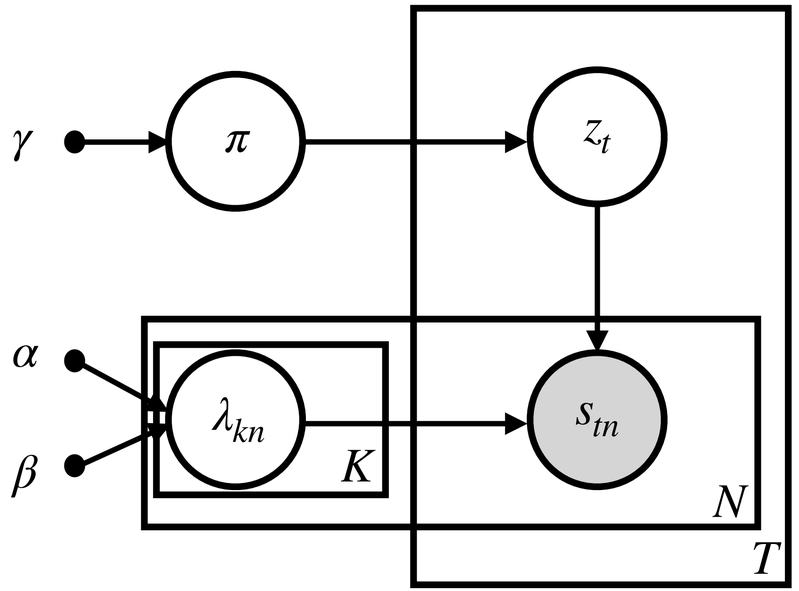
- Our goal is to infer the discrete states z and the parameters λ, π. We'll do so by finding the mode of the posterior distribution; i.e. by *maximum a posteriori* (MAP) estimation:

$$\begin{aligned}\hat{z}, \hat{\lambda}, \hat{\pi} &= \text{argmax}\, p(z, \lambda, \pi \mid s, \alpha, \beta, \gamma) \\ &= \text{argmax}\, \frac{p(s, z, \lambda, \pi \mid \alpha, \beta, \gamma)}{p(s \mid \alpha, \beta, \gamma)} \\ &= \text{argmax}\, p(s, z, \lambda, \pi \mid \alpha, \beta, \gamma)\end{aligned}$$

  where the second line follows by Bayes' rule and the third follows from the fact that the denominator is not a function of z and λ.

- In other words, we can find the MAP estimate by maximizing the joint probability.

Probabilistic Graphical Model



Coordinate ascent: discrete states

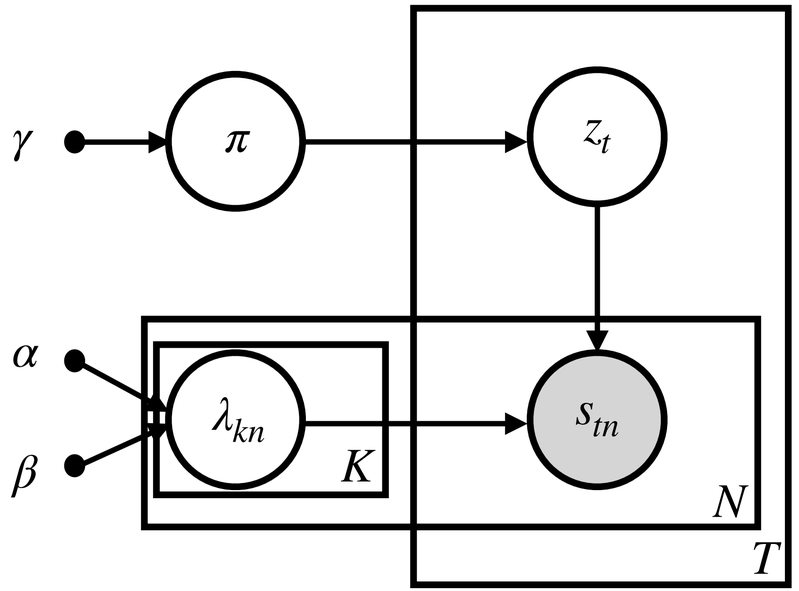
- We'll find the argmax by coordinate ascent, iteratively maximizing with respect to one variable at a time.
- First, consider z_t. Only a few terms in the objective (i.e. the log joint) contain this variable:

$$p(s, z, \lambda, \pi \mid \alpha, \beta, \gamma) \propto \prod_{k=1}^{K} \left(\left[\prod_{n=1}^{N} \mathrm{Po}(s_{tn} \mid \lambda_{kn} \Delta t) \right] \pi_k \right)^{\mathbb{I}[z_t = k]}$$

- We'll set z_t to the state k with the largest term inside the parentheses:

$$\begin{aligned} \hat{z}_t &= \mathrm{argmax}_k \left[\prod_{n=1}^{N} \mathrm{Po}(s_{tn} \mid \lambda_{kn} \Delta t) \right] \pi_k \\ &= \mathrm{argmax}_k \left[\sum_{n=1}^{N} \log \mathrm{Po}(s_{tn} \mid \lambda_{kn} \Delta t) \right] + \log \pi_k \end{aligned}$$

Probabilistic Graphical Model



Coordinate ascent: firing rates

- Now consider λ_{kn}. Again, only a few terms in the objective contain this variable:

$$
\begin{aligned}
p(s, z, \lambda, \pi \mid \alpha, \beta, \gamma) &\propto \left[\prod_{t=1}^{T} \mathrm{Po}(s_{tn} \mid \lambda_{kn}\Delta t)^{\mathbb{I}[z_t=k]}\right] \mathrm{Ga}(\lambda_{kn} \mid \alpha, \beta) \\
&\propto \left[\prod_{t=1}^{T} \lambda_{kn}^{s_{tn}\cdot\mathbb{I}[z_t=k]} e^{\lambda_{kn}\Delta t\cdot\mathbb{I}[z_t=k]}\right] \lambda_{kn}^{\alpha-1} e^{-\beta\lambda_{kn}} \\
&\propto \mathrm{Ga}(\lambda_{kn} \mid \alpha'_{kn}, \beta'_{kn})
\end{aligned}
$$

where

$$\alpha'_{kn} = \alpha + \sum_{t=1}^{T} s_{tn} \cdot \mathbb{I}[z_t = k]$$

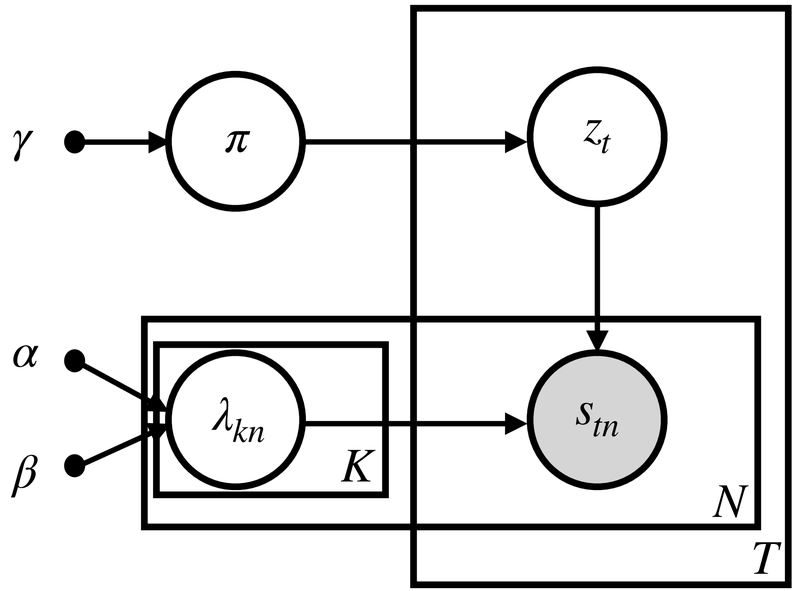
$$\beta'_{kn} = \beta + \sum_{t=1}^{T} \Delta t \cdot \mathbb{I}[z_t = k]$$

- Since the conditional distribution is in the same family (i.e. gamma) as the prior, we call it a **conjugate prior** distribution.
- This is maximized at the mode of the gamma distribution:

$$\hat{\lambda}_{kn} = \frac{\alpha'_{kn} - 1}{\beta'_{kn}} \text{ for } \alpha'_{kn} \geq 1.$$

- What happens as the prior goes to a “uniform” prior on λ_{kn}? That is, as $\alpha \to 1$ and $\beta \to 0$.

Probabilistic Graphical Model



Coordinate ascent: state probabilities

- Finally, consider the state probabilities π. We have,

$$p(s, z, \lambda, \pi \mid \alpha, \beta, \gamma) \propto \left[\prod_{t=1}^{T} \prod_{k=1}^{K} \pi_k^{\mathbb{I}[z_t = k]} \right] \mathrm{Dir}(\pi \mid \gamma 1_K)$$

$$\propto \left[\prod_{t=1}^{T} \prod_{k=1}^{K} \pi_k^{\mathbb{I}[z_t = k]} \right] \left[\prod_{k=1} \pi_k^{\gamma - 1} \right]$$

$$\propto \mathrm{Dir}(\pi \mid [\gamma'_1, \ldots, \gamma'_K])$$

where

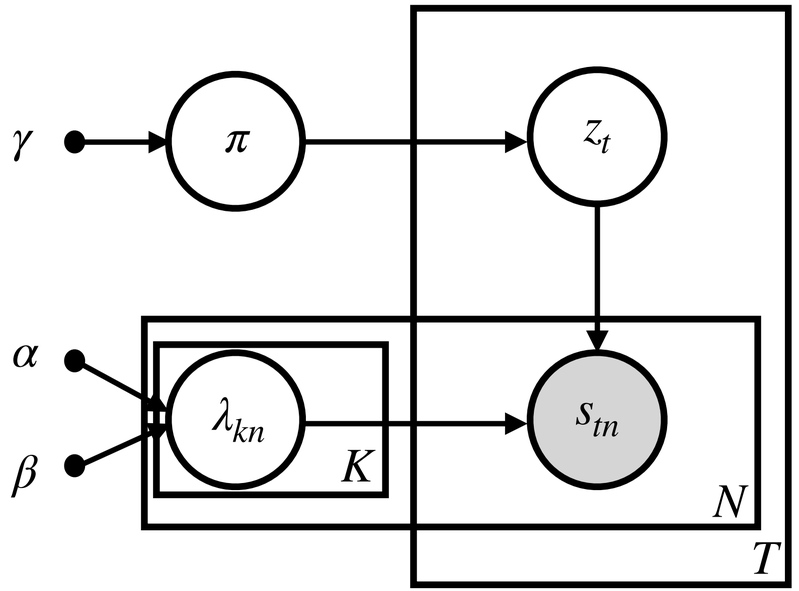
$$\gamma'_k = \gamma + \sum_{t=1}^{T} \mathbb{I}[z_t = k]$$

- We see that the Dirichlet distribution is a conjugate prior.
- The mode of the Dirichlet distribution is at,

$$\hat{\pi}_k = \frac{\gamma'_k - 1}{\sum_{j=1}^{K} (\gamma'_j - 1)} \text{ for } \gamma'_k \geq 1.$$

- What happens when we have a uniform prior on π? That is, when $\gamma = 1$.

Probabilistic Graphical Model



Python Code

- Spend some time after class to see that this code implements one iteration of the coordinate ascent updates we just derived.
- Copy/paste it into a Colab notebook and try it out!

```
import numpy as np
from scipy.stats import poisson

def update(spikes, states, rates, probs,
           alpha=0.1, beta=0.1, gamma=1.0, dt=1.0):
    """Perform a single step of coordinate ascent."""
    T, N = spikes.shape
    K, _ = rates.shape

    # update the states
    scores = np.column_stack([
        poisson(rates_k * dt).logpmf(spikes).sum(axis=1)
        for rates_k in rates])
    scores += np.log(probs)
    states = np.argmax(scores, axis=1)

    # update the rates
    alpha_prime = np.array([
        alpha + spikes[states == k].sum(axis=0)
        for k in range(K)])
    beta_prime = np.array([
        [beta + dt * (states == k).sum()]
        for k in range(K)])
    rates = (alpha_prime - 1) / beta_prime

    # update the state probs
    gamma_prime = np.array([
        gamma + (states == k).sum()
        for k in range(K)])
    probs = gamma_prime / gamma_prime.sum()

    return states, rates, probs
```

Further Reading

- For a recent perspective on statistical neuroscience:
  - Cunningham and Paninski (Curr. Opin. Neurobio., 2019)
- For more on probabilistic modeling:
  - Ch 1 of course notes
  - Ch 1.1-1.2 of Bishop's Pattern Recognition and Machine Learning

Conclusion

- Please take the survey: https://tinyurl.com/stats320
- Zoom recording should be online soon.
- Next time: Spike Sorting