Machine Learning Methods for Neural Data Analysis

EM, Mixture Models, and Hidden Markov Models

Scott Linderman

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

Announcements

- Correction in notes:
    - The blocks are given by $J_{tt} = Q^{-1} + A^\top Q^{-1} A + C^\top R^{-1} C$ (except for J_{11} and J_{TT}).

- 1 page **project proposal** due **Monday, Feb 27**. Teams of 2-3 people. Ed could be a great way to find teammates!

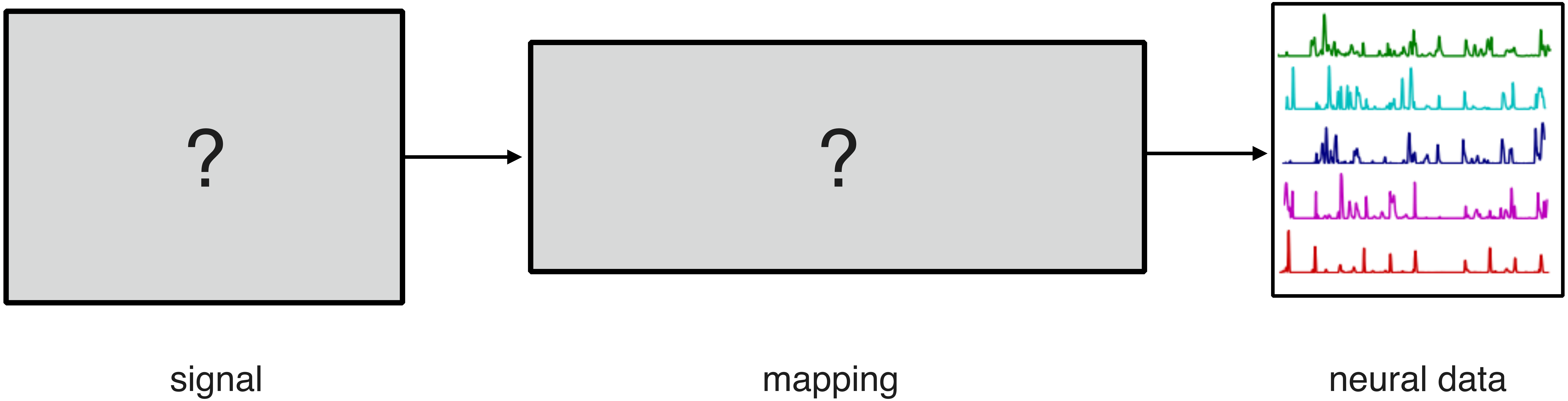
Agenda

- Intro to Unit III: Unsupervised Learning
- Expectation-maximization for Gaussian mixture models
- Hidden Markov models and the forward-backward algorithm

Unit III: Unsupervised learning

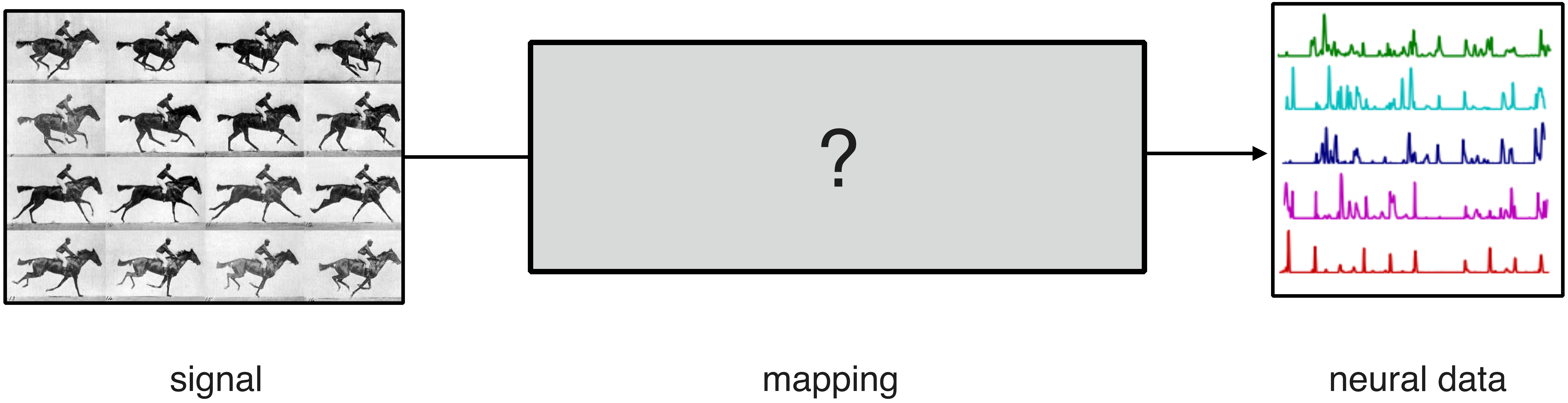
Data-driven modeling

Searching for signals to explain neural activity



Data-driven modeling

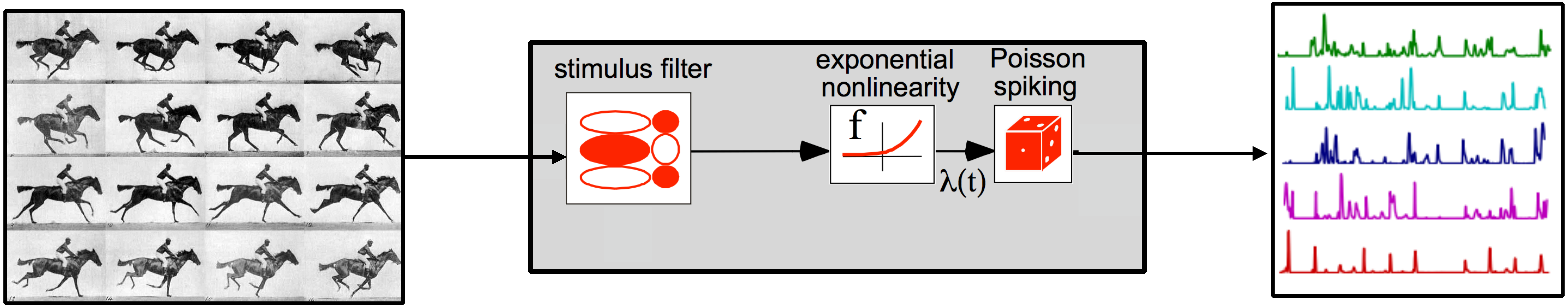
Searching for signals to explain neural activity



Encoding models: given stimulus (covariates) and response, find mapping.

Data-driven modeling

Searching for signals to explain neural activity



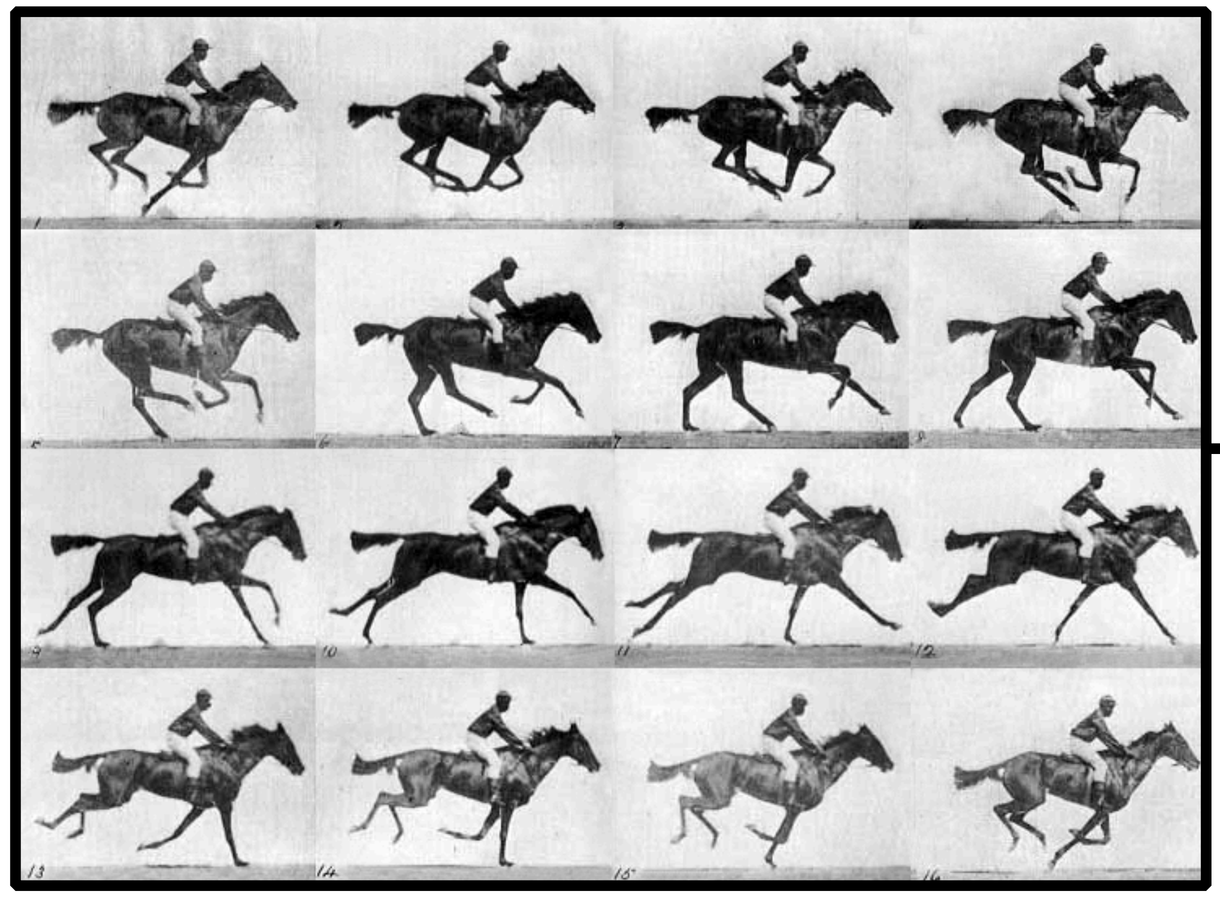
signal mapping neural data

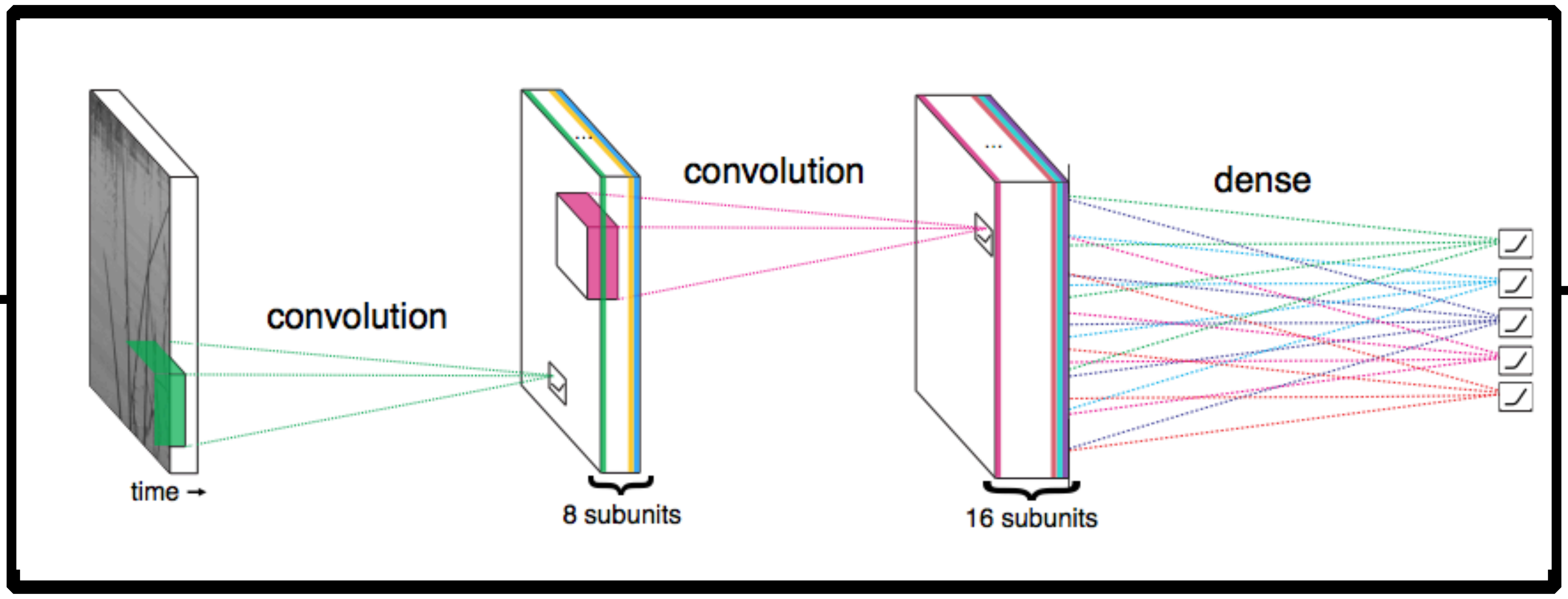
Recent examples: Musall et al (2018), Stringer et al (2018)

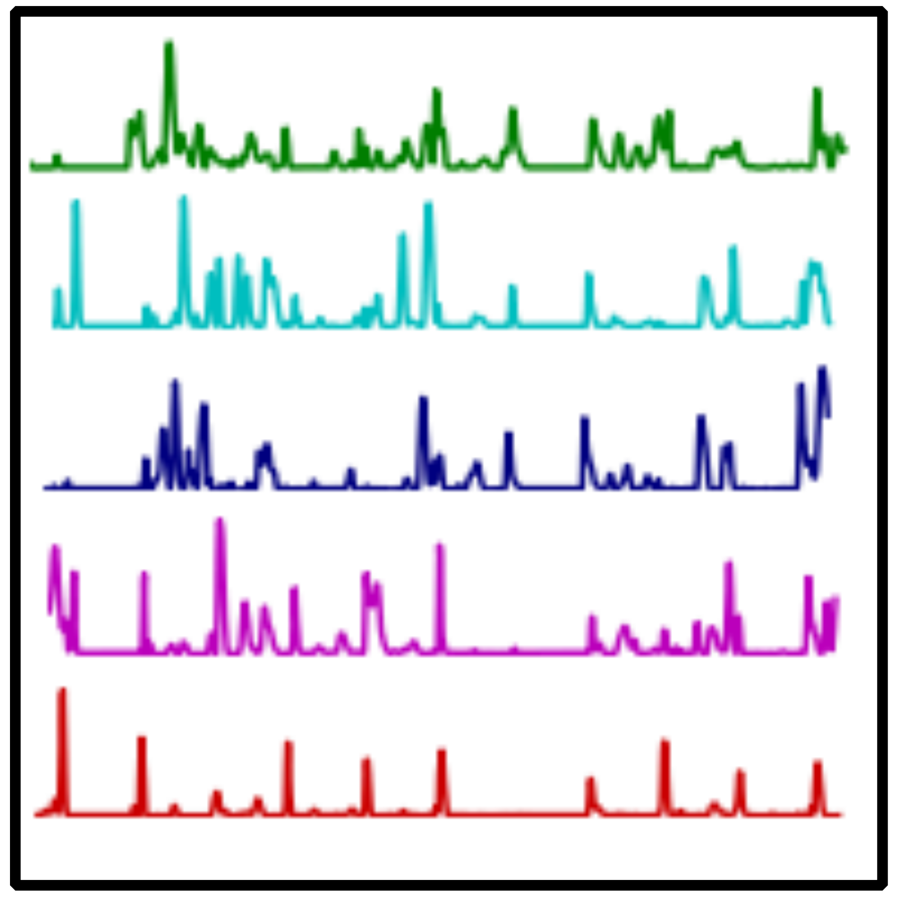
Paninski (2004)
Truccolo et al (2005)
Pillow et al (2008)

Data-driven modeling

Searching for signals to explain neural activity







signal

mapping

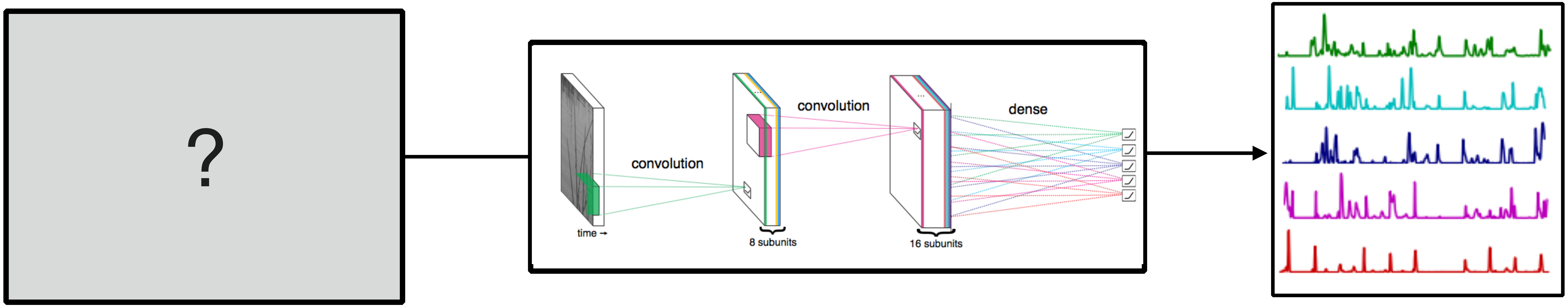
neural data

Toward nonlinear and/or more biophysically plausible mappings.

McIntosh et al (2017)

Data-driven modeling

Searching for signals to explain neural activity



latent signal

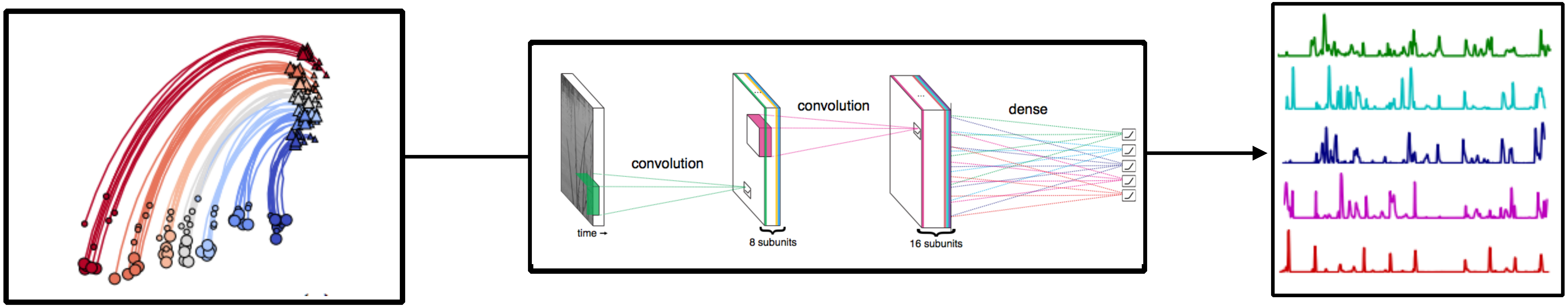
mapping

neural data

Alternative: try to infer latent signals from the data

Data-driven modeling

Searching for signals to explain neural activity



latent signal

mapping

neural data

Alternative: try to infer latent signals from the data, *subject to constraints.*

Latent variable modeling is all about constraints

The five D's

- *Dimensionality*: how many latent clusters, factors, etc.?
- *Domain*: are the latent variables discrete, continuous, bounded, sparse, etc.?
- *Dynamics*: how do the latent variables change over time?
- *Dependencies*: how do the latent variables relate to the observed data?
- *Distribution*: do we have prior knowledge about the variables' probability?

- We've already seen some examples in Unit 1!

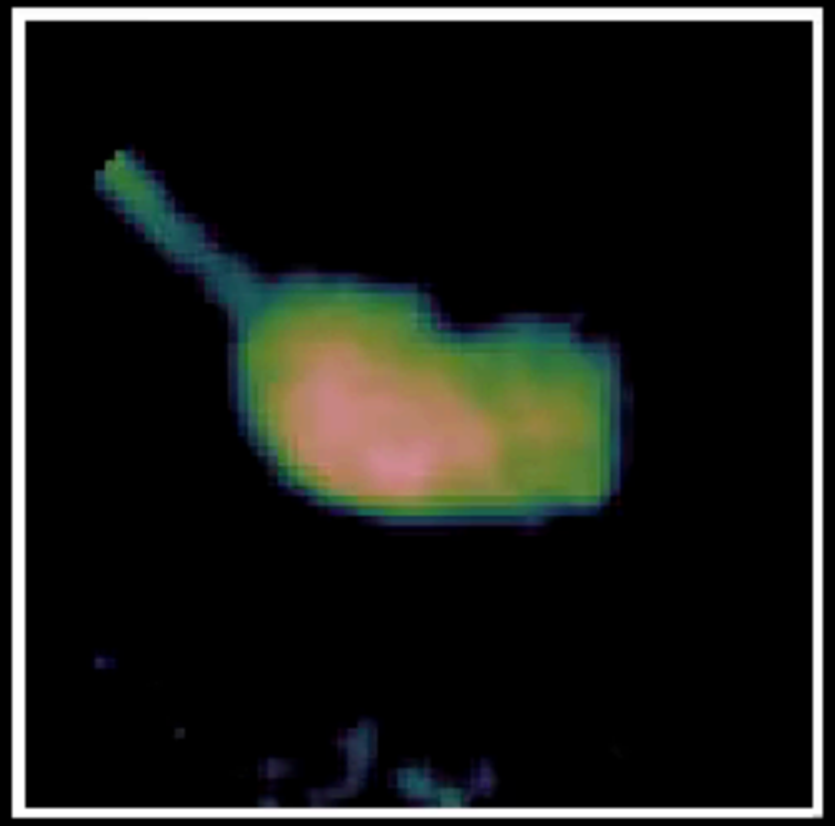
Latent variable modeling is all about constraints

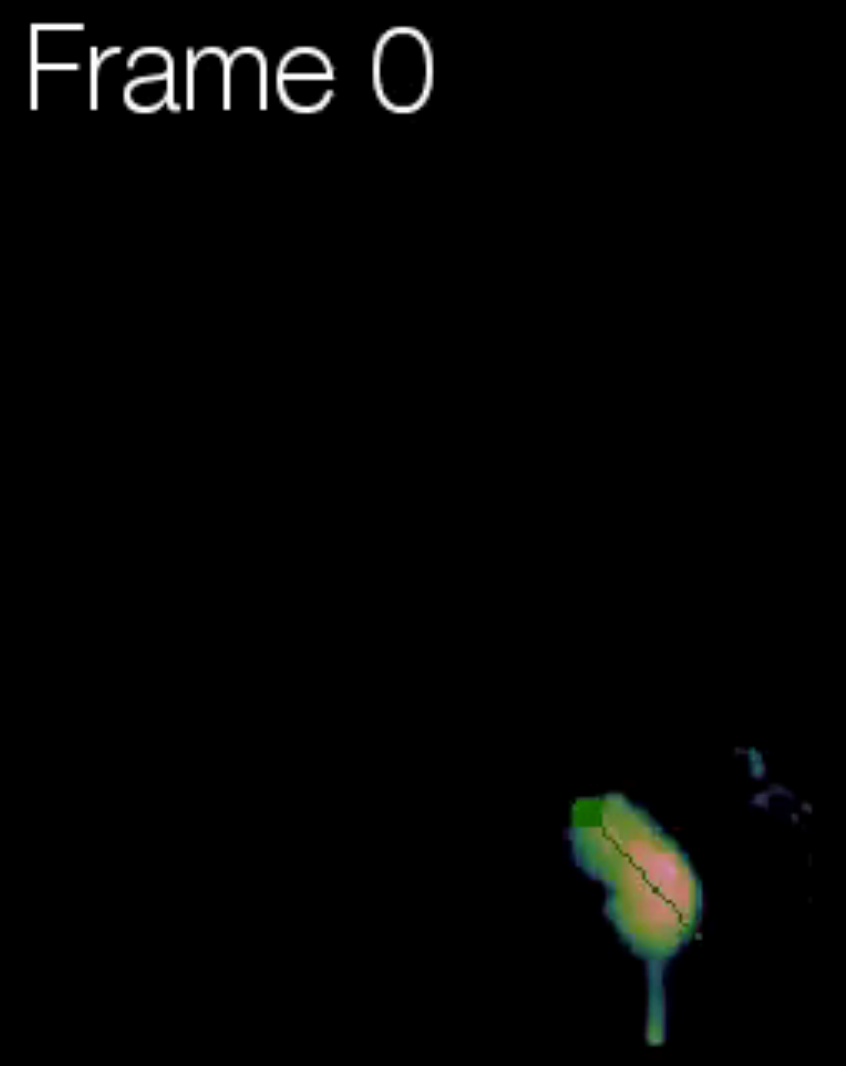
Domain/Dependency/Distribution

Dynamics /Domain

	Continuous Linear Gaussian	Discrete (Gen.) Linear Bernoulli/Poisson/etc.	Nonlinear Observation Models
Discrete Markovian Categorical	**HMM** *Rabiner (1989)*	**HMM** *Rabiner (1989)*	**Structured VAE** *Johnson et al (2016)*
Continuous Linear Gaussian	**LDS** *Kalman (1960)*	**Poisson LDS** *Smith and Brown (2003), Paninski et al (2010)*	**Deep PfLDS** Archer et al (2015); Gao et al (2016)
Continuous Nonlinear (parametric) Gaussian	**NLDS, e.g. Hodgkin-Huxley** *Ahrens, Huys, Paninski (2006)* *Huys and Paninski (2009)*	**NLDS, e.g. Hodgkin-Huxley** *Meng, Kramer, Eden (2011)*	**GPSSM, DKF, LFADS, VIND** *Frigola et al (2013) , Krishnan et al (2015), Sussillo et al (2016), Hernandez et*
Mixed Switching Linear	**SLDS** *Ghahramani and Hinton (1996)* *Murphy (1998)*	**Poisson SLDS** *Petreska et al (2013)*	**Structured VAE** *Johnson et al (2016)*
Mixed Recurrent Linear	**recurrent/augmented SLDS** *Barber (2006); Pachitariu et al (2014); Linderman et al (2017); Nassar et al*	**rSLDS** *Linderman et al (2017)* *Nassar et al (2019)*	**Structured VAE** *Johnson et al (2016)*
Continuous Nonlinear (smoothing) Gaussian	**GPFA** *Yu, Cunningham, et al (2009)*	**vLGP** *Zhao and Park (2017)*	**GPLVM** *Lawerence (2005), Wu et al (2017)*
Continuous Nonlinear (nonparametric) Gaussian	**GPSSM, DKF, LFADS, VIND** *Frigola et al (2013) , Krishnan et al (2015), Sussillo et al (2016), Hernandez*	**GPSSM, DKF, LFADS, VIND** *Frigola et al (2013) , Krishnan et al (2015), Sussillo et al (2016), Hernandez et*	**GPSSM, DKF, LFADS, VIND** *Frigola et al (2013) , Krishnan et al (2015), Sussillo et al (2016), Hernandez et*

Motivating Example: summarizing videos with behavioral states

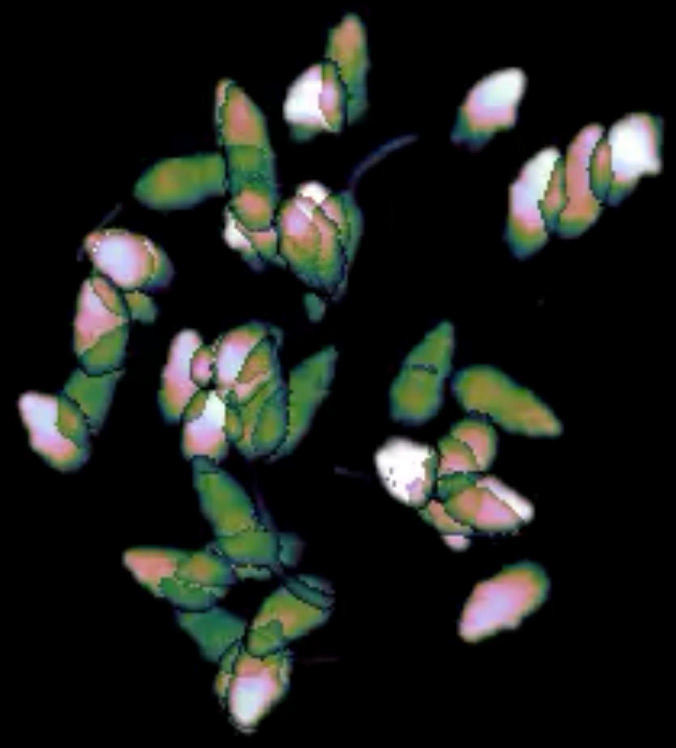




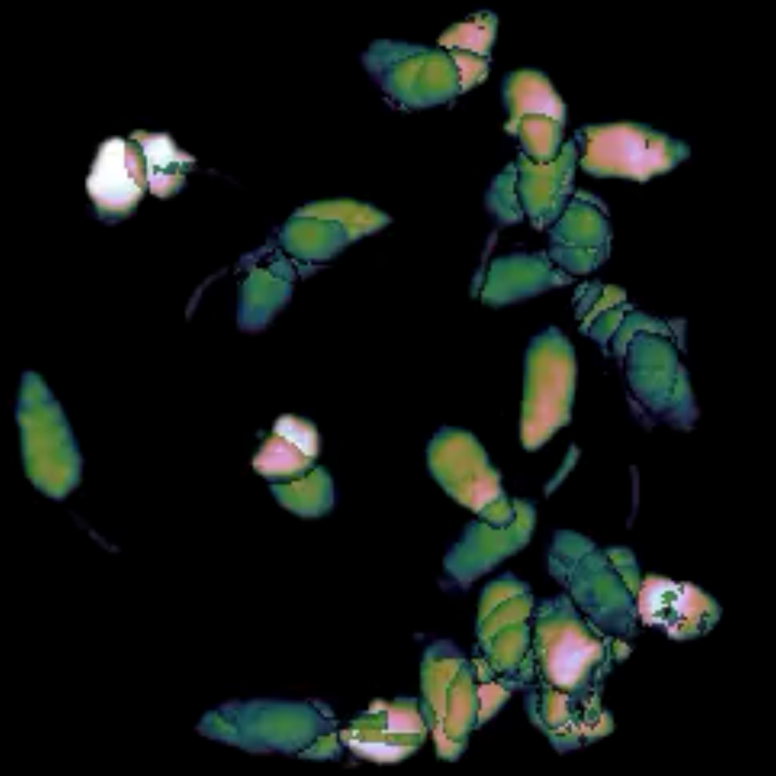
Wiltschko et al, 2015

Motivating Example: summarizing videos with behavioral states

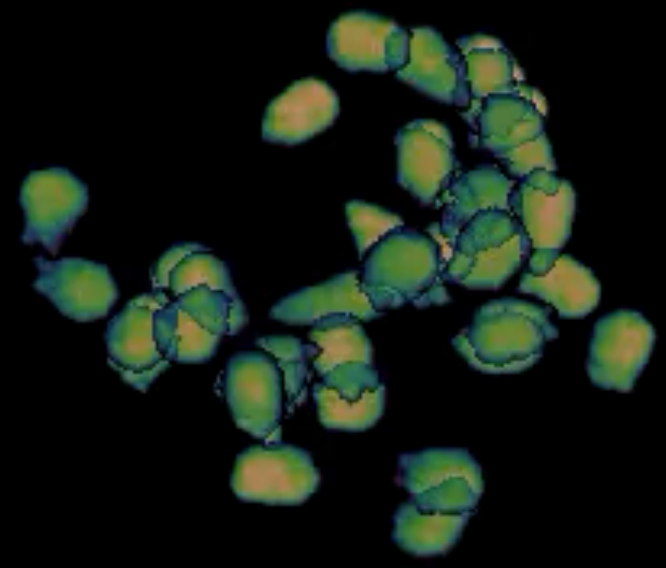
Rear down



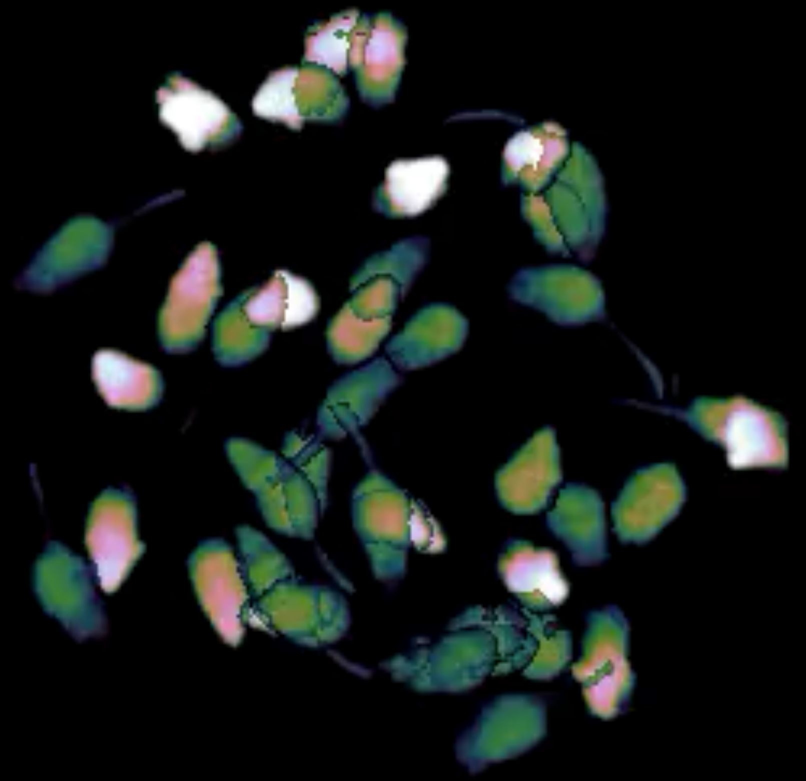
Walk forward



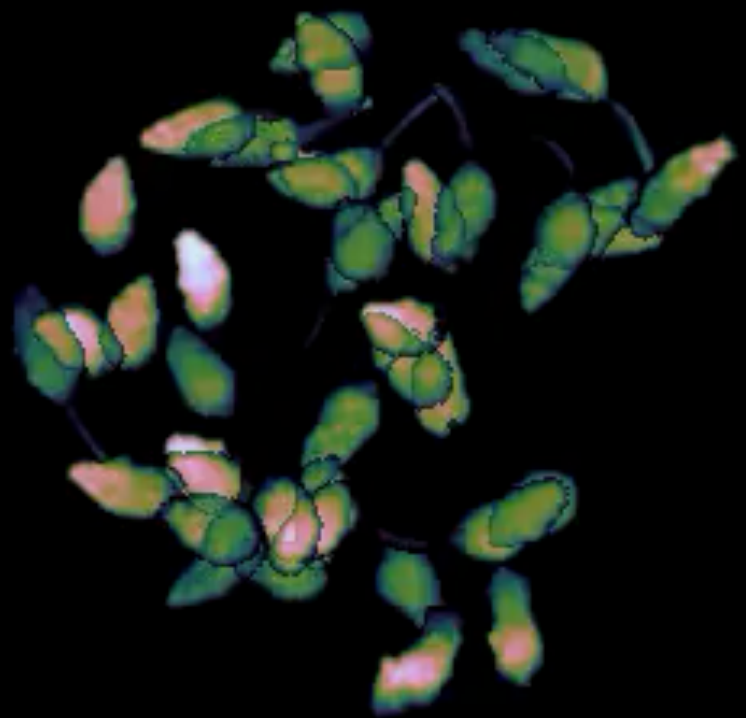
Grooming



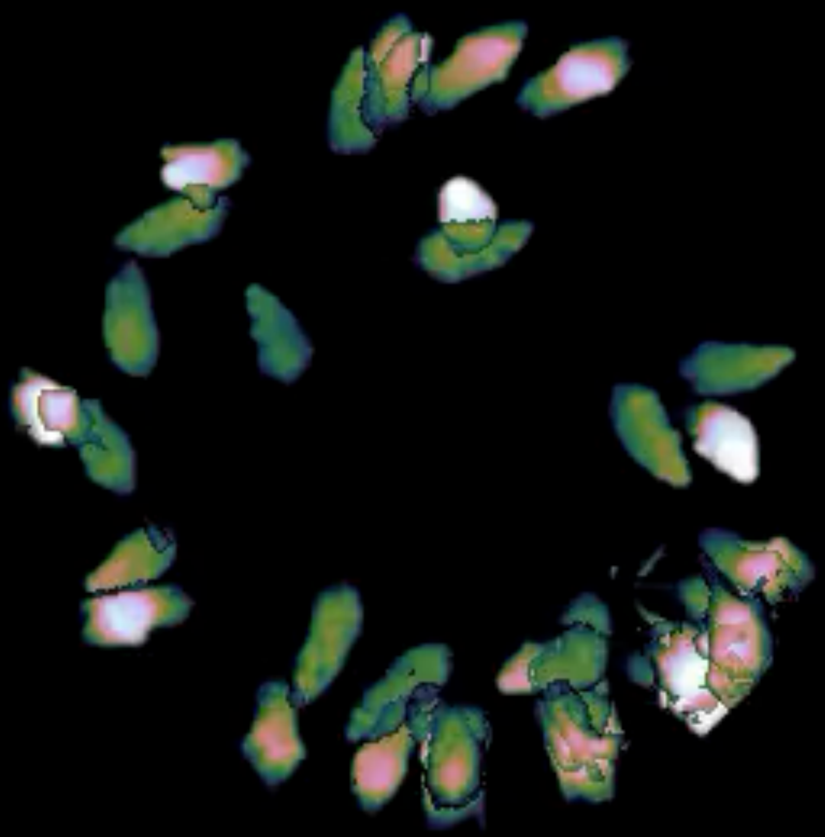
Scrunch



Rear up



Jump



Wiltschko et al, 2015

Bayesian inference in latent variable models

Formulating as a probabilistic model

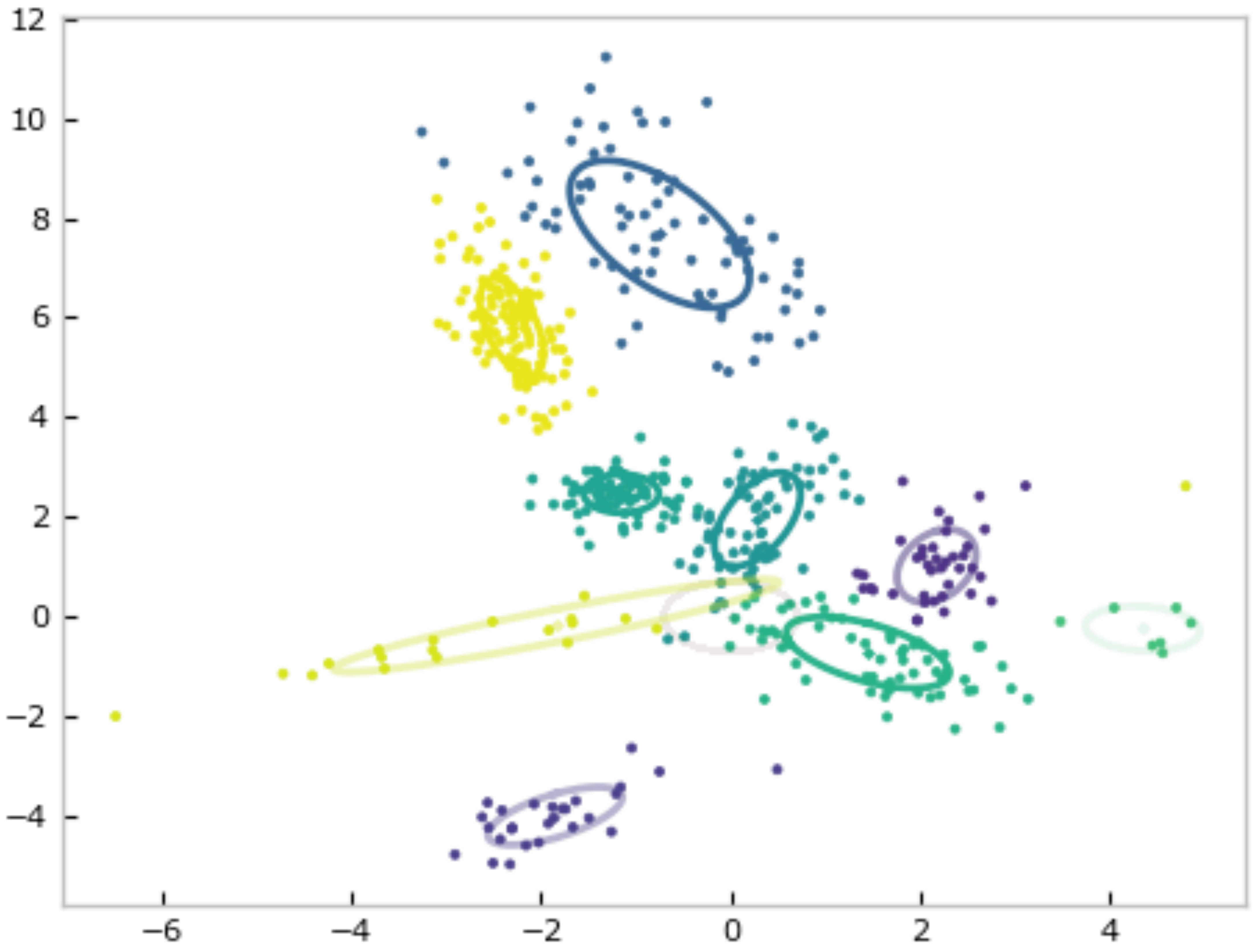
- **Variables:** Let,
  - $x_t \in \mathbb{R}^P$ denote the (vectorized) image at time t.
  - $z_t \in \{1, \ldots, K\}$ denote the discrete latent state (aka behavioral "syllable") at time t.
- **Model:** Assume each time frame is independent and,

$$\begin{aligned} z_t &\sim \mathrm{Cat}(\pi) \\ x_t \mid z_t &\sim \mathcal{N}(b_{z_t}, Q_{z_t}) \end{aligned}$$

- **Parameters:** Let $\Theta = \pi, \{b_k, Q_k\}_{k=1}^K$ denote the parameters,
  - $\pi \in \Delta_K$ is the prior probability of each state
  - $(b_k, Q_k) \in \mathbb{R}^P \times \mathbb{R}^{P \times P}$ are the conditional mean and variance of images for discrete state $z_t = k$.

The Gaussian Mixture Model

Example draw from a 2D GMM with 10 clusters



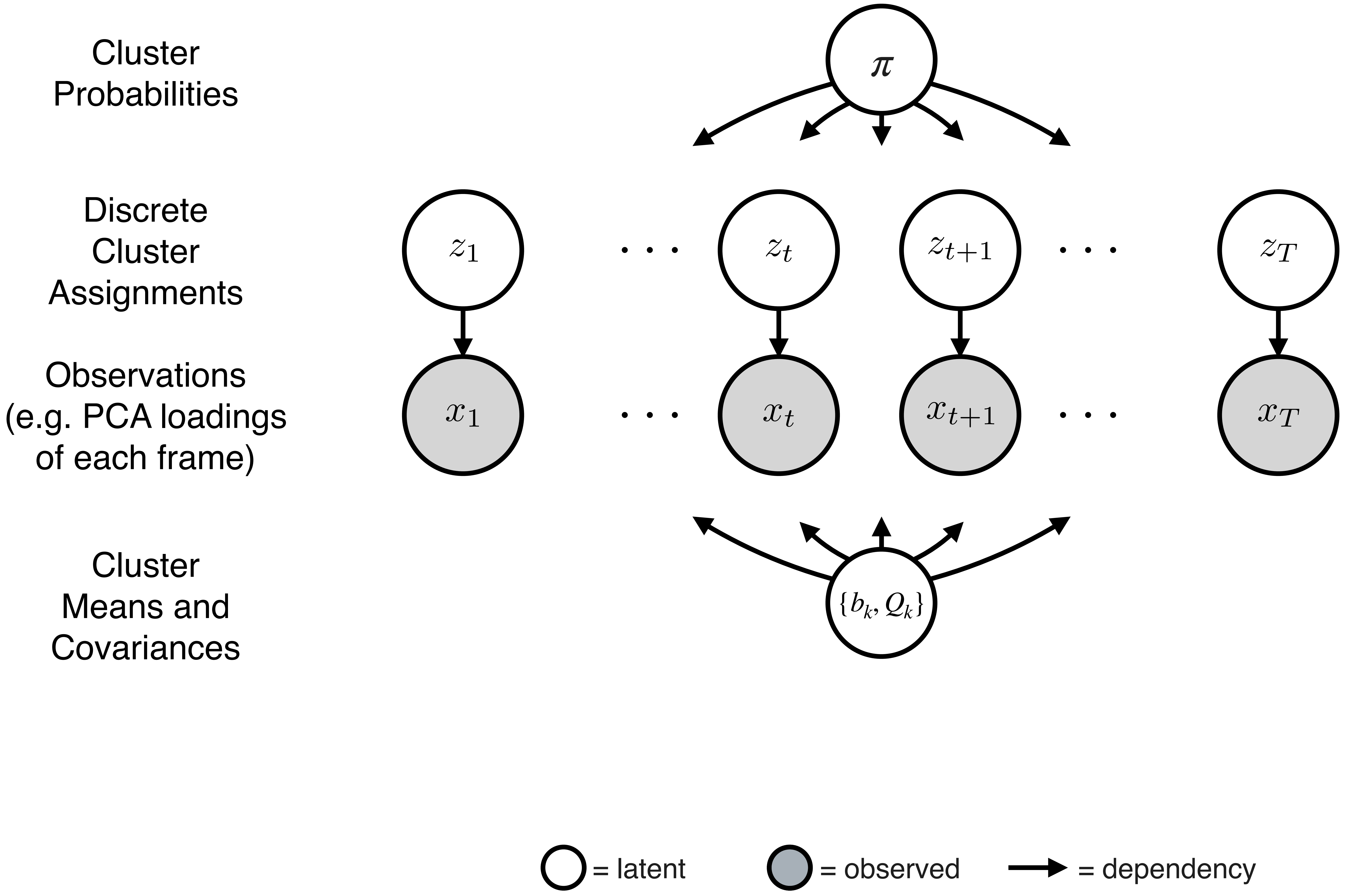
The Gaussian Mixture Model

The **joint probability** factors into a product over time bins,

$$p(x, z \mid \Theta) = \prod_{t=1}^{T} p(z_t)\, p(x_t \mid z_t)$$

The Gaussian Mixture Model

Graphical Model



Bayesian inference in latent variable models

MAP Estimation

- In Unit 1 we used ***maximum a posteriori* (MAP) estimation** to find,

$$z^\star, \Theta^\star = \arg\max_{z,\Theta} \log p(x, z, \Theta)$$

- Coordinate ascent (effectively the same as **k-means!**). Repeat:
  - Update cluster assignments:

$$z_t = \arg\max_k \pi_k \cdot \mathcal{N}(y_t \mid b_k, Q_k)$$

assign each data point to the most likely cluster

  - Update parameters for each $k = 1,\ldots,K$:

$$T_k = \sum_{t=1}^{T} \mathbb{I}[z_t = k]$$

count number of frames assigned to each cluster

$$b_k = \frac{1}{T_k} \sum_{t=1}^{T} y_t \, \mathbb{I}[z_t = k]$$

set means equal to the sample mean of assigned data points

$$Q_k = \frac{1}{T_k} \sum_{t=1}^{T} (y_t - d_k)(y_t - d_k)^\top \, \mathbb{I}[z_t = k]$$

set covariance equal to the sample covariance of assigned data points

Bayesian inference in latent variable models

MAP Estimation

- This gives us a **point estimate of the latent variables z and parameters Θ.**
- Point estimates can lead to an **overly optimistic** view of the model.
- Specifically, MAP estimation found **the best assignment**, which may not reflect the **average performance** under the prior $p(z, \Theta)$.
- **Question:** What if only one data point is assigned to a cluster on some iteration?

Bayesian inference in latent variable models

Integrating over the latent variables

- A more **Bayesian approach** is to **integrate** over the latent variables.
- First, **learn** a point estimate of the parameters,

$$\Theta^\star = \arg\max_\Theta \log p(x, \Theta)$$

where $p(x, \Theta) = \int p(x, z, \Theta)\, \mathrm{d}z = \mathbb{E}_{p(z,\Theta)}[p(x \mid z, \Theta)]$ is the **marginal likelihood.**

Bayesian inference in latent variable models

Integrating over the latent variables

- A more **Bayesian approach** is to **integrate** over the latent variables.
- First, **learn** a point estimate of the parameters,

$$\Theta^\star = \arg\max_\Theta \; \log p(x, \Theta)$$

where $p(x, \Theta) = \int p(x, z, \Theta) \, \mathrm{d}z = \mathbb{E}_{p(z,\Theta)}[p(x \mid z, \Theta)]$ is the **marginal likelihood.**

- Then, **infer** the posterior distribution over latent variables given observed data and parameters,

$$p(z \mid x, \Theta) = \frac{p(x \mid z, \Theta) \, p(z \mid \Theta) \, p(\Theta)}{p(x, \Theta)}$$

- (A "fully Bayesian" approach would integrate over both z and Θ.)

Bayesian inference in latent variable models

Maximizing the marginal likelihood

- How to learn the parameters?
- First idea: **gradient ascent**,

$$\nabla_{\Theta} \log p(x, \Theta) = \frac{\nabla_{\Theta} p(x, \Theta)}{p(x, \Theta)} = \frac{\int \nabla_{\Theta} p(x, z, \Theta)\, \mathrm{d}z}{\int p(x, z, \Theta)\, \mathrm{d}z}$$

- Sometimes, these integrals are available in **closed form**.
    - For example, when z **is discrete** the integrals become sums.
- Can we do better?

Bayesian inference in latent variable models

Lower bound the marginal likelihood

- Next idea: lower bound the marginal likelihood with a more tractable form,

$$\log p(x, \Theta) = \log \int p(x, z, \Theta) \, \mathrm{d}z$$

Bayesian inference in latent variable models

Lower bound the marginal likelihood

- Next idea: lower bound the marginal likelihood with a more tractable form,

$$\log p(x, \Theta) = \log \int p(x, z, \Theta) \, \mathrm{d}z$$

$$= \log \int \frac{q(z)}{q(z)} p(x, z, \Theta) \, \mathrm{d}z \qquad \text{for any distribution } q(z)$$

Bayesian inference in latent variable models

Lower bound the marginal likelihood

- Next idea: lower bound the marginal likelihood with a more tractable form,

$$\log p(x, \Theta) = \log \int p(x, z, \Theta)\, \mathrm{d}z$$

$$= \log \int \frac{q(z)}{q(z)} p(x, z, \Theta)\, \mathrm{d}z \qquad \text{for any distribution } q(z)$$

$$= \log \mathbb{E}_{q(z)} \left[\frac{p(x, z, \Theta)}{q(z)} \right]$$

Bayesian inference in latent variable models

Lower bound the marginal likelihood

- Next idea: lower bound the marginal likelihood with a more tractable form,

$$\begin{aligned}
\log p(x, \Theta) &= \log \int p(x, z, \Theta)\, \mathrm{d}z \\
&= \log \int \frac{q(z)}{q(z)} p(x, z, \Theta)\, \mathrm{d}z && \text{for any distribution } q(z) \\
&= \log \mathbb{E}_{q(z)} \left[\frac{p(x, z, \Theta)}{q(z)} \right] \\
&\geq \mathbb{E}_{q(z)} \left[\log p(x, z, \Theta) - \log q(z) \right] && \text{by Jensen's inequality}
\end{aligned}$$

Bayesian inference in latent variable models

Lower bound the marginal likelihood

- Next idea: lower bound the marginal likelihood with a more tractable form,

$$\begin{aligned}
\log p(x, \Theta) &= \log \int p(x, z, \Theta)\, \mathrm{d}z \\
&= \log \int \frac{q(z)}{q(z)} p(x, z, \Theta)\, \mathrm{d}z && \text{for any distribution } q(z) \\
&= \log \mathbb{E}_{q(z)} \left[\frac{p(x, z, \Theta)}{q(z)} \right] \\
&\geq \mathbb{E}_{q(z)} \left[\log p(x, z, \Theta) - \log q(z) \right] && \text{by Jensen's inequality} \\
&\triangleq \mathscr{L}[q, \Theta]
\end{aligned}$$

- $\mathscr{L}$ is called the **evidence lower bound** or the **ELBO** for short.

Bayesian inference in latent variable models

Coordinate ascent on the ELBO

- Update the parameters,

$$\Theta \leftarrow \arg\max_{\Theta} \mathscr{L}[q, \Theta] = \arg\max_{\Theta} \mathbb{E}_{q(z)}[\log p(x, z, \Theta)]$$

- Update the distribution on latent variables,

$$q \leftarrow \arg\max_{q} \mathscr{L}[q, \Theta]$$

Bayesian inference in latent variable models

Coordinate ascent on the ELBO

- Update the parameters,

$$\Theta \leftarrow \arg\max_{\Theta} \mathscr{L}[q, \Theta] = \arg\max_{\Theta} \mathbb{E}_{q(z)}[\log p(x, z, \Theta)]$$

- Update the distribution on latent variables,

$$\begin{aligned} q &\leftarrow \arg\max_q \mathscr{L}[q, \Theta] \\ &= \arg\max_q \mathbb{E}_{q(z)} \left[\frac{\log p(x, z, \Theta)}{q(z)} \right] \\ &= \arg\min_q \mathrm{KL}\left(q(z) \,\|\, p(z \mid x, \Theta) \right) \\ &= p(z \mid x, \Theta) \end{aligned}$$

Bayesian inference in latent variable models

The Expectation-Maximization (EM) algorithm

- **M-step**: Maximize the expected log probability

$$\Theta \leftarrow \arg\max_{\Theta} \mathbb{E}_{q(z)}[\log p(x, z, \Theta)]$$

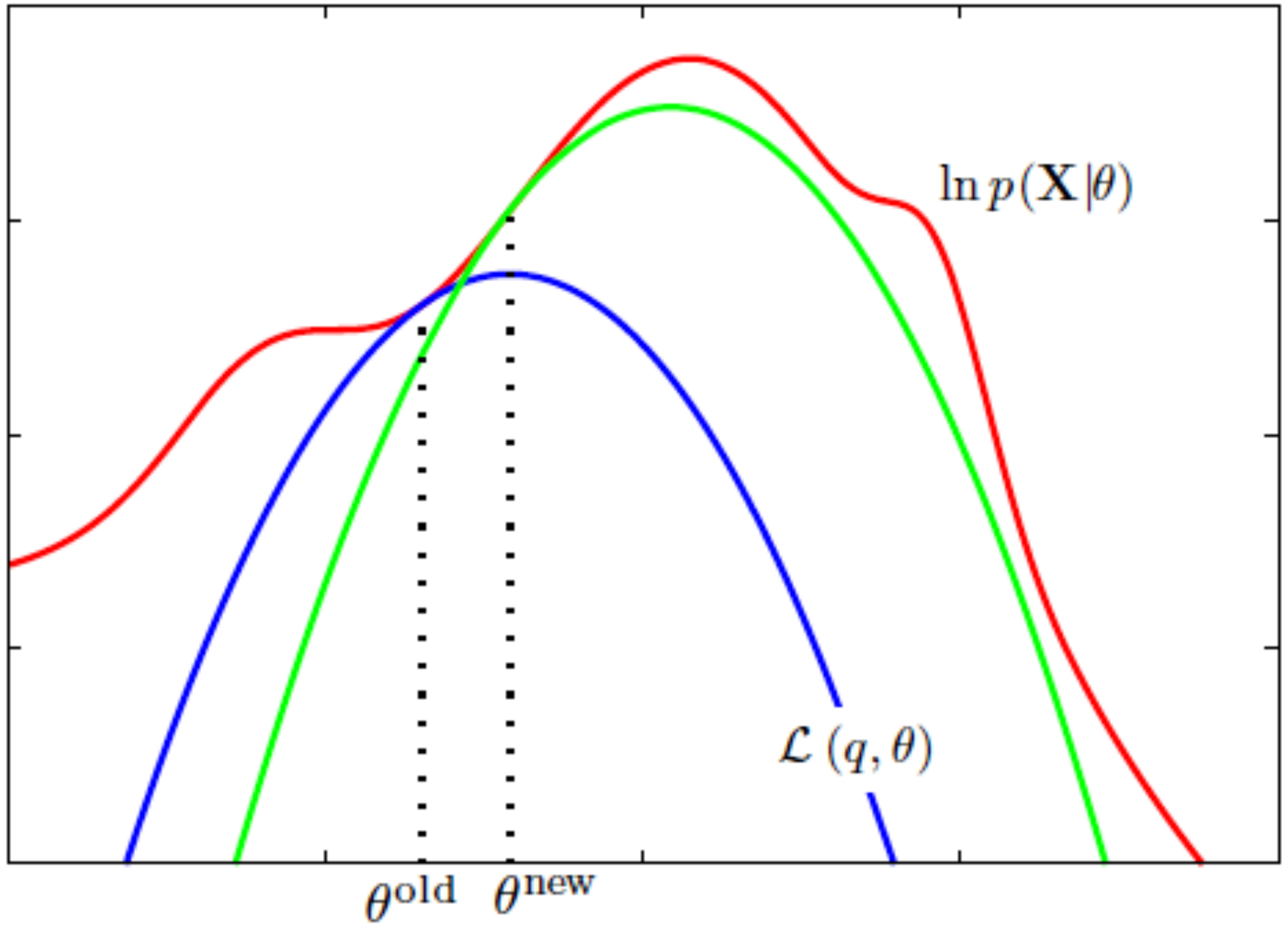
- **E-step**: Update the posterior over latent variables

$$q \leftarrow p(z \mid x, \Theta)$$

- After each E-step, the **ELBO is tight**:

$$\begin{aligned} \mathscr{L}[q, \Theta] &= \mathbb{E}_{p(z|x,\Theta)}\left[\log \frac{p(x, z, \Theta)}{p(z \mid x, \Theta)}\right] \\ &= \mathbb{E}_{p(z|x,\Theta)}\left[\log p(x, \Theta)\right] \\ &= \log p(x, \Theta) \end{aligned}$$

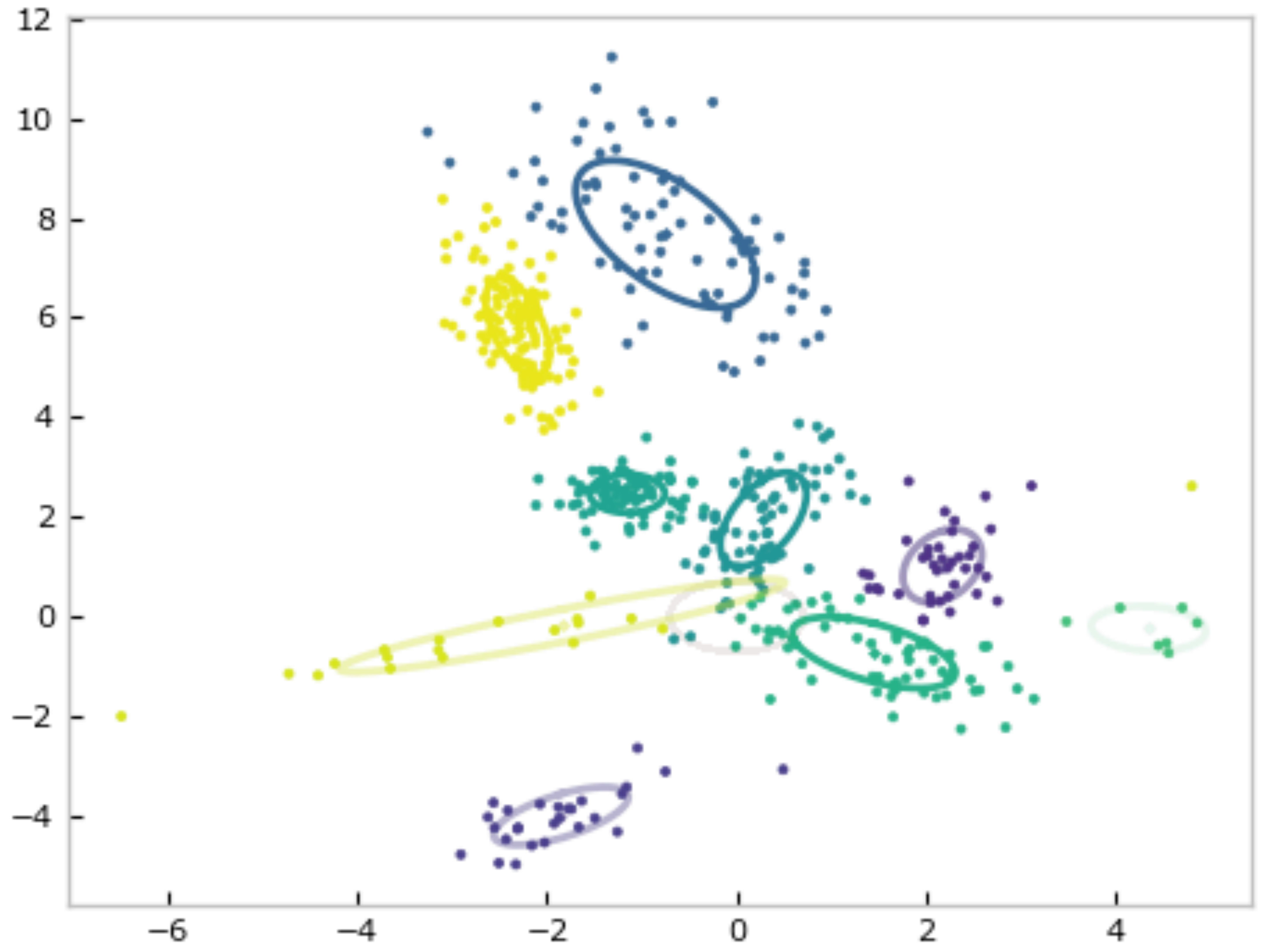
- EM converges to **local optima** of the marginal distribution.



Bishop (2006). Pattern Recognition and Machine Learning, Ch 9.4.

The Gaussian Mixture Model

Example draw from a 2D GMM with 10 clusters



EM for the Gaussian mixture model

- **E-step**: Update the posterior over latent variables,

$$q(z_t = k) \leftarrow p(z_t = k \mid x_t, \Theta) = \frac{\pi_k \mathcal{N}(x_t \mid b_k, Q_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(x_t \mid b_j, Q_j)}$$

- **M-step**: Update the parameters. Let $T_k = \sum_{t=1}^{T} q(z_t = k)$, then

$$\pi_k \leftarrow \frac{T_k}{T}, \qquad b_k \leftarrow \frac{1}{T_k} \sum_{t=1}^{T} q(z_t = k)\, x_t, \qquad Q_k \leftarrow \frac{1}{T_k} \sum_{t=1}^{T} q(z_t = k)\, (x_t - b_k)(x_t - b_k)^\top .$$

  i.e. set the parameters to their weighted averages.

- **Compare** these updates to the MAP estimation / coordinate ascent updates from before!

Hidden Markov Models

The Gaussian HMM

A Gaussian HMM is just a Gaussian mixture model but where cluster assignments are linked across time!

$$
\begin{aligned}
z_1 &\sim \mathrm{Cat}(\pi), \\
z_t \mid z_{t-1} &\sim \mathrm{Cat}(P_{z_{t-1}}), \qquad \text{for } t = 2,\ldots,T. \\
x_t \mid z_t &\sim \mathcal{N}(b_{z_t}, Q_{z_t}) \qquad \text{for } t = 1,\ldots,T
\end{aligned}
$$

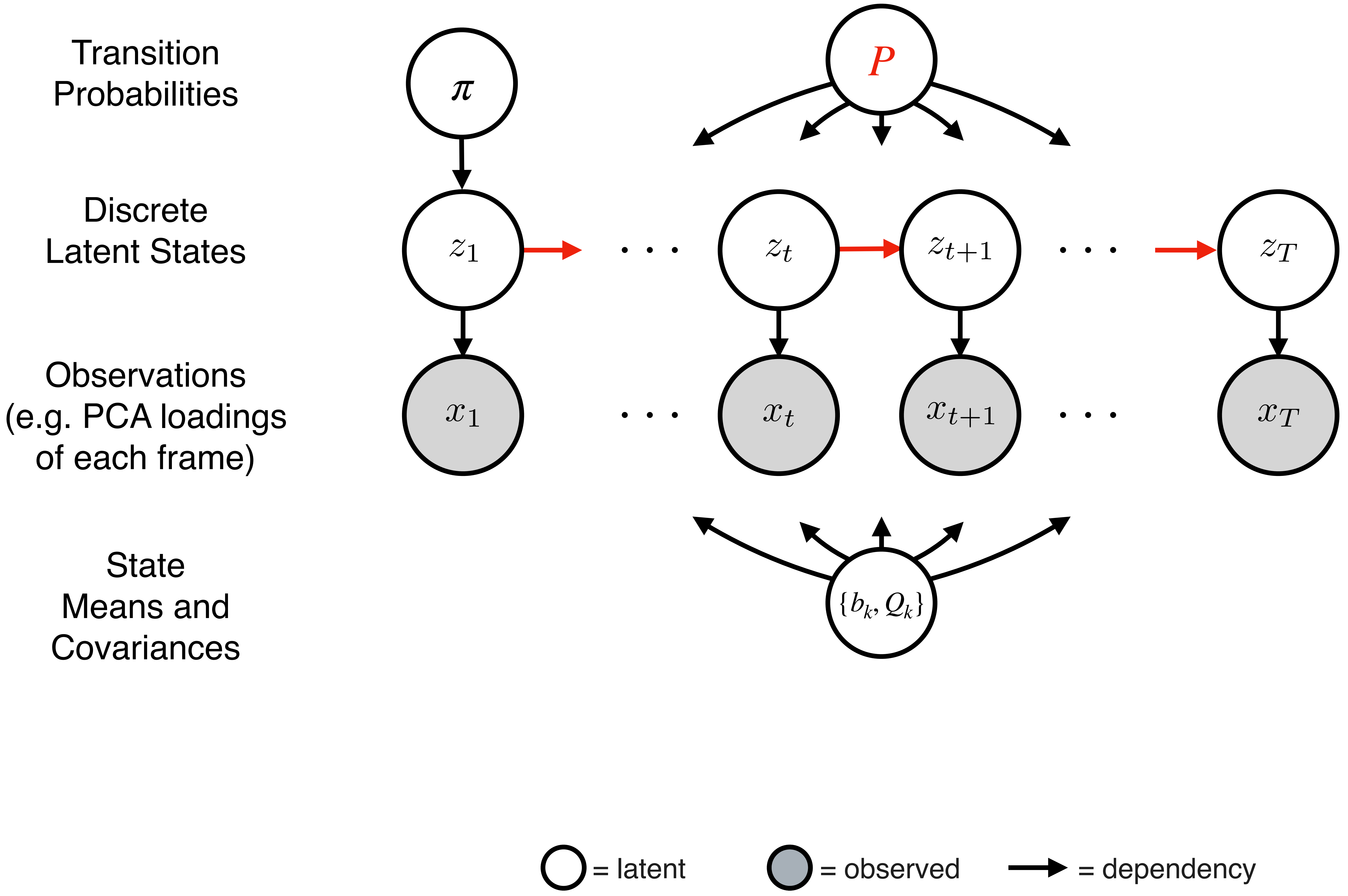
Its parameters are $\Theta = \pi, P, \{b_k, Q_k\}_{k=1}^{K}$ where $P \in [0,1]^{K \times K}$ is a row-stochastic **transition matrix.**

Under this model, the **joint probability** factors as

$$
p(x, z, \Theta) = p(z_1) \prod_{t=1}^{T-1} p(z_{t+1} \mid z_t) \prod_{t=1}^{T} p(x_t \mid z_t)
$$

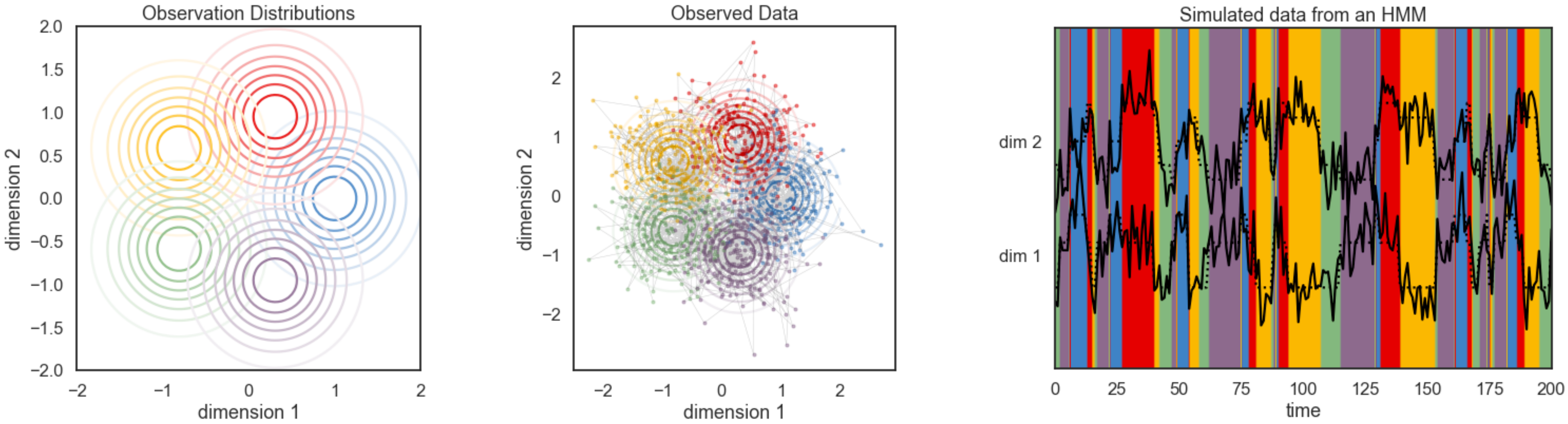
The Gaussian HMM

Graphical Model



The Gaussian HMM

Example draw from a 2D Gaussian HMM with 5 clusters



EM for the Gaussian HMM

The posterior is a little trickier…

- **E-step**: Update the posterior over latent variables,

$$q(z) \leftarrow p(z \mid x, \Theta) \propto p(x, z, \Theta) = p(z_1) \prod_{t=1}^{T-1} p(z_{t+1} \mid z_t) \prod_{t=1}^{T} p(x_t \mid z_t)$$

- The normalized posterior no longer has a simple **closed form!**

- However, we can still **efficiently compute** the **marginal probabilities** for the **M-step**.

EM for the Gaussian HMM

Computing the marginal likelihood

- Consider the marginal probability of state k at time t:

$$q(z_t = k) = \sum_{z_1=1}^{K} \cdots \sum_{z_{t-1}=1}^{K} \sum_{z_{t+1}=1}^{K} \cdots \sum_{z_T=1}^{K} q(z_, \ldots, z_{t-1}, z_t = k, z_{t+1}, \ldots, z_T)$$

EM for the Gaussian HMM

Computing the marginal likelihood

- Consider the marginal probability of state k at time t:

$$q(z_t = k) = \sum_{z_1=1}^{K} \cdots \sum_{z_{t-1}=1}^{K} \sum_{z_{t+1}=1}^{K} \cdots \sum_{z_T=1}^{K} q(z_, \ldots, z_{t-1}, z_t = k, z_{t+1}, \ldots, z_T)$$

$$\propto \left[\sum_{z_1=1}^{K} \cdots \sum_{z_{t-1}=1}^{K} p(z_1) \prod_{s=1}^{t-1} p(x_s \mid z_s)\, p(z_{s+1} \mid z_s) \right] \times \left[p(x_t \mid z_t) \right]$$

$$\times \left[\sum_{z_{t+1}=1}^{K} \cdots \sum_{z_T=1}^{K} \prod_{u=t+1}^{T} p(z_u \mid z_{u-1})\, p(x_u \mid z_u) \right]$$

EM for the Gaussian HMM

Computing the marginal likelihood

- Consider the marginal probability of state k at time t:

$$
\begin{aligned}
q(z_t = k) &= \sum_{z_1=1}^{K} \cdots \sum_{z_{t-1}=1}^{K} \sum_{z_{t+1}=1}^{K} \cdots \sum_{z_T=1}^{K} q(z_{,}\ldots, z_{t-1}, z_t = k, z_{t+1}, \ldots, z_T) \\
&\propto \left[\sum_{z_1=1}^{K} \cdots \sum_{z_{t-1}=1}^{K} p(z_1) \prod_{s=1}^{t-1} p(x_s \mid z_s)\, p(z_{s+1} \mid z_s) \right] \times \Big[p(x_t \mid z_t) \Big] \\
&\quad \times \left[\sum_{z_{t+1}=1}^{K} \cdots \sum_{z_T=1}^{K} \prod_{u=t+1}^{T} p(z_u \mid z_{u-1})\, p(x_u \mid z_u) \right] \\
&\triangleq \alpha_t(z_t) \times p(x_t \mid z_t) \times \beta_t(z_t)
\end{aligned}
$$

EM for the Gaussian HMM

Computing the forward messages $\alpha_t(z_t)$

- Consider the **forward messages**:

$$\alpha_t(z_t) \triangleq \sum_{z_1=1}^{K} \cdots \sum_{z_{t-1}=1}^{K} p(z_1) \prod_{s=1}^{t-1} p(x_s \mid z_s)\, p(z_{s+1} \mid z_s)$$

EM for the Gaussian HMM

Computing the forward messages $\alpha_t(z_t)$

- Consider the **forward messages**:

$$\alpha_t(z_t) \triangleq \sum_{z_1=1}^{K} \cdots \sum_{z_{t-1}=1}^{K} p(z_1) \prod_{s=1}^{t-1} p(x_s \mid z_s)\, p(z_{s+1} \mid z_s)$$

$$= \sum_{z_{t-1}=1}^{K} \left[\left(\sum_{z_1=1}^{K} \cdots \sum_{z_{t-2}=1}^{K} p(z_1) \prod_{s=1}^{t-2} p(x_s \mid z_s) p(z_{s+1} \mid z_s) \right) p(x_{t-1} \mid z_{t-1})\, p(z_t \mid z_{t-1}) \right]$$

EM for the Gaussian HMM

Computing the forward messages $\alpha_t(z_t)$

- Consider the forward messages:

$$\alpha_t(z_t) \triangleq \sum_{z_1=1}^{K} \cdots \sum_{z_{t-1}=1}^{K} p(z_1) \prod_{s=1}^{t-1} p(x_s \mid z_s)\, p(z_{s+1} \mid z_s)$$

$$= \sum_{z_{t-1}=1}^{K} \left[\left(\sum_{z_1=1}^{K} \cdots \sum_{z_{t-2}=1}^{K} p(z_1) \prod_{s=1}^{t-2} p(x_s \mid z_s) p(z_{s+1} \mid z_s) \right) p(x_{t-1} \mid z_{t-1})\, p(z_t \mid z_{t-1}) \right]$$

$$= \sum_{z_{t-1}=1}^{K} \alpha_{t-1}(z_{t-1})\, p(x_{t-1} \mid z_{t-1})\, p(z_t \mid z_{t-1})$$

- We can compute these messages **recursively!**

EM for the Gaussian HMM

Computing the forward messages $\alpha_t(z_t)$. Vectorized.

- Let $\alpha_t = [\alpha_t(z_t = 1), \ldots, \alpha_t(z_t = K)]^\top$ denote the column vector of forward messages. Then,

$$\alpha_t = P^\top(\alpha_{t-1} \odot \ell_{t-1})$$

  where

  - $\ell_{t-1} = [p(x_{t-1} \mid z_{t-1} = 1), \ldots, p(x_{t-1} \mid z_{t-1} = K)]^\top$ is the vector of likelihoods,
  - $\odot$ denotes the element-wise product, and
  - P is the transition matrix with $P_{ij} = p(z_t = j \mid z_{t-1} = i)$.

- For the base case, let $\alpha_1(z_1) = p(z_1)$.

EM for the Gaussian HMM

Computing the backward messages $\beta_t(z_t)$

- Now take the **backward messages**:

$$\beta_t(z_t) \triangleq \sum_{z_{t+1}=1}^{K} \cdots \sum_{z_T=1}^{K} \prod_{u=t+1}^{T} p(z_u \mid z_{u-1})\, p(x_u \mid z_u)$$

EM for the Gaussian HMM

Computing the backward messages $\beta_t(z_t)$

- Now take the **backward messages**:

$$\beta_t(z_t) \triangleq \sum_{z_{t+1}=1}^{K} \cdots \sum_{z_T=1}^{K} \prod_{u=t+1}^{T} p(z_u \mid z_{u-1})\, p(x_u \mid z_u)$$

$$= \sum_{z_{t+1}=1}^{K} p(z_{t+1} \mid z_t)\, p(x_{t+1} \mid z_{t+1}) \sum_{z_{t+2}=1}^{K} \cdots \sum_{z_T=1}^{K} \prod_{u=t+2}^{T} p(z_u \mid z_{u-1})\, p(x_u \mid z_u)$$

EM for the Gaussian HMM

Computing the backward messages $\beta_t(z_t)$

- Now take the **backward messages**:

$$\beta_t(z_t) \triangleq \sum_{z_{t+1}=1}^{K} \cdots \sum_{z_T=1}^{K} \prod_{u=t+1}^{T} p(z_u \mid z_{u-1})\, p(x_u \mid z_u)$$

$$= \sum_{z_{t+1}=1}^{K} p(z_{t+1} \mid z_t)\, p(x_{t+1} \mid z_{t+1}) \sum_{z_{t+2}=1}^{K} \cdots \sum_{z_T=1}^{K} \prod_{u=t+2}^{T} p(z_u \mid z_{u-1})\, p(x_u \mid z_u)$$

$$= \sum_{z_{t+1}=1}^{K} p(z_{t+1} \mid z_t)\, p(x_{t+1} \mid z_{t+1})\, \beta_{t+1}(z_{t+1})$$

- Again, we can compute the backward messages recursively!

EM for the Gaussian HMM

Computing the backward messages $\beta_t(z_t)$. Vectorized.

- Let $\beta_t = [\beta_t(z_t = 1), \ldots, \beta_t(z_t = K)]^\top$ denote the column vector of backward messages. Then,

$$\beta_t = P(\beta_{t+1} \odot \ell_{t+1})$$

- For the base case, let $\beta_T(z_T) = 1$.

EM for the Gaussian HMM

Combining the forward and backward messages

- The posterior marginal probability of state k at time t is,

$$
\begin{aligned}
q(z_t = k) &\propto \alpha_t(z_t = k) \times p(x_t \mid z_t = k) \times \beta_t(z_t = k) \\
&= \alpha_{tk} \ell_{tk} \beta_{tk}
\end{aligned}
$$

- The probabilities need to sum to one. Normalizing yields,

$$
q(z_t = k) = \frac{\alpha_{tk} \ell_{tk} \beta_{tk}}{\sum_{j=1}^{K} \alpha_{tj} \ell_{tj} \beta_{tj}}
$$

- Finally, note the marginal is invariant to multiplying α_t and/or β_t by a constant.

EM for the Gaussian HMM

Normalizing the messages to prevent underflow

- The messages involve **products of probabilities**, which quickly underflow.
- We can leverage the scale invariance to renormalize the messages. I.e. replace:

$$\alpha_t = P^\top(\alpha_{t-1} \odot \ell_{t-1}) \quad \text{with} \quad \begin{aligned} A_{t-1} &= \textstyle\sum_k \tilde{\alpha}_{t-1,k} \ell_{t-1,k} \\ \tilde{\alpha}_t &= \tfrac{1}{A_{t-1}} P^\top(\tilde{\alpha}_{t-1} \odot \ell_{t-1}) \end{aligned}$$

where $\tilde{\alpha}_t$ are normalized for numerical stability. As before, $\tilde{\alpha}_1 = \pi$.

- This lends a nice **interpretation**: the **forward messages are conditional probabilities** $\tilde{\alpha}_{tk} = p(z_t = k \mid x_{1:t-1})$ and the **normalization constants are the marginal likelihoods** $A_t = p(x_t \mid x_{1:t-1})$.

EM for the Gaussian HMM

Computing the marginal likelihood

- Finally, we can compute the marginal likelihood alongside the forward messages

$$\log p(x \mid \Theta) = \log \sum_{z_1=1}^{K} \cdots \sum_{z_T=1}^{K} \left[p(z_1) \prod_{t=1}^{T-1} p(z_{t+1} \mid z_t) \prod_{t=1}^{T} p(x_t \mid z_t) \right]$$

$$= \log \sum_{z_T=1}^{K} \alpha_T(z_T)\, p(x_T \mid z_T)$$

$$= \log \prod_{t=1}^{T} A_t = \sum_{t=1}^{T} \log A_t$$

- Again, makes sense since the normalization constants are $A_t = p(x_t \mid x_{1:t-1})$.

EM for the Gaussian HMM

Putting it all together

- **E-step**: Run the forward-backward algorithm to compute

$$q(z_t = k) \leftarrow p(z_t = k \mid x_{1:T}, \Theta) = \frac{\alpha_{tk}\ell_{tk}\beta_{tk}}{\sum_{j=1}^{K} \alpha_{tj}\ell_{tj}\beta_{tj}}$$ and the marginal log likelihood $\log p(x_{1:T} \mid \Theta)$.

- **M-step**: Update the parameters.

$$T_k = \sum_{t=1}^{T} q(z_t = k) \qquad b_k = \frac{1}{T_k}\sum_{t=1}^{T} q(z_t = k)x_t \qquad Q_k = \frac{1}{T_k}\sum_{t=1}^{T} q(z_t = k)(x_t - b_k)(x_t - b_k)^\top$$

- ***Note:*** *You can use the forward-backward algorithm to compute* $q(z_t = i, z_{t+1} = j)$ *too. That's all you need to update the transition matrix* P.

Conclusion

- EM for mixture models (with exponential family likelihoods) amounts to **computing cluster assignment probabilities** and **expected sufficient statistics**, then updating parameters based on them.

- **Stochastic EM** generalizes this approach to work with mini-batches of data.

- Hidden Markov models (HMMs) are just mixture models with dependencies across time.

- The EM algorithm is nearly the same, but we use the **forward-backward algorithm** to compute latent state probabilities and expected sufficient stats.