

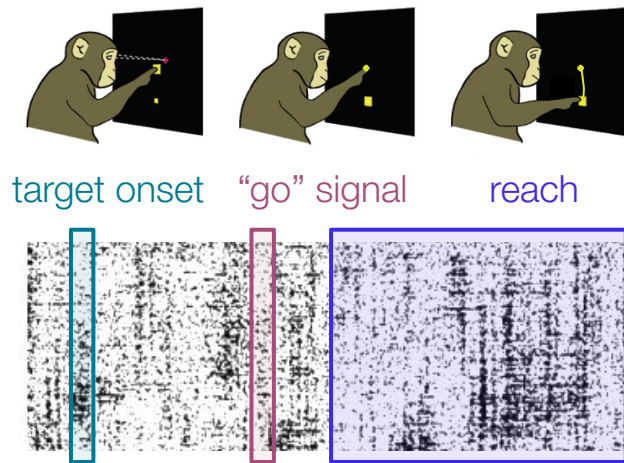
# Latent variable models

Expectation maximization, mixture models, hidden Markov models

BAMB! '25

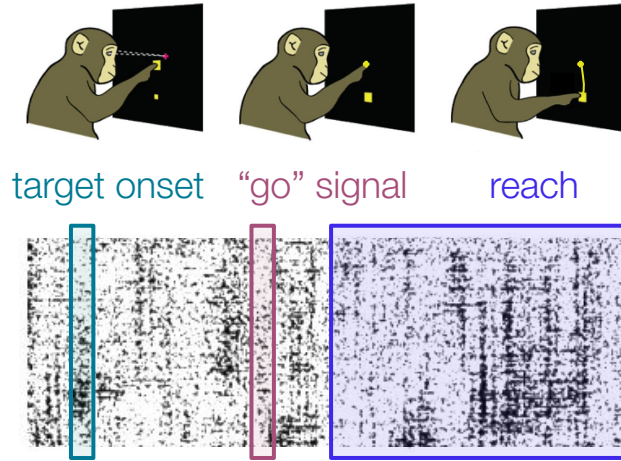
Heike Stein

# Behavior in the lab vs. behavior “in the wild”

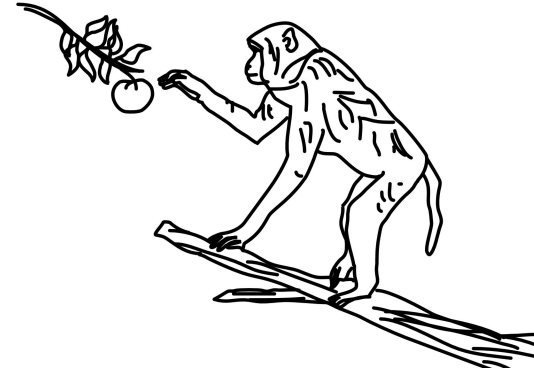


Behavior in the lab

# Behavior in the lab vs. behavior “in the wild”

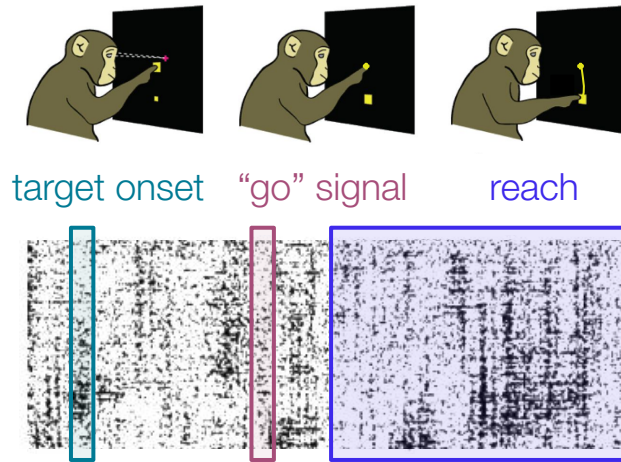


Behavior in the lab

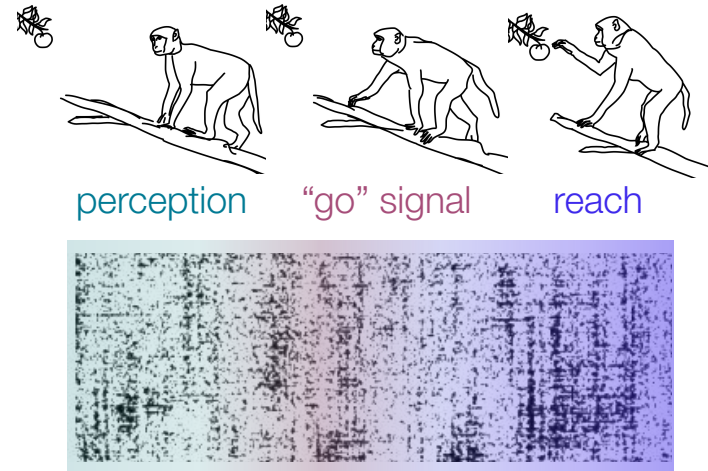


Natural behavior

# Behavior in the lab vs. behavior “in the wild”

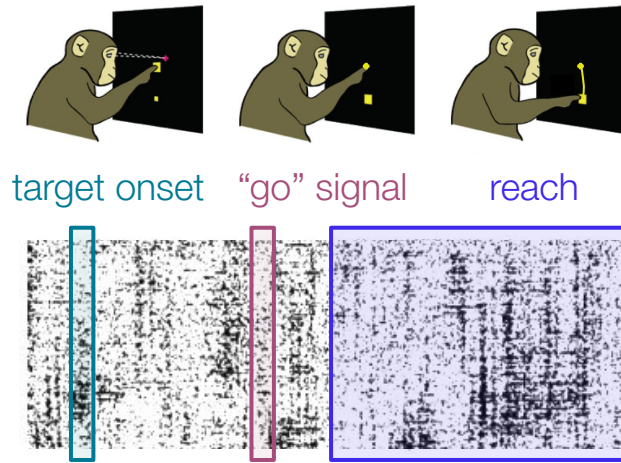


Behavior in the lab

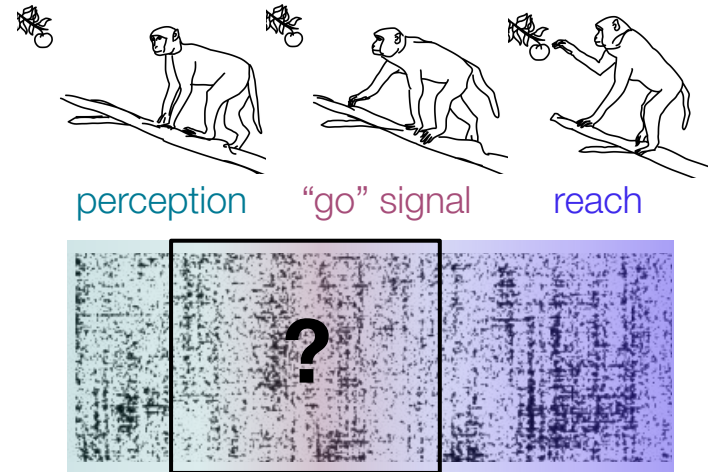


Natural behavior

# Behavior in the lab vs. behavior “in the wild”



Behavior in the lab



Natural behavior

Even in strictly controlled lab-based tasks, we can see unexpected variability in behavior



Block prior

20:80

50:50

80:20



Even in strictly controlled lab-based tasks, we can see unexpected variability in behavior



Block prior



**Only 50:50 trials!**












nature  
neuroscience

ARTICLES

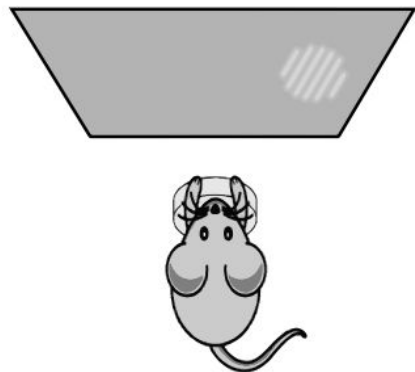
<https://doi.org/10.1038/s41593-021-01007-z>

Check for updates

## Mice alternate between discrete strategies during perceptual decision-making

Zoe C. Ashwood<sup>1,2</sup>  , Nicholas A. Roy<sup>2</sup>, Iris R. Stone<sup>2</sup>  , The International Brain Laboratory\*, Anne E. Urai<sup>3</sup>  , Anne K. Churchland<sup>4</sup>  , Alexandre Pouget<sup>5</sup>   and Jonathan W. Pillow<sup>2,6</sup>  

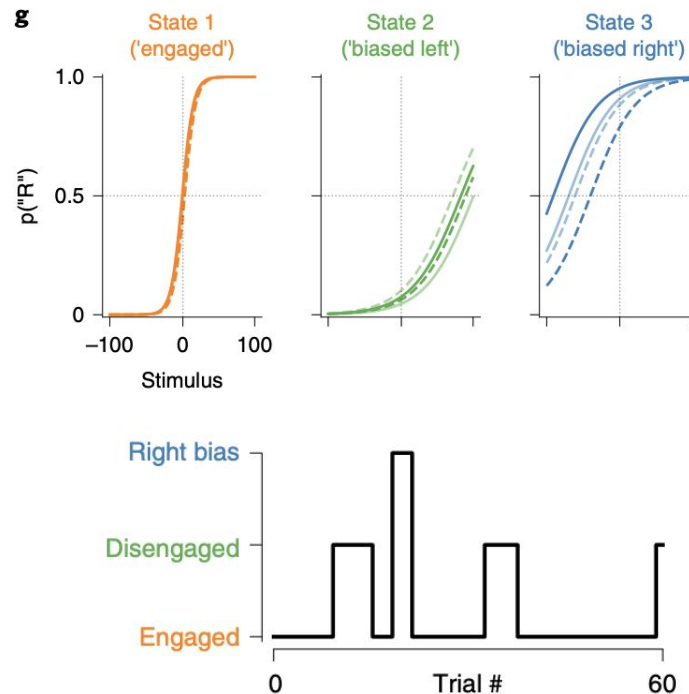
Even in strictly controlled lab-based tasks, we can see unexpected variability in behavior



Block prior



**Only 50:50 trials!**





# Latent variable models

**Problem:** How to deal with uninstructed variability in behavioral patterns?

# Latent variable models

**Problem:** How to deal with uninstructed variability in behavioral patterns?

**Solution:** Latent variable models use hypothetical “helper” variables (*latent variables*) to find structure underlying different behavioral patterns

# Latent variable models

**Problem:** How to deal with uninstructed variability in behavioral patterns?

**Solution:** Latent variable models use hypothetical “helper” variables (*latent variables*) to find structure underlying different behavioral patterns

→ We use probabilistic methods (e.g. Bayesian inference) to find latents

→ This will work if the data is explained by a mix of simpler models

# Latent variable models

**Problem:** How to deal with uninstructed variability in behavioral patterns?

**Solution:** Latent variable models use hypothetical “helper” variables (*latent variables*) to find structure underlying different behavioral patterns

→ We use probabilistic methods (e.g. Bayesian inference) to find latents

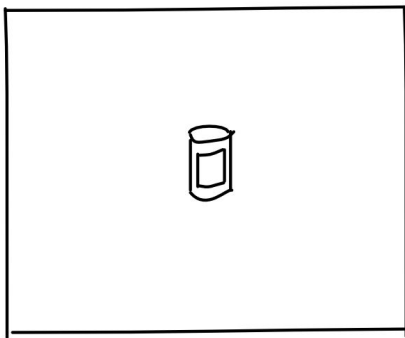
→ This will work if the data is explained by a mix of simpler models

**What this buys us:** We can fit and interpret simple models despite unexpected changes in data patterns

# General intro: Probabilistic models

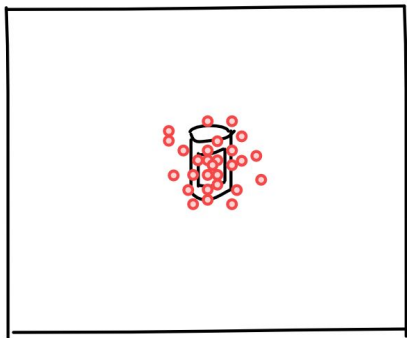
# Stochasticity

Behavioral and neural measurements are inherently noisy.



# Stochasticity

Behavioral and neural measurements are inherently noisy.

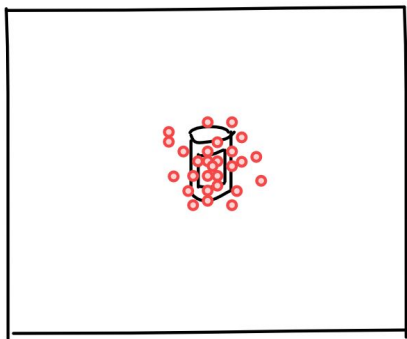


# Stochasticity and probabilistic modeling

Behavioral and neural measurements are inherently noisy.

Probabilistic models specify a *noise model* that

- (1) quantifies variability, and
- (2) uses it for computation under uncertainty



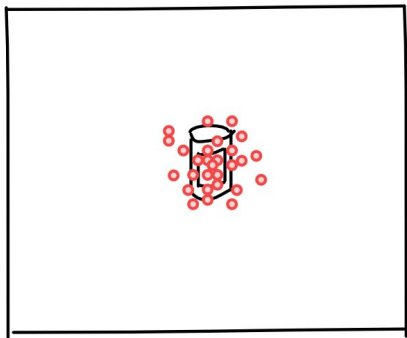


# Stochasticity and probabilistic modeling

Behavioral and neural measurements are inherently noisy.

Probabilistic models specify a *noise model* that

- (1) quantifies variability, and
- (2) uses it for computation under uncertainty

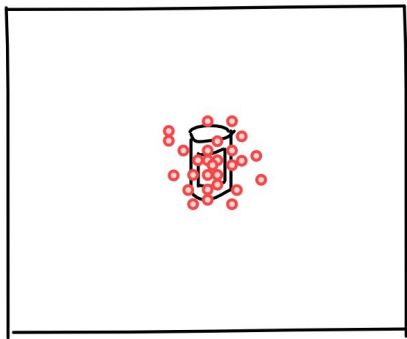


*Inference* is merely a statistical process of estimating variables or parameters.

It is not assumed to happen in the brain.

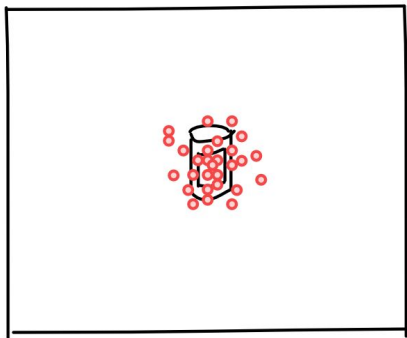
# Random variables

A variable  $X$  whose value is not deterministic. Its realizations are called *observations  $x$*



# Random variables

A variable  $X$  whose value is not deterministic. Its realizations are called *observations*  $x$

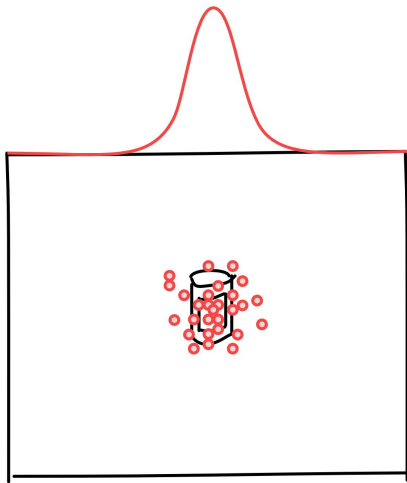


observations  $x \in \mathbb{R}^2$

( $X$  is a 2-dimensional,  
real-valued random variable)

# Random variables and distributions

A variable  $X$  whose value is not deterministic. Its realizations are called *observations*  $x$

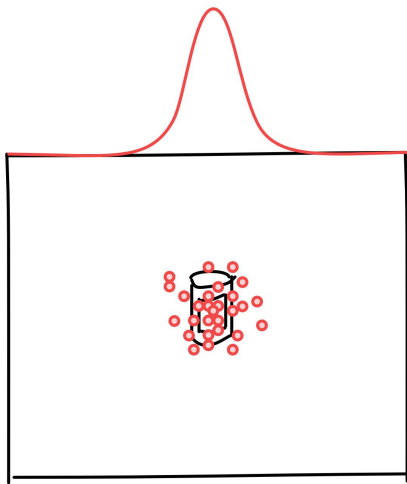


Observations  $x \in \mathbb{R}^2$  are distributed according to a Gaussian probability density func.

$$f_X(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$

# Random variables and distributions

A variable  $X$  whose value is not deterministic. Its realizations are called *observations*  $x$

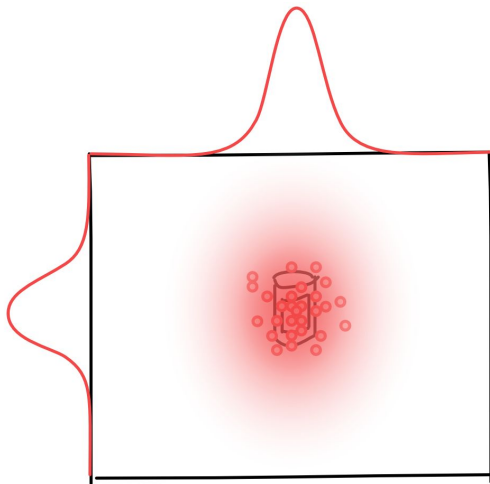


Observations  $x \in \mathbb{R}^2$  are distributed according to a Gaussian probability density func.

$$f_X(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$

# Random variables and distributions

A variable  $X$  whose value is not deterministic. Its realizations are called *observations*  $x$



Observations  $x \in \mathbb{R}^2$  are distributed according to a *multivariate Gaussian* pdf

$$f_X(\mathbf{x}) = \frac{1}{(2\pi)^{\frac{d}{2}} \Sigma^{\frac{1}{2}}} \exp \left( -\frac{1}{2} (\mathbf{x} - \mu)^\top \Sigma^{-1} (\mathbf{x} - \mu) \right)$$

# Multivariate distributions

Multivariate distributions are *joint distributions* over random variables

$\mathbf{X} = (X_1, X_2, \dots, X_d)$ , with a joint PMF or PDF  $f_{X_1, X_2, \dots, X_d}(x_1, x_2, \dots, x_d)$

# Multivariate distributions

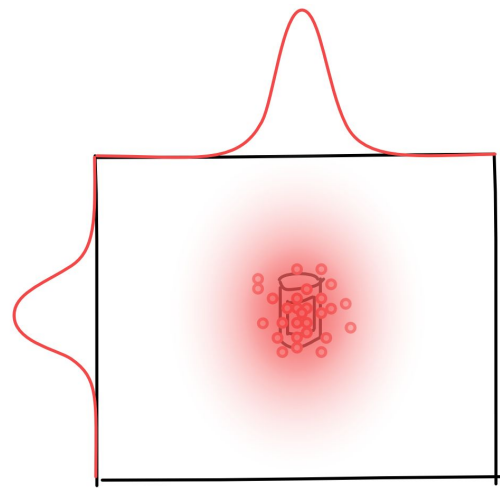
Multivariate distributions are *joint distributions* over random variables

$\mathbf{X} = (X_1, X_2, \dots, X_d)$ , with a joint PMF or PDF  $f_{X_1, X_2, \dots, X_d}(x_1, x_2, \dots, x_d)$

For the multivariate Gaussian

$$f_{\mathbf{X}}(\mathbf{x}) = \frac{1}{(2\pi)^{\frac{d}{2}} \Sigma^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(\mathbf{x} - \mu)^{\top} \Sigma^{-1}(\mathbf{x} - \mu)\right),$$

the covariance matrix  $\Sigma$  describes dependencies between variables





# Multivariate distributions

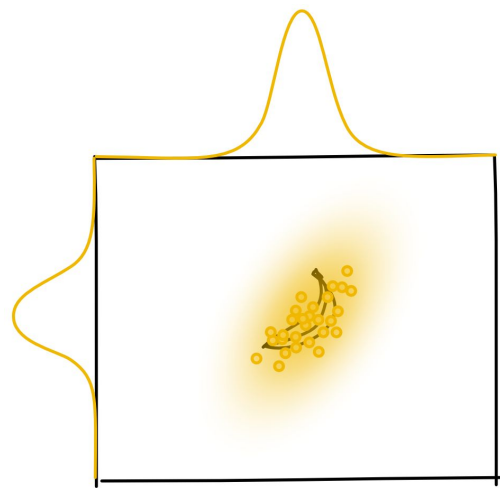
Multivariate distributions are *joint distributions* over random variables

$\mathbf{X} = (X_1, X_2, \dots, X_d)$ , with a joint PMF or PDF  $f_{X_1, X_2, \dots, X_d}(x_1, x_2, \dots, x_d)$

For the multivariate Gaussian

$$f_{\mathbf{X}}(\mathbf{x}) = \frac{1}{(2\pi)^{\frac{d}{2}} \Sigma^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(\mathbf{x} - \mu)^{\top} \Sigma^{-1}(\mathbf{x} - \mu)\right),$$

the covariance matrix  $\Sigma$  describes dependencies between variables



# Multivariate distributions

Multivariate distributions are *joint distributions* over random variables

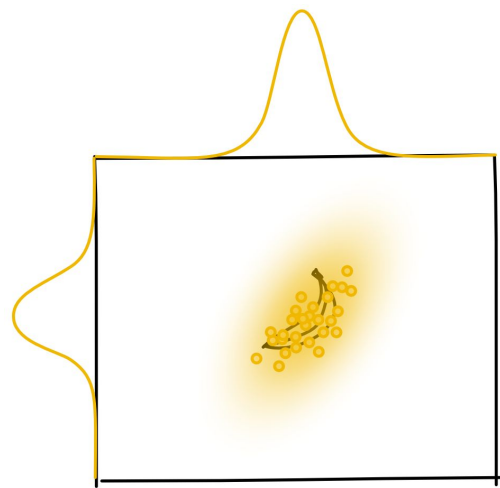
$\mathbf{X} = (X_1, X_2, \dots, X_d)$ , with a joint PMF or PDF  $f_{X_1, X_2, \dots, X_d}(x_1, x_2, \dots, x_d)$

For the multivariate Gaussian

$$f_{\mathbf{X}}(\mathbf{x}) = \frac{1}{(2\pi)^{\frac{d}{2}} \Sigma^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(\mathbf{x} - \mu)^{\top} \Sigma^{-1}(\mathbf{x} - \mu)\right),$$

the covariance matrix  $\Sigma$  describes dependencies between variables

$$\Sigma = \begin{pmatrix} \sigma_1^2 & \sigma_{12} \\ \sigma_{12} & \sigma_2^2 \end{pmatrix} = \begin{pmatrix} \sigma_1^2 & \rho\sigma_1\sigma_2 \\ \rho\sigma_1\sigma_2 & \sigma_2^2 \end{pmatrix}$$



# The likelihood function in probabilistic terms

For different parameter values, how likely is the observed data?

$$L(\theta \mid \mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n)$$

# The likelihood function in probabilistic terms

For different parameter values, how likely is the observed data?

$$L(\theta \mid \mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n)$$

What is the *probability of observations  $\mathbf{x}$ , given the parameters  $\theta$* ?

$$L(\theta \mid \mathbf{x}) = p(\mathbf{x} \mid \theta)$$

# The likelihood function in probabilistic terms

For different parameter values, how likely is the observed data?

$$L(\theta \mid \mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n)$$

What is the *probability of observations  $\mathbf{x}$ , given the parameters  $\theta$* ?

$$L(\theta \mid \mathbf{x}) = p(\mathbf{x} \mid \theta)$$

→ different parameter values represent *different hypotheses* about the model

→ the likelihood is the *evidence* for each hypothesis

# Bayesian inference: Key ingredients

- the *likelihood*  $p(\mathbf{x} \mid \theta)$ : evidence for our hypothesis about  $\theta$

# Bayesian inference: Key ingredients

- the *likelihood*  $p(\mathbf{x} \mid \theta)$ : evidence for our hypothesis about  $\theta$
- the *prior distribution*  $p(\theta)$ : before observing  $\mathbf{x}$ , what's our belief and certainty about  $\theta$

# Bayesian inference: Key ingredients

- the *likelihood*  $p(\mathbf{x} \mid \theta)$ : evidence for our hypothesis about  $\theta$
- the *prior distribution*  $p(\theta)$ : before observing  $\mathbf{x}$ , what's our belief and certainty about  $\theta$
- the *posterior*  $p(\theta \mid \mathbf{x})$ : after observing  $\mathbf{x}$ , what's our updated belief about  $\theta$



# Bayesian inference: Key ingredients

- the *likelihood*  $p(\mathbf{x} \mid \theta)$ : evidence for our hypothesis about  $\theta$
- the *prior distribution*  $p(\theta)$ : before observing  $\mathbf{x}$ , what's our belief and certainty about  $\theta$
- the posterior  $p(\theta \mid \mathbf{x})$ : after observing  $\mathbf{x}$ , what's our updated belief about  $\theta$
- the marginal likelihood  $p(\mathbf{x}) = \int p(\mathbf{x}|\theta)p(\theta)d\theta$ : For any value  $\theta$  might take, what is the total evidence we have for our model?

# Bayesian inference: Combining the likelihood with a prior

In Bayesian statistics, the *likelihood*  $p(\mathbf{x} \mid \theta)$  is regarded as *evidence* in favor of a specific parameter set  $\theta$ . We combine it with our *prior belief* (*prior distribution*)  $p(\theta)$  about how the parameters are distributed to obtain the *posterior distribution*  $p(\theta \mid \mathbf{x})$

$$p(\theta \mid \mathbf{x}) = \frac{p(\mathbf{x} \mid \theta)p(\theta)}{p(\mathbf{x})}$$

# Bayesian inference: Combining the likelihood with a prior

In Bayesian statistics, the *likelihood*  $p(\mathbf{x} \mid \theta)$  is regarded as *evidence* in favor of a specific parameter set  $\theta$ . We combine it with our *prior belief* (*prior distribution*)  $p(\theta)$  about how the parameters are distributed to obtain the *posterior distribution*  $p(\theta \mid \mathbf{x})$

$$p(\theta \mid \mathbf{x}) = \frac{p(\mathbf{x} \mid \theta)p(\theta)}{p(\mathbf{x})}$$

*Bayes' theorem immediately follows from basic rules of probability,*

*specifically the product rule :*  $p(\mathbf{B} \mid \mathbf{A}) = p(\mathbf{A}, \mathbf{B}) / p(\mathbf{A})$

$$p(\mathbf{A}, \mathbf{B}) = p(\mathbf{A} \mid \mathbf{B}) / p(\mathbf{B})$$

# Bayesian inference: Combining the likelihood with a prior

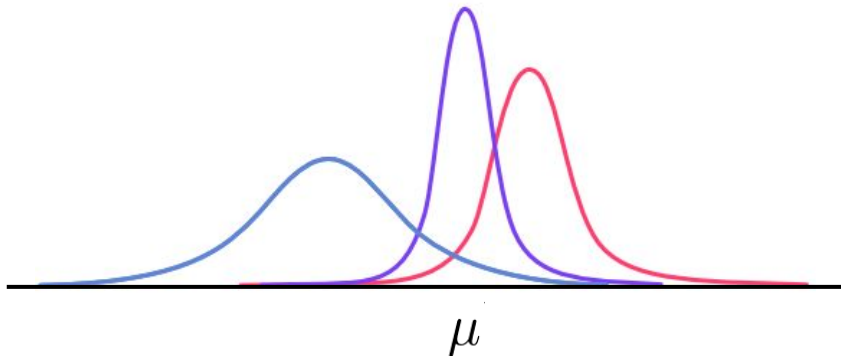
In Bayesian statistics, the *likelihood*  $p(\mathbf{x} \mid \theta)$  is regarded as *evidence* in favor of a specific parameter set  $\theta$ . We combine it with our *prior belief* (*prior distribution*)  $p(\theta)$  about how the parameters are distributed to obtain the *posterior distribution*  $p(\theta \mid \mathbf{x})$

$$p(\theta \mid \mathbf{x}) = \frac{p(\mathbf{x} \mid \theta)p(\theta)}{p(\mathbf{x})}$$

# Bayesian inference: Combining the likelihood with a prior

In Bayesian statistics, the *likelihood*  $p(\mathbf{x} \mid \theta)$  is regarded as *evidence* in favor of a specific parameter set  $\theta$ . We combine it with our *prior belief* (*prior distribution*)  $p(\theta)$  about how the parameters are distributed to obtain the *posterior distribution*  $p(\theta \mid \mathbf{x})$

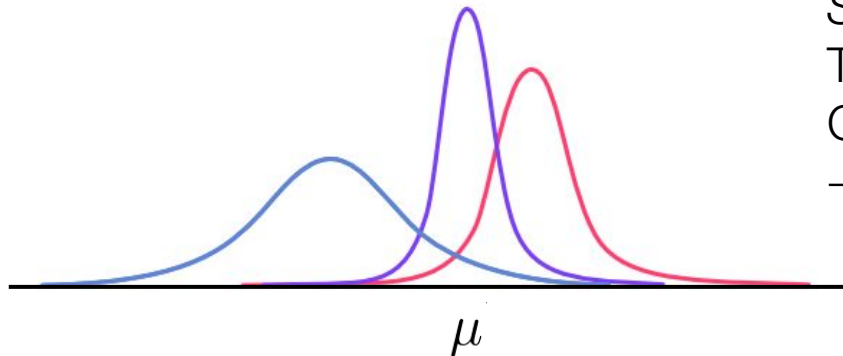
$$p(\theta \mid \mathbf{x}) = \frac{p(\mathbf{x} \mid \theta)p(\theta)}{p(\mathbf{x})}$$



# Bayesian inference: Combining the likelihood with a prior

In Bayesian statistics, the *likelihood*  $p(\mathbf{x} \mid \theta)$  is regarded as *evidence* in favor of a specific parameter set  $\theta$ . We combine it with our *prior belief* (*prior distribution*)  $p(\theta)$  about how the parameters are distributed to obtain the *posterior distribution*  $p(\theta \mid \mathbf{x})$

$$p(\theta \mid \mathbf{x}) = \frac{p(\mathbf{x} \mid \theta)p(\theta)}{p(\mathbf{x})}$$



*Side note:*

The likelihood is not always Gaussian (e.g. for  $\sigma$ )  
→ choose matching, *i.e.*  
*conjugate priors*

# Bayesian inference: Combining the likelihood with a prior

In Bayesian statistics, the *likelihood*  $p(\mathbf{x} \mid \theta)$  is regarded as *evidence* in favor of a specific parameter set  $\theta$ . We combine it with our *prior belief* (*prior distribution*)  $p(\theta)$  about how the parameters are distributed to obtain the *posterior distribution*  $p(\theta \mid \mathbf{x})$

$$p(\theta \mid \mathbf{x}) = \frac{p(\mathbf{x} \mid \theta)p(\theta)}{p(\mathbf{x})}$$

$p(\mathbf{x}) = \int p(\mathbf{x} \mid \theta)p(\theta)d\theta$  is a normalization constant to ensure that  $p(\theta \mid \mathbf{x})$  integrates to 1. It is called the *marginal likelihood* (sometimes also called *evidence*).

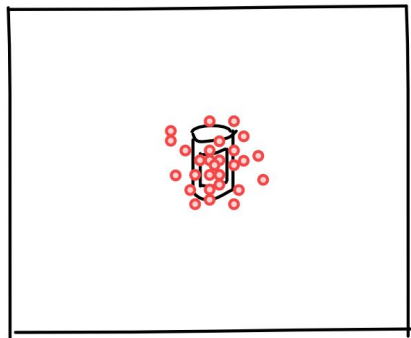
# Generative model

It formalizes our *knowledge* or our *hypothesis* about how data was generated.



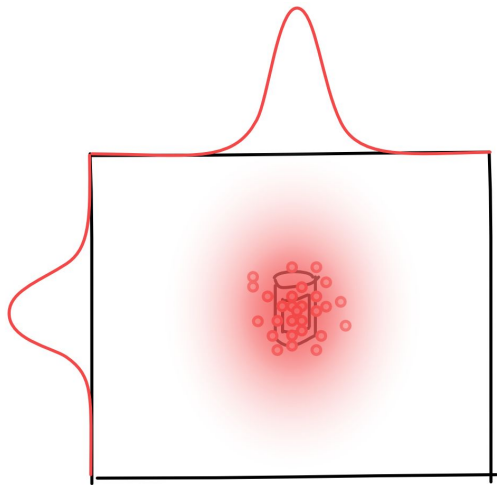
# Generative model

It formalizes our *knowledge* or our *hypothesis* about how data was generated.



# Generative model

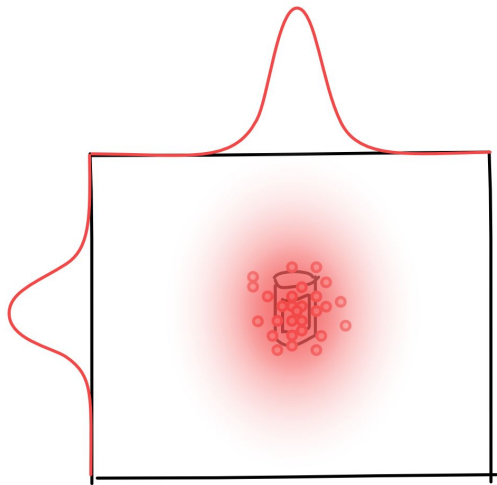
It formalizes our *knowledge* or our *hypothesis* about how data was generated.



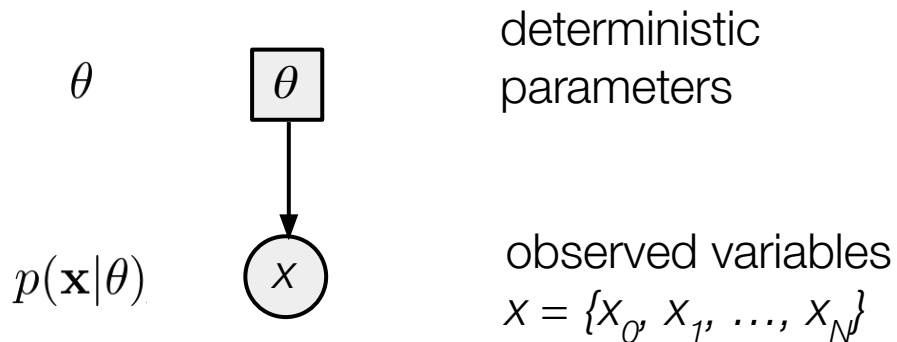
“a multivariate Normal parametrized by  $\theta = \{\mu, \Sigma\}$ “

# Generative model

It formalizes our *knowledge* or our *hypothesis* about how data was generated.

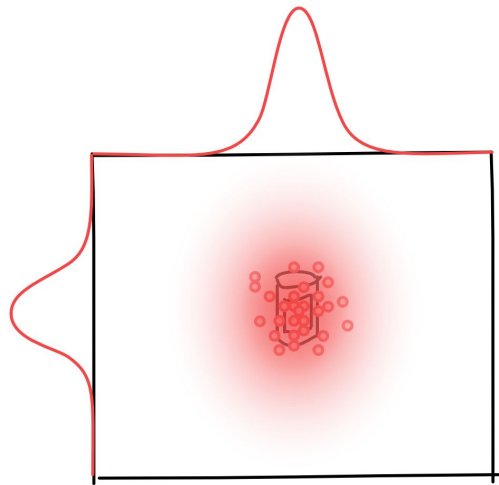


“a multivariate Normal parametrized by  $\theta = \{\mu, \Sigma\}$ ”



# Generative model

It formalizes our *knowledge* or our *hypothesis* about how data was generated.



“a multivariate Normal parametrized by  $\theta = \{\mu, \Sigma\}$ “

$$p(\theta)$$



parameters as  
random variables

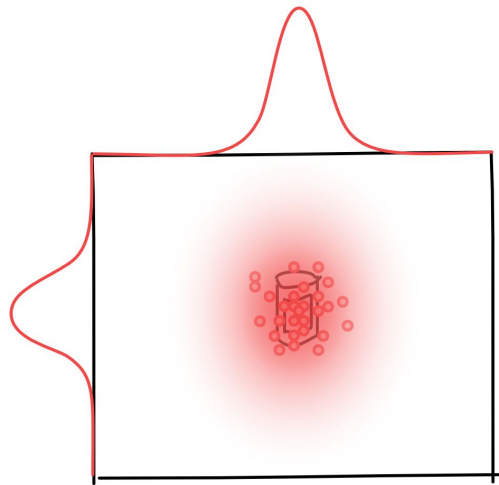
$$p(\mathbf{x}|\theta)$$



observed variables  
 $\mathbf{x} = \{x_0, x_1, \dots, x_N\}$

# Generative model

It formalizes our *knowledge* or our *hypothesis* about how data was generated.



“a multivariate Normal parametrized by  $\theta = \{\mu, \Sigma\}$ “

$$p(\theta)$$

$$p(\mathbf{x}|\theta)$$



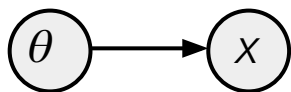
random variables  
with unknown value  
(:= *latent* variable)

observed variables  
 $x = \{x_0, x_1, \dots, x_N\}$

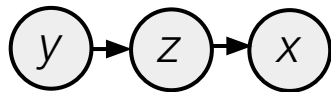
# Generative model: Joint distribution of the “complete dataset”

It formalizes our *knowledge* or our *hypothesis* about how data was generated.

- (1) It allows us to write down the *joint distribution* of all variables in our model:



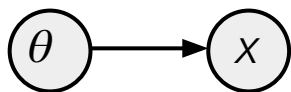
(2)



# Generative model: Joint distribution of the “complete dataset”

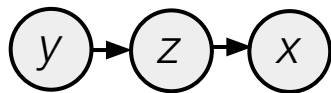
It formalizes our *knowledge* or our *hypothesis* about how data was generated.

- (1) It allows us to write down the *joint distribution* of all variables in our model:



$$(1) \quad p(\mathbf{x}, \theta) = p(\mathbf{x}|\theta)p(\theta)$$

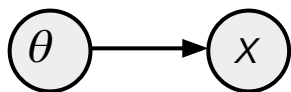
- (2)  $(2) \quad p(\mathbf{x}, \mathbf{z}, \mathbf{y}) = p(\mathbf{x}|\mathbf{z})p(\mathbf{z}|\mathbf{y})p(\mathbf{y})$



# Generative model: Sampling

With the generative model, we can *sample* datasets:

(1)

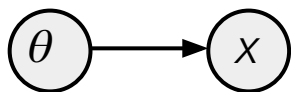




# Generative model: Sampling

With the generative model, we can *sample* datasets:

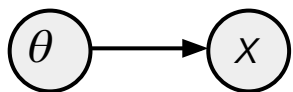
(1)  $p(\mathbf{x}, \theta) = p(\mathbf{x}|\theta)p(\theta)$  with a Gaussian observation model:



# Generative model: Sampling

With the generative model, we can *sample* datasets:

(1)  $p(\mathbf{x}, \theta) = p(\mathbf{x}|\theta)p(\theta)$  with a Gaussian observation model:

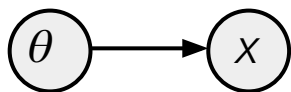


- sample  $\theta = \{\mu, \Sigma\}$  from  $p(\theta)$  (or assume fixed values)

# Generative model: Sampling

With the generative model, we can *sample* datasets:

(1)  $p(\mathbf{x}, \theta) = p(\mathbf{x}|\theta)p(\theta)$  with a Gaussian observation model:

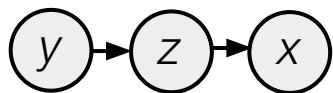


- sample  $\theta = \{\mu, \Sigma\}$  from  $p(\theta)$  (or assume fixed values)
- then, with fixed  $\theta = \{\mu, \Sigma\}$ , sample  $\mathbf{x}$  from  $p(\mathbf{x}|\theta) = \mathcal{N}(\mathbf{x}; \mu, \Sigma)$ :  
 $\mathbf{x}_i | \mu, \Sigma \sim \mathcal{N}(\mu, \Sigma)$

# Generative model: Sampling

With the generative model, we can *sample* datasets:

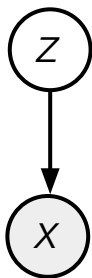
(2)  $p(\mathbf{x}, \mathbf{z}, \mathbf{y}) = p(\mathbf{x}|\mathbf{z})p(\mathbf{z}|\mathbf{y})p(\mathbf{y})$  assuming all variables are binary



- sample  $y$ :  $U \sim \text{Uniform}(0, 1)$   
 $\mathbf{y} = 1_{\{U \leq p_{\mathbf{y}}\}}$
- sample  $z$ :  $P(\mathbf{z} = 1|\mathbf{y}) = p_{\mathbf{z}|\mathbf{y}}, \quad P(\mathbf{z} = 0|\mathbf{y}) = 1 - p_{\mathbf{z}|\mathbf{y}}$   
 $U \sim \text{Uniform}(0, 1)$   
 $\mathbf{z} = 1_{\{U \leq p_{\mathbf{z}|\mathbf{y}}\}}$
- sample  $x$  analogously

# Latent variable models

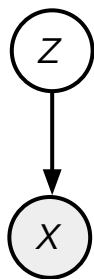
Models that explain observations with the help of unobserved, *latent* variables.



# Latent variable models

Models that explain observations with the help of unobserved, *latent* variables.

*Two challenges:*

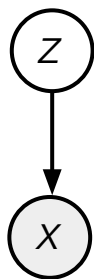


- 1) Infer the values of the latent variable
- 2) Estimate parameters of the model under uncertainty

# Latent variable models

Models that explain observations with the help of unobserved, *latent* variables.

*Two challenges:*



- 1) Infer the values of the latent variable
- 2) Estimate parameters of the model under uncertainty

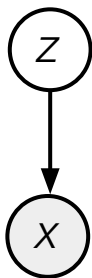
*General recipe:*

- 1) Use Bayesian inference to find a posterior for the latent variable

# Latent variable models

Models that explain observations with the help of unobserved, *latent* variables.

*Two challenges:*



- 1) Infer the values of the latent variable
- 2) Estimate parameters of the model under uncertainty

*General recipe:*

- 1) Use Bayesian inference to find a posterior for the latent variable
- 2) Maximize a likelihood (via ML) that
  - a) considers all possible values of the latent and
  - b) weights them by their probability (the posterior)

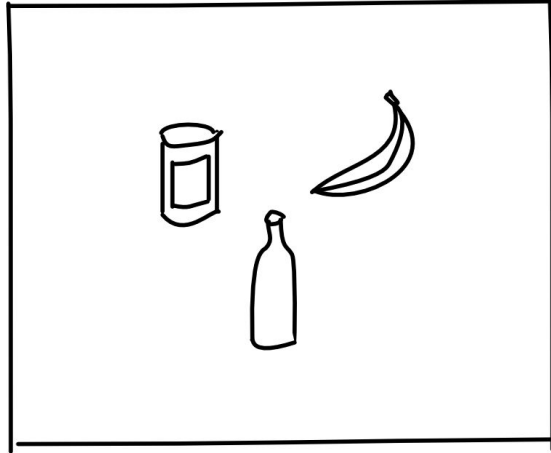


# Mixture Models and Expectation Maximization

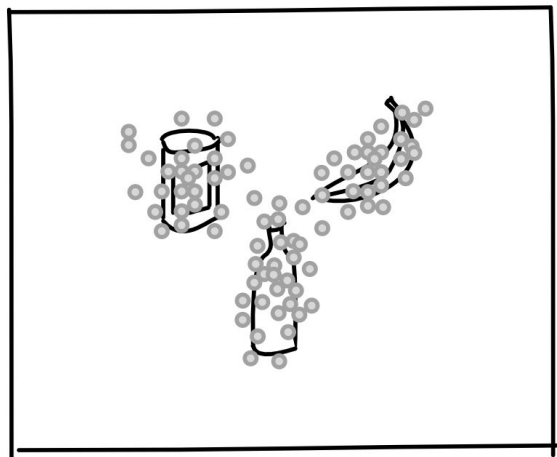
# Mixture models

## *Experiment:*

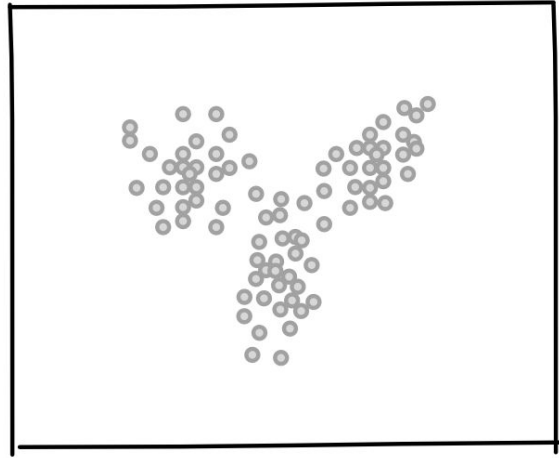
In each trial, subjects have to choose between three objects.



# Mixture models



# Mixture models

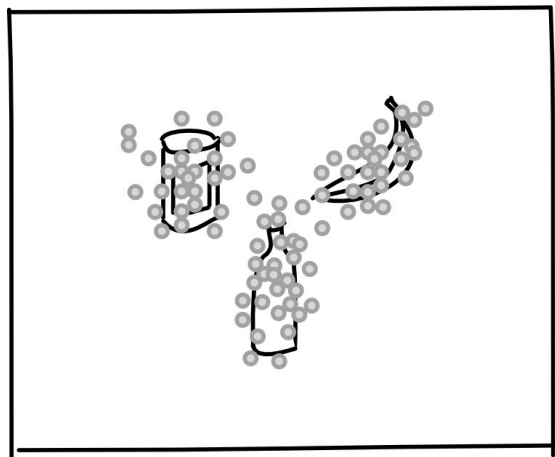


# Mixture models

*Hypothesis:*

For each trial, subjects

- (1) choose one of three objects, and
- (2) make a noisy saccade to the object's center

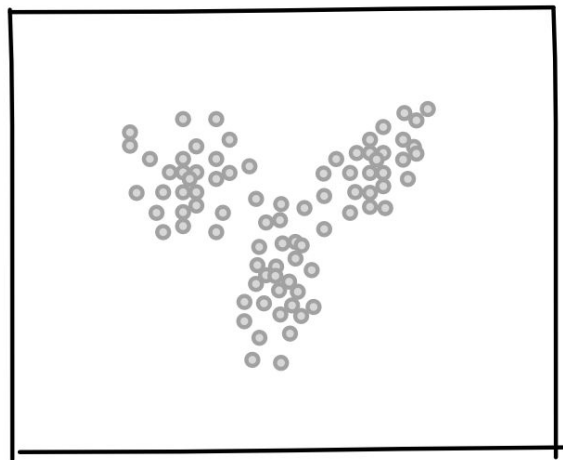


# Mixture models

*Hypothesis:*

For each trial, subjects

- (1) choose one of three objects, and
- (2) make a noisy saccade to the object's center



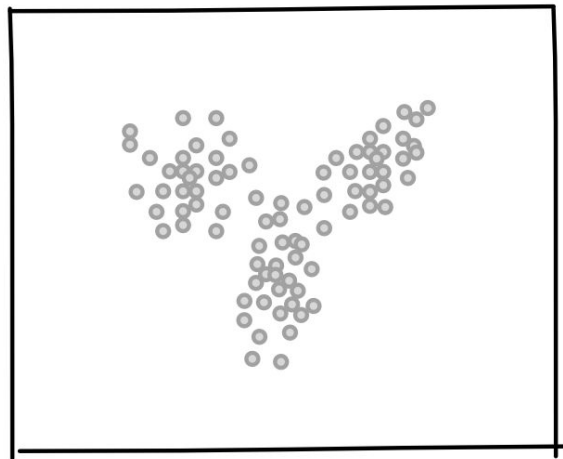
We have only observed  
saccades  $\mathbf{x} \in \mathbb{R}^{N \times D}$

# Mixture models

*Hypothesis:*

For each trial, subjects

- (1) choose one of three objects, and
- (2) make a noisy saccade to the object's center



We assume a categorical, *latent* variable  $\mathbf{z} \in \{0, 1\}^{N \times K}$

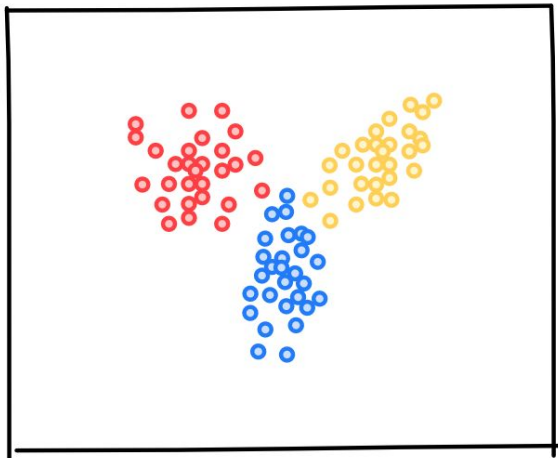
We have only observed saccades  $\mathbf{x} \in \mathbb{R}^{N \times D}$

# Mixture models

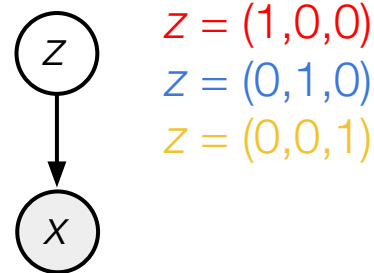
*Hypothesis:*

For each trial, subjects

- (1) choose one of three objects, and
- (2) make a noisy saccade to the object's center



We assume a categorical, *latent* variable  $\mathbf{z} \in \{0, 1\}^{N \times K}$



We have only observed saccades  $\mathbf{x} \in \mathbb{R}^{N \times D}$

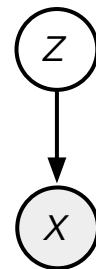
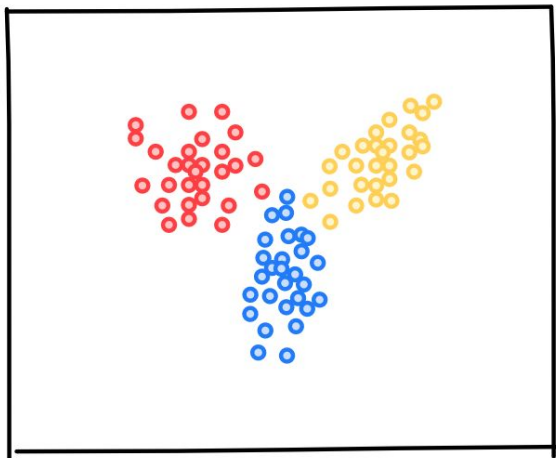


# Mixture models

*Hypothesis:*

For each trial, subjects

- (1) choose one of three objects, and
- (2) make a noisy saccade to the object's center



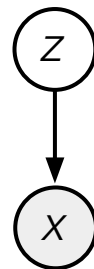
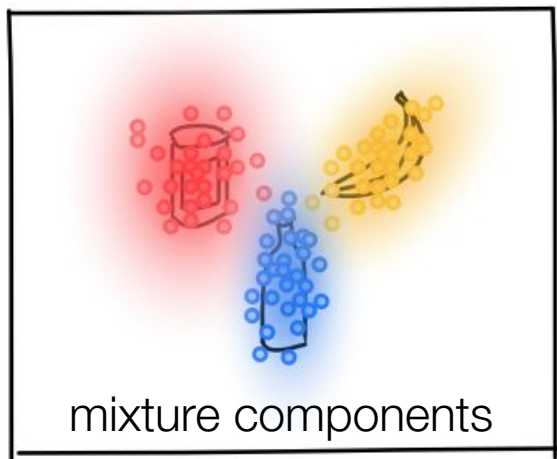
Joint distribution:

$$p(\mathbf{x}, \mathbf{z}) = p(\mathbf{x}|\mathbf{z})p(\mathbf{z})$$

# Mixture models

$$p(\mathbf{x}|z_k = 1) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

For each class  $k$ ,  
observations  $\mathbf{x}$  are  
distributed as a class-  
specific Gaussian.



Joint distribution:

$$p(\mathbf{x}, \mathbf{z}) = p(\mathbf{x}|\mathbf{z})p(\mathbf{z})$$

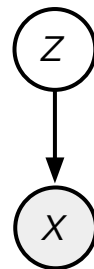
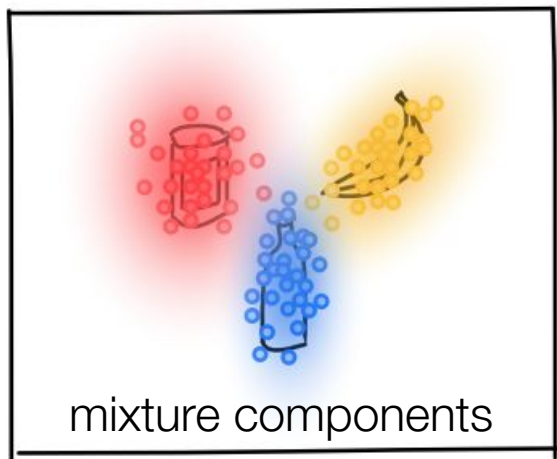
# Mixture models

$$p(\mathbf{x}|z_k = 1) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

For each class  $k$ ,  
observations  $\mathbf{x}$  are  
distributed as a class-  
specific Gaussian.

$$p(z_k = 1) = \pi_k$$

The probability of class  $k$   
(:= *mixing coefficient*) is its  
relative frequency.



Joint distribution:

$$p(\mathbf{x}, \mathbf{z}) = p(\mathbf{x}|\mathbf{z})p(\mathbf{z})$$

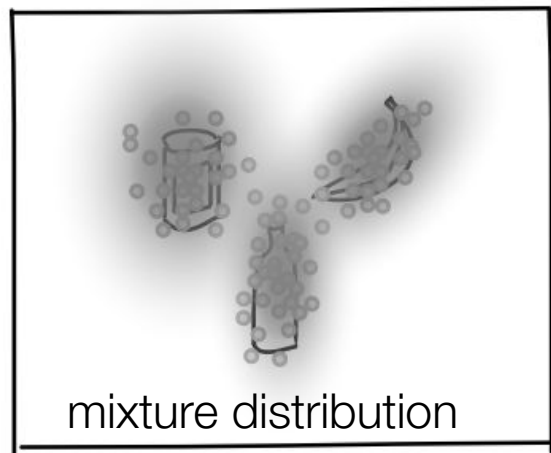
# Mixture models

$$p(\mathbf{x}|z_k = 1) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

For each class  $k$ ,  
observations  $\mathbf{x}$  are  
distributed as a class-  
specific Gaussian.

$$p(z_k = 1) = \pi_k$$

The probability of class  $k$   
(:= *mixing coefficient*) is its  
relative frequency.



Joint distribution:

$$p(\mathbf{x}, \mathbf{z}) = p(\mathbf{x}|\mathbf{z})p(\mathbf{z})$$



Marginal distribution:

$$p(\mathbf{x})$$

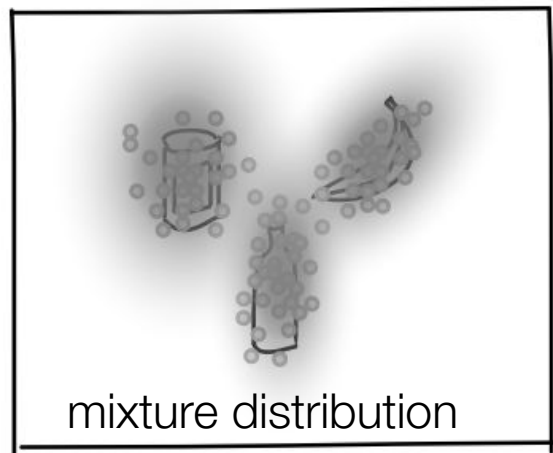
# Mixture models

$$p(\mathbf{x}|z_k = 1) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

For each class  $k$ ,  
observations  $\mathbf{x}$  are  
distributed as a class-  
specific Gaussian.

$$p(z_k = 1) = \pi_k$$

The probability of class  $k$   
(:= *mixing coefficient*) is its  
relative frequency.



Joint distribution:

$$p(\mathbf{x}, \mathbf{z}) = p(\mathbf{x}|\mathbf{z})p(\mathbf{z})$$



Marginal distribution:

$$p(\mathbf{x}) = \sum_{\mathbf{z}} p(\mathbf{x}, \mathbf{z})$$

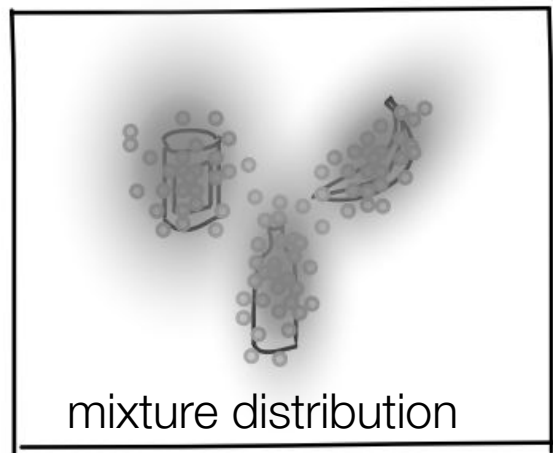
# Mixture models

$$p(\mathbf{x}|z_k = 1) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

For each class  $k$ ,  
observations  $\mathbf{x}$  are  
distributed as a class-  
specific Gaussian.

$$p(z_k = 1) = \pi_k$$

The probability of class  $k$   
( $:=$  *mixing coefficient*) is its  
relative frequency.



Joint distribution:

$$p(\mathbf{x}, \mathbf{z}) = p(\mathbf{x}|\mathbf{z})p(\mathbf{z})$$



Marginal distribution:

$$\begin{aligned} p(\mathbf{x}) &= \sum_{\mathbf{z}} p(\mathbf{x}, \mathbf{z}) \\ &= \sum_{\mathbf{z}} p(\mathbf{z})p(\mathbf{x}|\mathbf{z}) \end{aligned}$$

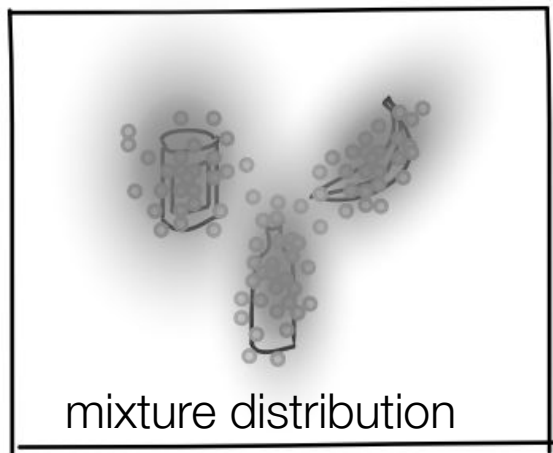
# Mixture models

$$p(\mathbf{x}|z_k = 1) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

For each class  $k$ , observations  $\mathbf{x}$  are distributed as a class-specific Gaussian.

$$p(z_k = 1) = \pi_k$$

The probability of class  $k$  ( $:=$  *mixing coefficient*) is its relative frequency.



Joint distribution:

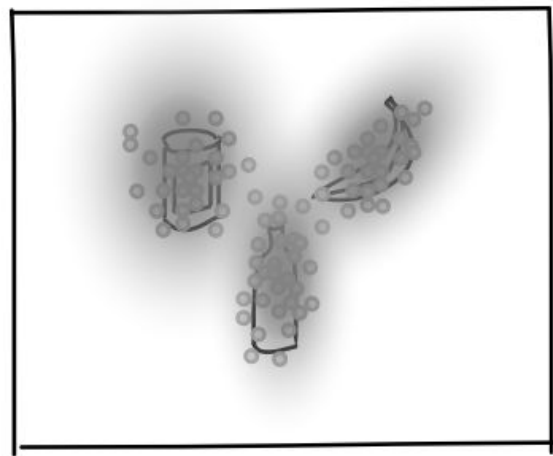
$$p(\mathbf{x}, \mathbf{z}) = p(\mathbf{x}|\mathbf{z})p(\mathbf{z})$$



Marginal distribution:

$$\begin{aligned} p(\mathbf{x}) &= \sum_{\mathbf{z}} p(\mathbf{x}, \mathbf{z}) \\ &= \sum_{\mathbf{z}} p(\mathbf{z})p(\mathbf{x}|\mathbf{z}) \\ &= \sum_{k=1}^K \pi_k \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \end{aligned}$$

# Mixture models

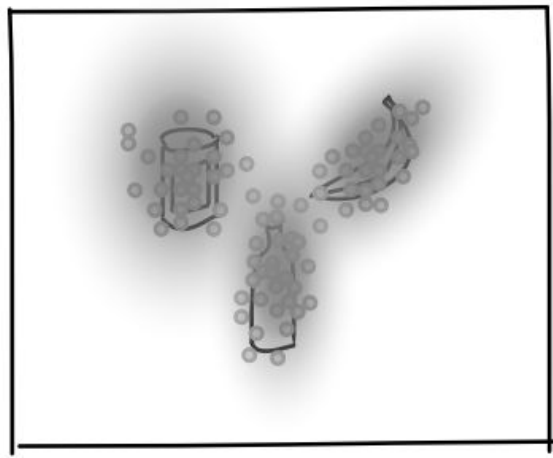


$$p(\mathbf{x}) = \sum_{k=1}^K \pi_k \mathcal{N}(\mathbf{x} | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$



# Mixture models

How do we estimate  
class-specific parameters?

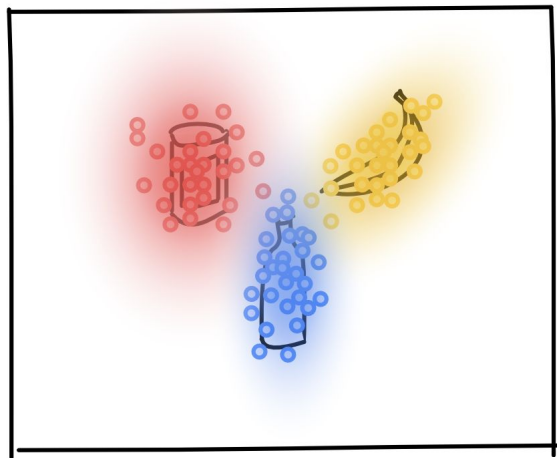


$$p(\mathbf{x}) = \sum_{k=1}^K \pi_k \mathcal{N}(\mathbf{x} | \mu_k, \Sigma_k)$$

# Mixture models

How do we estimate  
class-specific parameters?

Which data point belongs  
to which class?

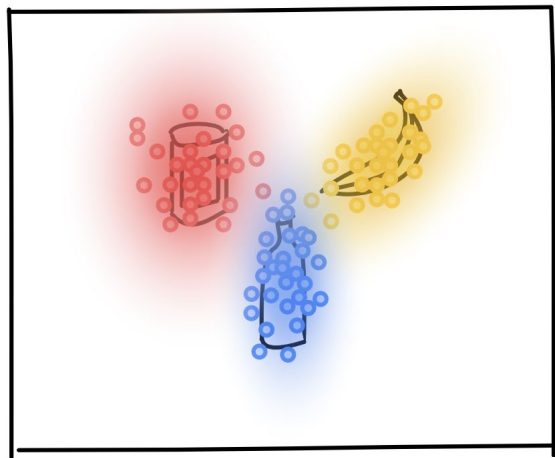


$$p(\mathbf{x}) = \sum_{k=1}^K \pi_k \mathcal{N}(\mathbf{x} | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

# Mixture models

How do we estimate  
class-specific parameters?

Which data point belongs  
to which class?



*Expectation maximization*  
(EM) is a general algorithm  
for parameter estimation in  
models of the form

$$p(\mathbf{x}, \mathbf{z} | \theta) = p(\mathbf{x} | \mathbf{z}, \theta) p(\mathbf{z} | \theta)$$

$$p(\mathbf{x}) = \sum_{k=1}^K \pi_k \mathcal{N}(\mathbf{x} | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

# Expectation maximization (EM)

EM tackles two problems:

**E-Step:** Determine which data point belongs to which class

**M-Step:** Fit class-specific parameters

# Expectation maximization (EM)

EM tackles two problems:

**E-Step:** Determine which data point belongs to which class

→ Inference of the **posterior** over the latent indicator variable **z**

$$p(\mathbf{z}|\mathbf{x}, \theta) = \frac{p(\mathbf{x}|\mathbf{z}, \theta)p(\mathbf{z}|\theta)}{p(\mathbf{x}|\theta)}$$

# Expectation maximization (EM)

EM tackles two problems:

**E-Step:** Determine which data point belongs to which class

→ Inference of the **posterior** over the latent indicator variable  $\mathbf{z}$

$$p(\mathbf{z}|\mathbf{x}, \theta) = \frac{p(\mathbf{x}|\mathbf{z}, \theta)p(\mathbf{z}|\theta)}{p(\mathbf{x}|\theta)} \longleftarrow \sum_{\mathbf{z}} p(\mathbf{x}, \mathbf{z}|\theta) = \sum_{\mathbf{z}} p(\mathbf{x}|\mathbf{z}, \theta)p(\mathbf{z}|\theta)$$

# Expectation maximization (EM)

EM tackles two problems:

**E-Step:** Determine which data point belongs to which class

→ Inference of the **posterior** over the latent indicator variable **z**

$$\begin{aligned} p(\mathbf{z}|\mathbf{x}, \theta) &= \frac{p(\mathbf{x}|\mathbf{z}, \theta)p(\mathbf{z}|\theta)}{p(\mathbf{x}|\theta)} \\ &= \frac{\pi_k \mathcal{N}(\mathbf{x}; \mu_k, \Sigma_k)}{\sum_K \pi_k \mathcal{N}(\mathbf{x}; \mu_k, \Sigma_k)} \end{aligned} \quad \text{for Gaussian mixture models}$$

# Expectation maximization (EM)

EM tackles two problems:

**E-Step:** Determine which data point belongs to which class

→ Inference of the **posterior** over the latent indicator variable  $\mathbf{z}$

$$\begin{aligned} p(\mathbf{z}|\mathbf{x}, \theta) &= \frac{p(\mathbf{x}|\mathbf{z}, \theta)p(\mathbf{z}|\theta)}{p(\mathbf{x}|\theta)} \\ &= \frac{\pi_k \mathcal{N}(\mathbf{x}; \mu_k, \Sigma_k)}{\sum_K \pi_k \mathcal{N}(\mathbf{x}; \mu_k, \Sigma_k)} \end{aligned}$$

Then: plug in current parameter estimates,  
evaluate PDF for each datapoint  
and each class, normalize

→ **responsibilities**  $\gamma_{ik} = p(z_{ik} = 1|x_i, \theta)$

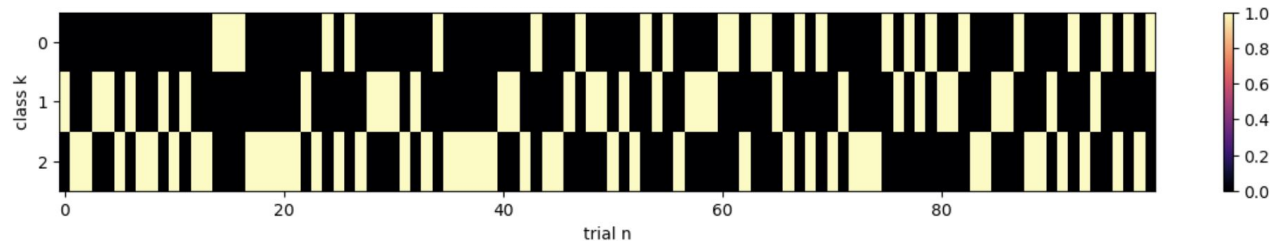


# Expectation maximization (EM)

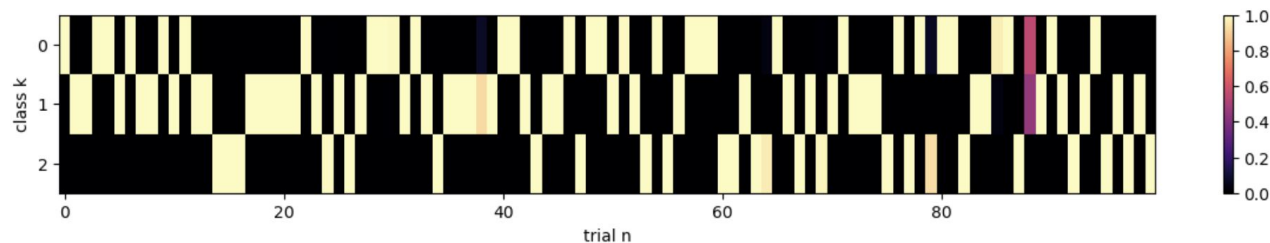
EM tackles two problems:

**E-Step:** Determine which data point belongs to which class

→ Inference of the **posterior** over the latent indicator variable  $\mathbf{z}$



$$z_{ik} = 1$$



$$\gamma_{ik} = p(z_{ik} = 1 | x_i, \theta)$$

# Expectation maximization (EM)

EM tackles two problems:

**E-Step:** Determine which data point belongs to which class

→ Inference of the **posterior**  $p(\mathbf{z}|\mathbf{x}, \theta)$  over the latent indicator variable  $\mathbf{z}$

**M-Step:** Fit class-specific parameters

→ Maximize the *complete-data log LL*  $\ln p(\mathbf{x}, \mathbf{z}|\theta)$

# Expectation maximization (EM)

EM tackles two problems:

**E-Step:** Determine which data point belongs to which class

→ Inference of the **posterior**  $p(\mathbf{z}|\mathbf{x}, \theta)$  over the latent indicator variable  $\mathbf{z}$

**M-Step:** Fit class-specific parameters

→ Maximize the *complete-data log LL*  $\ln p(\mathbf{x}, \mathbf{z}|\theta)$

→ Problem: We don't know the value of  $\mathbf{z}$

# Expectation maximization (EM)

EM tackles two problems:

**E-Step:** Determine which data point belongs to which class

→ Inference of the **posterior**  $p(\mathbf{z}|\mathbf{x}, \theta)$  over the latent indicator variable  $\mathbf{z}$

**M-Step:** Fit class-specific parameters

→ Maximize the *complete-data log LL*  $\ln p(\mathbf{x}, \mathbf{z}|\theta)$

→ Problem: We don't know the value of  $\mathbf{z}$

**Solution:** Optimize expected value under the posterior distribution of  $\mathbf{z}$

$$\mathbb{E}_{\mathbf{z}} [\ln p(\mathbf{x}, \mathbf{z}|\theta')] = \sum_{\mathbf{z}} p(\mathbf{z}|\mathbf{x}, \theta) \ln p(\mathbf{x}, \mathbf{z}|\theta')$$

# Expectation maximization (EM)

EM tackles two problems:

**E-Step:** Determine which data point belongs to which class

→ Inference of the **posterior**  $p(\mathbf{z}|\mathbf{x}, \theta)$  over the latent indicator variable  $\mathbf{z}$

**M-Step:** Fit class-specific parameters

→ Maximize the *complete-data log LL*  $\ln p(\mathbf{x}, \mathbf{z}|\theta)$

→ Problem: We don't know the value of  $\mathbf{z}$

**Solution:** Optimize expected value under the posterior distribution of  $\mathbf{z}$

$$\mathbb{E}_{\mathbf{z}} [\ln p(\mathbf{x}, \mathbf{z}|\theta')] = \sum_{\mathbf{z}} p(\mathbf{z}|\mathbf{x}, \theta) \ln p(\mathbf{x}, \mathbf{z}|\theta')$$

# Expectation maximization (EM)

EM tackles two problems:

**E-Step:** Determine which data point belongs to which class

→ Inference of the **posterior**  $p(\mathbf{z}|\mathbf{x}, \theta)$  over the latent indicator variable  $\mathbf{z}$

**M-Step:** Fit class-specific parameters

→ Find parameters that optimize expected complete-data log likelihood

$$\mathbb{E}_{\mathbf{z}} [\ln p(\mathbf{x}, \mathbf{z}|\theta')] = \sum_{\mathbf{z}} p(\mathbf{z}|\mathbf{x}, \theta) \ln p(\mathbf{x}, \mathbf{z}|\theta')$$

# Expectation maximization (EM)

EM tackles two problems:


**E-Step:** Determine which data point belongs to which class

→ Inference of the **posterior**  $p(\mathbf{z}|\mathbf{x}, \theta)$  over the latent indicator variable  $\mathbf{z}$

**M-Step:** Fit class-specific parameters

→ Find parameters that optimize expected complete-data log likelihood

$$\mathbb{E}_{\mathbf{z}} [\ln p(\mathbf{x}, \mathbf{z}|\theta')] = \sum_{\mathbf{z}} p(\mathbf{z}|\mathbf{x}, \theta) \ln p(\mathbf{x}, \mathbf{z}|\theta')$$


$$p(\mathbf{x}|\mathbf{z}, \theta')p(\mathbf{z}|\theta')$$

Gaussian pdf weighted by  
class-specific mixing coefficient

# Expectation maximization (EM)

EM tackles two problems:

**E-Step:** Determine which data point belongs to which class

→ Inference of the **posterior**  $p(\mathbf{z}|\mathbf{x}, \theta)$  over the latent indicator variable  $\mathbf{z}$

**M-Step:** Fit class-specific parameters

→ Find parameters that optimize expected complete-data log likelihood

$$\mathbb{E}_{\mathbf{z}} [\ln p(\mathbf{x}, \mathbf{z}|\theta')] = \sum_{\mathbf{z}} p(\mathbf{z}|\mathbf{x}, \theta) \ln p(\mathbf{x}, \mathbf{z}|\theta')$$

Posterior probabilities  
of each class for  
each datapoint

$p(\mathbf{x}|\mathbf{z}, \theta')p(\mathbf{z}|\theta')$   
Gaussian pdf weighted by  
class-specific mixing coefficient



# Expectation maximization (EM)

EM tackles two problems:

**E-Step:** Determine which data point belongs to which class

→ Inference of the **posterior**  $p(\mathbf{z}|\mathbf{x}, \theta)$  over the latent indicator variable  $\mathbf{z}$

**M-Step:** Fit class-specific parameters

→ Find parameters that optimize expected complete-data log likelihood

$$\mathbb{E}_{\mathbf{z}} [\ln p(\mathbf{x}, \mathbf{z}|\theta')] = \sum_{\mathbf{z}} p(\mathbf{z}|\mathbf{x}, \theta) \ln p(\mathbf{x}, \mathbf{z}|\theta')$$

Class-specific likelihoods  
are summed over all  
possible values of  $\mathbf{z}$

Posterior probabilities  
of each class for  
each datapoint

$p(\mathbf{x}|\mathbf{z}, \theta')p(\mathbf{z}|\theta')$   
Gaussian pdf weighted by  
class-specific mixing coefficient

# Expectation maximization (EM)

EM tackles two problems:

**E-Step:** Determine which data point belongs to which class

→ Inference of the **posterior**  $p(\mathbf{z}|\mathbf{x}, \theta)$  over the latent indicator variable  $\mathbf{z}$

**M-Step:** Fit class-specific parameters

→ Find parameters that optimize expected complete-data log likelihood

$$\mathbb{E}_{\mathbf{z}} [\ln p(\mathbf{x}, \mathbf{z}|\theta')] = \sum_{\mathbf{z}} p(\mathbf{z}|\mathbf{x}, \theta) \ln p(\mathbf{x}, \mathbf{z}|\theta')$$

**Iterate**

# Expectation maximization (EM)

EM tackles two problems *for any model with observations  $\mathbf{x}$  depending on latents  $\mathbf{z}$ :*

**E-Step:** Determine which data point belongs to which class

→ Inference of the **posterior**  $p(\mathbf{z}|\mathbf{x}, \theta)$  over the latent indicator variable  $\mathbf{z}$

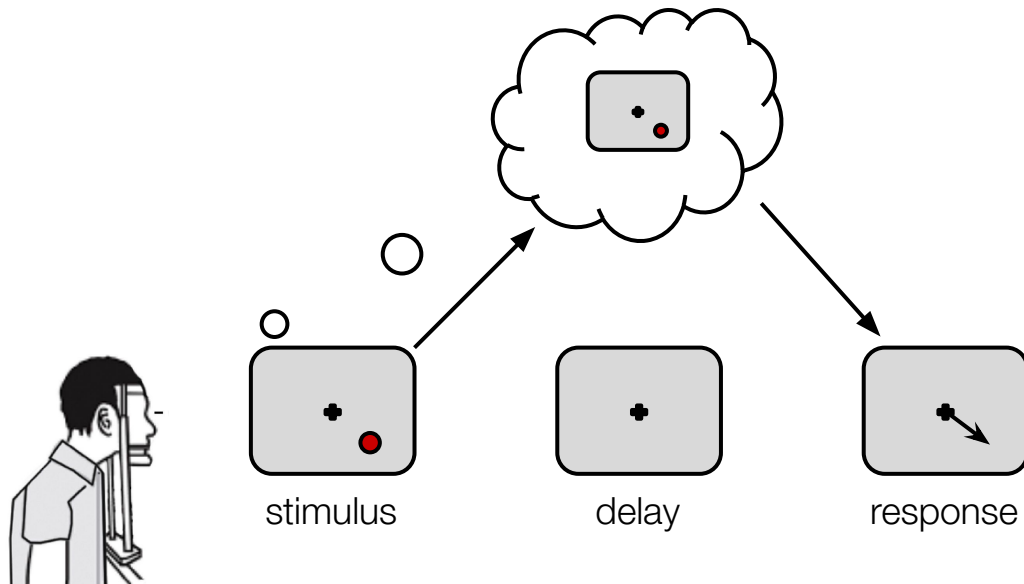
**M-Step:** Fit class-specific parameters

→ Find parameters that optimize expected complete-data log likelihood

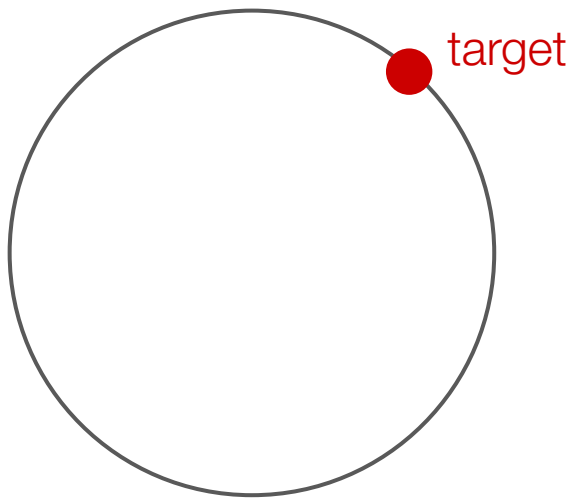
$$\mathbb{E}_{\mathbf{z}} [\ln p(\mathbf{x}, \mathbf{z}|\theta')] = \sum_{\mathbf{z}} p(\mathbf{z}|\mathbf{x}, \theta) \ln p(\mathbf{x}, \mathbf{z}|\theta')$$

**Iterate**

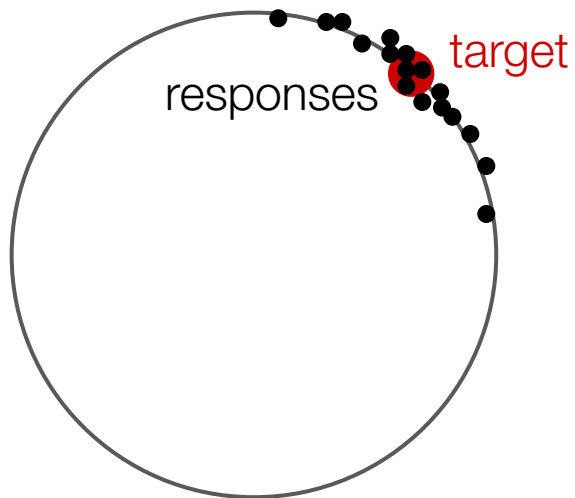
# Example of mixture models: classic WM task



## Example of mixture models: classic WM task

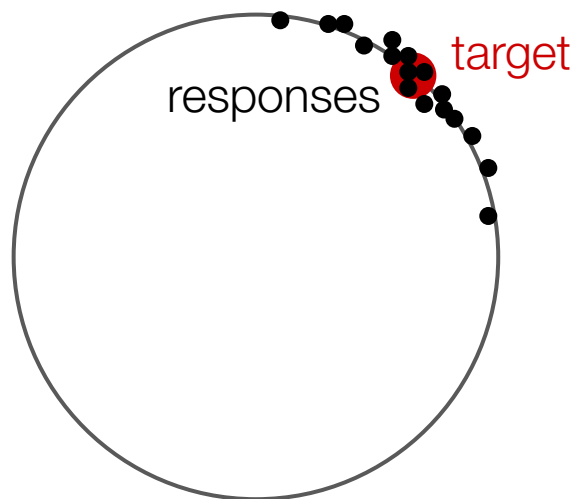


## Example of mixture models: classic WM task

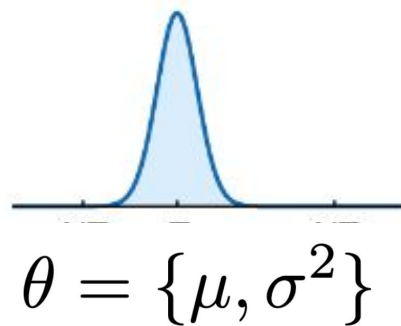


How precise is the working memory representation?

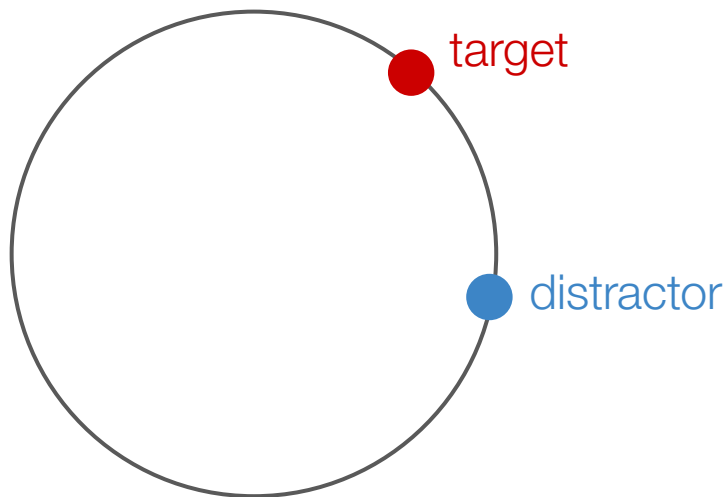
# Example of mixture models: classic WM task



How precise is the working memory representation?

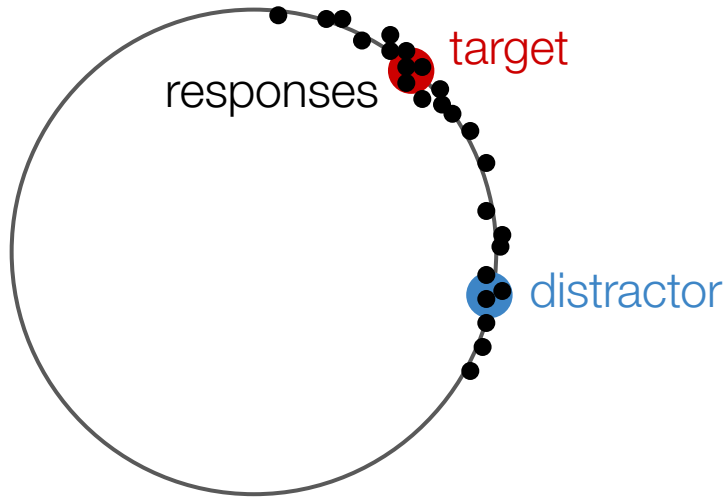


## Example of mixture models: classic WM task

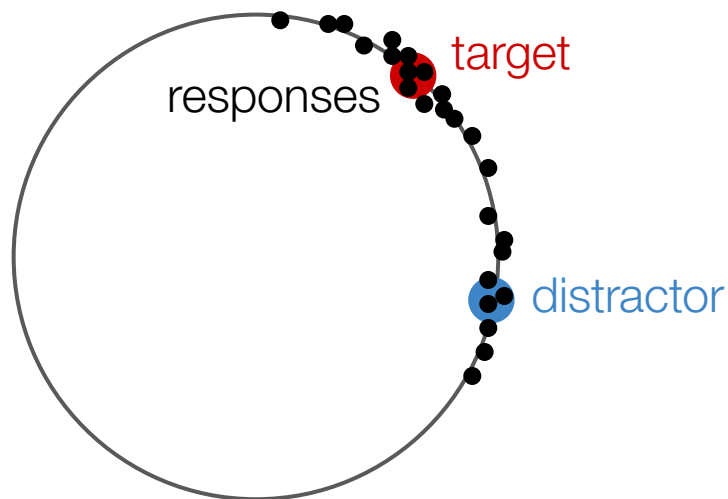




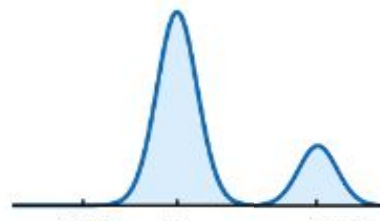
# Example of mixture models: classic WM task



# Example of mixture models: classic WM task



How precise is the working memory representation?



$$\theta = \{\mu_1, \mu_2, \sigma_1^2, \sigma_2^2\}$$

# Example of mixture models: classic WM task

More flexible models are possible! e.g. when target positions change from trial to trial

$$x^{(1)} = x_{\text{saccade}} - x_{\text{target}}$$

$$x^{(2)} = x_{\text{saccade}} - x_{\text{distractor}}$$

Mixtures of different distributions (e.g. Gaussian, uniform, Student t... → last exercise)

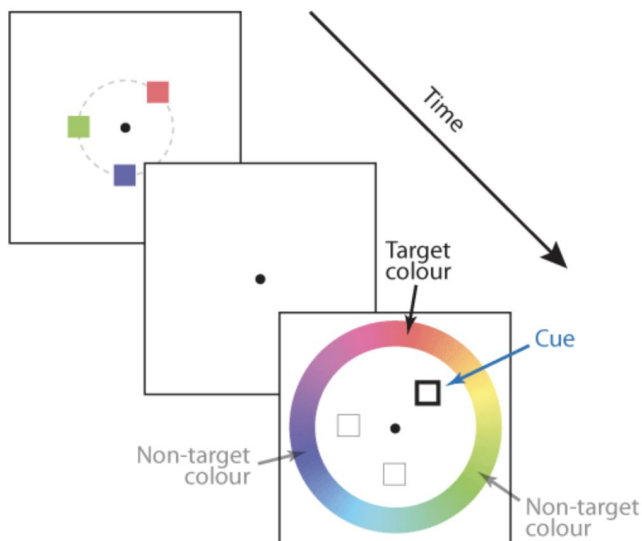
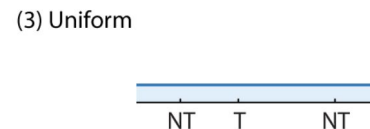
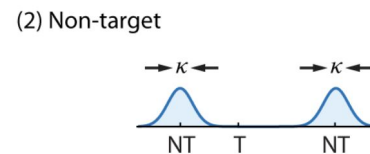
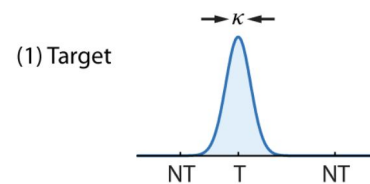


Figure 1 | The colour report task.



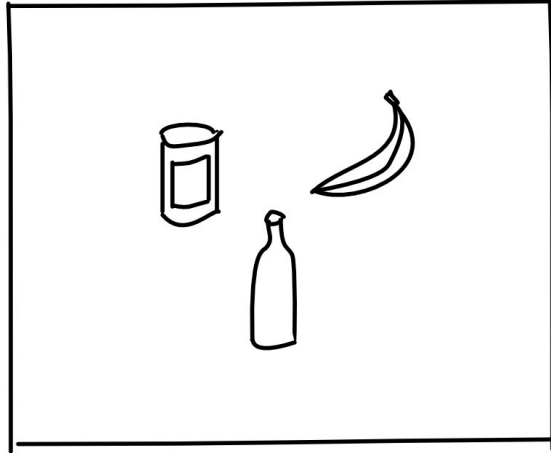
Bays, Catalao &  
Husain, *J Vis* (2009);  
Schneegans & Bays,  
*Cortex* (2016)

# Hidden Markov Models

# Hidden Markov models

*Experiment:*

In each trial, subjects have to choose between three objects.

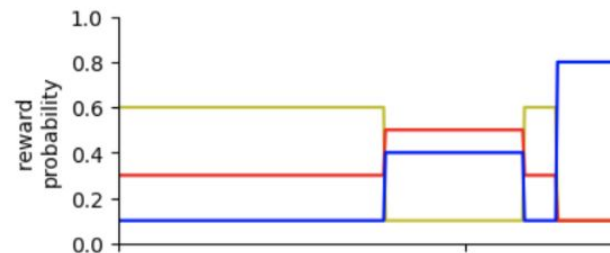
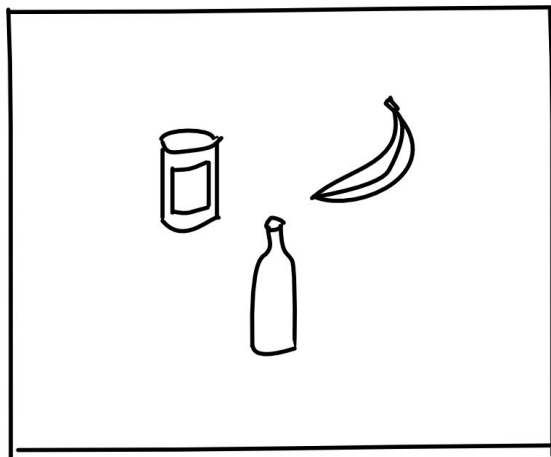


# Hidden Markov models

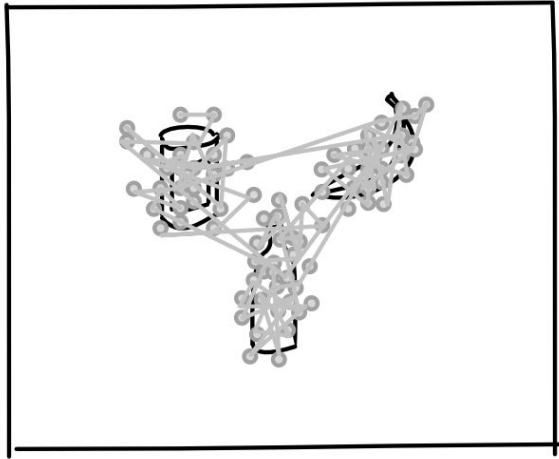
## *Experiment:*

In each trial, subjects have to choose between three objects.

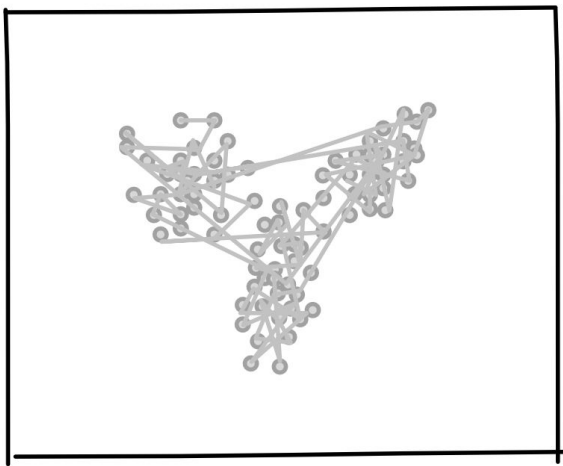
The relative value of objects fluctuates.



# Hidden Markov models



# Hidden Markov models



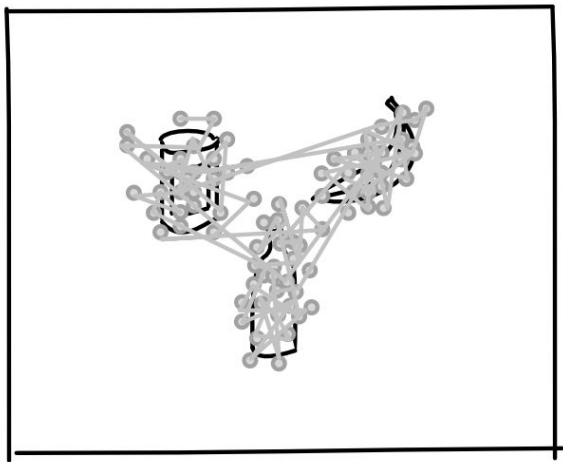


# Hidden Markov models

*Hypothesis:*

For each trial, subjects

- (1) choose one of three objects
- (2) which object they choose depends on their previous choice
- (3) make a noisy saccade to the object's center

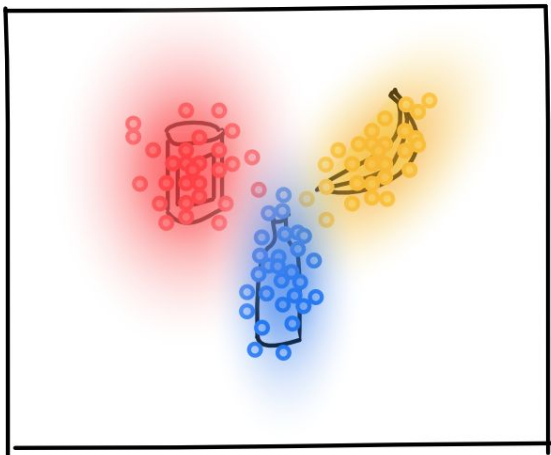


# Hidden Markov models

*Hypothesis:*

For each trial, subjects

- (1) choose one of three objects
- (2) which object they choose depends on their previous choice
- (3) make a noisy saccade to the object's center



latent variable  $\mathbf{z} \in \{0, 1\}^{N \times K}$



$z = (1, 0, 0)$

$z = (0, 1, 0)$

$z = (0, 0, 1)$



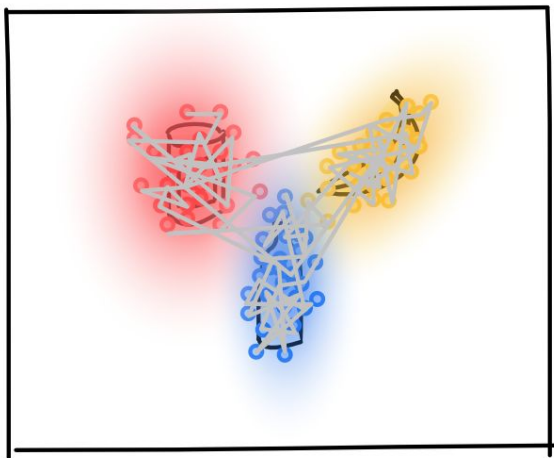
observations  $\mathbf{x} \in \mathbb{R}^{N \times D}$

# Hidden Markov models

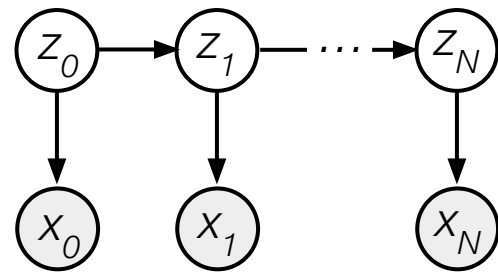
*Hypothesis:*

For each trial, subjects

- (1) choose one of three objects
- (2) which object they choose depends on their previous choice
- (3) make a noisy saccade to the object's center



latent variable  $\mathbf{z} \in \{0, 1\}^{N \times K}$

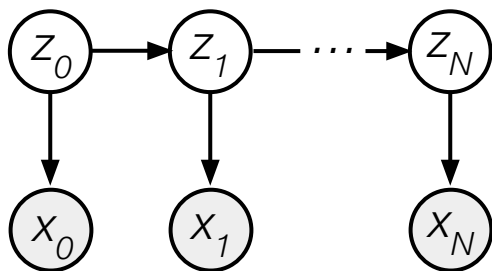


observations  $\mathbf{x} \in \mathbb{R}^{N \times D}$

→ temporal dependencies  
in the latent variable

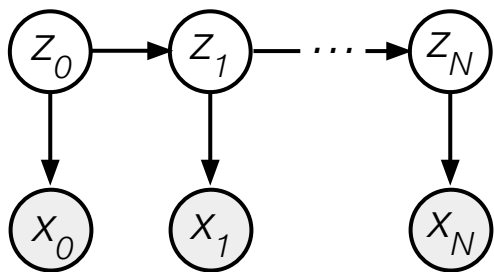
# Hidden Markov models

To capture sequential dependencies in choice behavior, we need an explicit model of transition structure of a latent variable



# Hidden Markov models

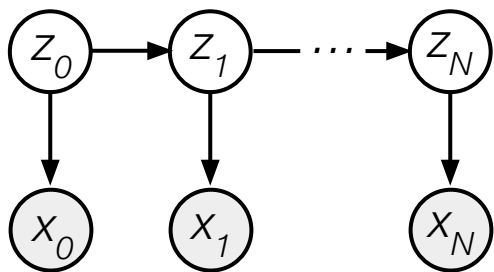
To capture sequential dependencies in choice behavior, we need an explicit model of transition structure of a latent variable



Markov property:  $p(\mathbf{z}_t | \mathbf{z}_0, \mathbf{z}_1, \dots, \mathbf{z}_{t-1}) = p(\mathbf{z}_t | \mathbf{z}_{t-1})$

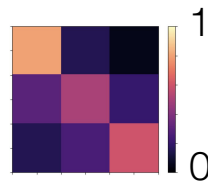
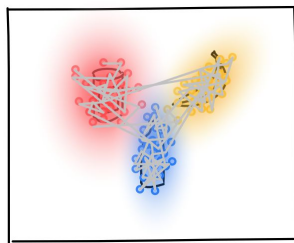
# Hidden Markov models

To capture sequential dependencies in choice behavior, we need an explicit model of transition structure of a latent variable



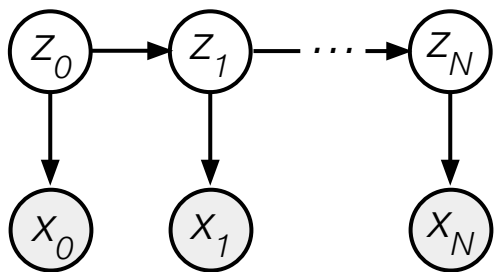
Markov property:  $p(\mathbf{z}_t | \mathbf{z}_0, \mathbf{z}_1, \dots, \mathbf{z}_{t-1}) = p(\mathbf{z}_t | \mathbf{z}_{t-1})$

→ We can summarize transition structure in the transition matrix  $\mathbf{A}$ , with  $A_{jk} \equiv p(z_{nk} = 1 | z_{n-1,j} = 1)$



# Hidden Markov models

To capture sequential dependencies in choice behavior, we need an explicit model of transition structure of a latent variable

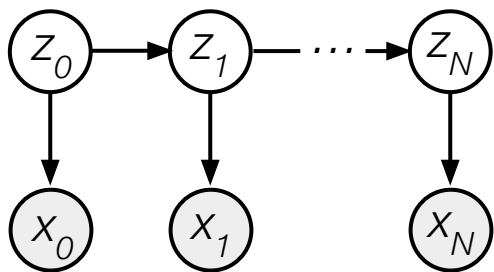


Joint distribution:

$$p(\mathbf{x}, \mathbf{z} | \theta) = p(\mathbf{z}_0) \prod_{n=1}^N p(\mathbf{z}_n | \mathbf{z}_{n-1}) \prod_{n=0}^N p(\mathbf{x}_n | \mathbf{z}_n, \theta)$$

# Hidden Markov models

To capture sequential dependencies in choice behavior, we need an explicit model of transition structure of a latent variable



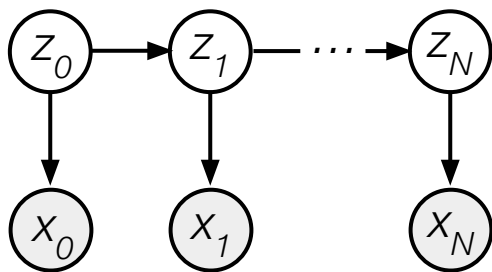
Joint distribution:

$$p(\mathbf{x}, \mathbf{z} | \theta) = \underbrace{p(z_0)}_{\text{initial state}} \underbrace{\prod_{n=1}^N p(z_n | z_{n-1})}_{\text{transition matrix}} \underbrace{\prod_{n=0}^N p(x_n | z_n, \theta)}_{\text{likelihoods}}$$



# Hidden Markov models

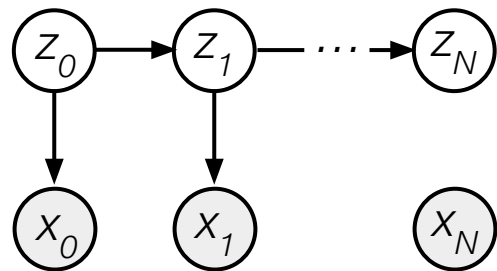
To capture sequential dependencies in choice behavior, we need an explicit model of transition structure of a latent variable



Joint distribution:

$$p(\mathbf{x}, \mathbf{z} | \theta) = \underbrace{p(z_0)}_{\substack{\text{initial} \\ \text{state} \\ \pi}} \underbrace{\prod_{n=1}^N p(z_n | z_{n-1})}_{\substack{\text{transition} \\ \text{matrix} \\ A}} \underbrace{\prod_{n=0}^N p(\mathbf{x}_n | z_n, \theta)}_{\substack{\text{likelihoods} \\ \mathcal{N}(\mathbf{x}_n; \mu_k, \Sigma_k)}}$$

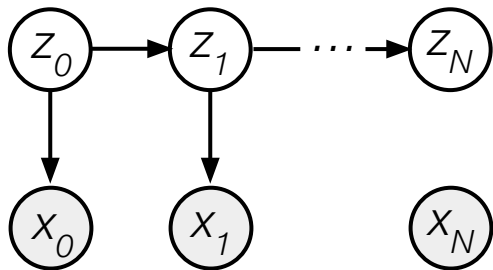
# EM for Hidden Markov Models



Joint distribution:

$$p(\mathbf{x}, \mathbf{z}|\theta) = p(z_0) \prod_{n=1}^N p(z_n|z_{n-1}) \prod_{n=0}^N p(x_n|z_n, \theta)$$

# EM for Hidden Markov Models

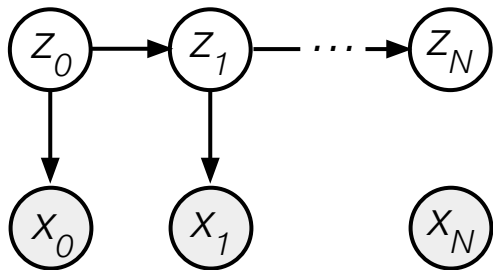


Joint distribution:

$$p(\mathbf{x}, \mathbf{z} | \theta) = p(z_0) \prod_{n=1}^N p(z_n | z_{n-1}) \prod_{n=0}^N p(x_n | z_n, \theta)$$

We want to infer the latent states  $\mathbf{z}$ , and estimate parameters  $\theta = \{\boldsymbol{\pi}, \mathbf{A}, \boldsymbol{\mu}, \boldsymbol{\Sigma}\}$

# EM for Hidden Markov Models



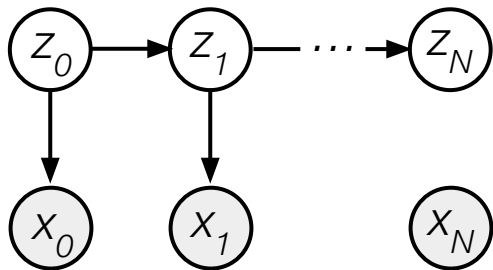
Joint distribution:

$$p(\mathbf{x}, \mathbf{z}|\theta) = p(\mathbf{z}_0) \prod_{n=1}^N p(\mathbf{z}_n|\mathbf{z}_{n-1}) \prod_{n=0}^N p(\mathbf{x}_n|\mathbf{z}_n, \theta)$$

We want to infer the latent states  $\mathbf{z}$ , and estimate parameters  $\theta = \{\boldsymbol{\pi}, \mathbf{A}, \boldsymbol{\mu}, \boldsymbol{\Sigma}\}$

**E-Step:** infer posteriors, and calculate initial and state transition probabilities  $\boldsymbol{\pi}, \mathbf{A}$

# EM for Hidden Markov Models



Joint distribution:

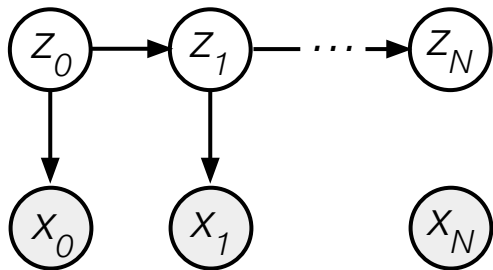
$$p(\mathbf{x}, \mathbf{z} | \theta) = p(\mathbf{z}_0) \prod_{n=1}^N p(\mathbf{z}_n | \mathbf{z}_{n-1}) \prod_{n=0}^N p(\mathbf{x}_n | \mathbf{z}_n, \theta)$$

We want to infer the latent states  $\mathbf{z}$ , and estimate parameters  $\theta = \{\boldsymbol{\pi}, \mathbf{A}, \boldsymbol{\mu}, \boldsymbol{\Sigma}\}$

**E-Step:** infer posteriors, and calculate initial and state transition probabilities  $\boldsymbol{\pi}, \mathbf{A}$

**M-Step:** update parameters  $\boldsymbol{\mu}, \boldsymbol{\Sigma}$  based on complete-data likelihood

# EM for Hidden Markov Models



Joint distribution:

$$p(\mathbf{x}, \mathbf{z}|\theta) = p(z_0) \prod_{n=1}^N p(z_n|z_{n-1}) \prod_{n=0}^N p(x_n|z_n, \theta)$$

We want to infer the latent states  $\mathbf{z}$ , and estimate parameters  $\theta = \{\pi, \mathbf{A}, \mu, \Sigma\}$

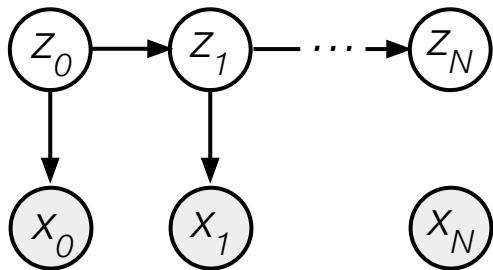
**E-Step:** infer posteriors, and calculate initial and state transition probabilities  $\pi, \mathbf{A}$

**M-Step:** update parameters  $\mu, \Sigma$  based on complete-data likelihood

→ easy: optimize expected complete data log LL  $\mathbb{E}_{\mathbf{z}} [\ln p(\mathbf{x}, \mathbf{z}|\theta')]$

$$= \sum_{\mathbf{z}} p(\mathbf{z}|\mathbf{x}, \theta) \ln p(\mathbf{x}, \mathbf{z}|\theta')$$

# EM for Hidden Markov Models

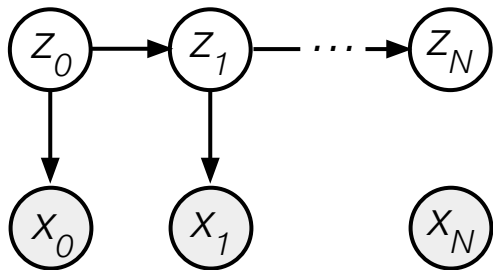


Joint distribution:

$$p(\mathbf{x}, \mathbf{z} | \theta) = p(z_0) \prod_{n=1}^N p(z_n | z_{n-1}) \prod_{n=0}^N p(x_n | z_n, \theta)$$

**E-Step:** infer posteriors

# EM for Hidden Markov Models



Joint distribution:

$$p(\mathbf{x}, \mathbf{z} | \theta) = p(\mathbf{z}_0) \prod_{n=1}^N p(\mathbf{z}_n | \mathbf{z}_{n-1}) \prod_{n=0}^N p(\mathbf{x}_n | \mathbf{z}_n, \theta)$$

**E-Step:** infer posteriors

→ In HMMs, there are two posteriors over  $\mathbf{z}$ :

$$p(\mathbf{z}_n | \mathbf{x}, \theta)$$

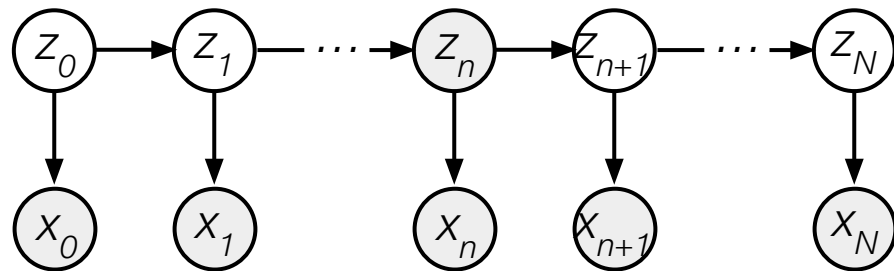
probability of each latent state value, given observations

$$p(\mathbf{z}_n, \mathbf{z}_{n-1} | \mathbf{x}, \theta)$$

probability of observing a pair of subsequent states, – “ –



# EM for Hidden Markov Models

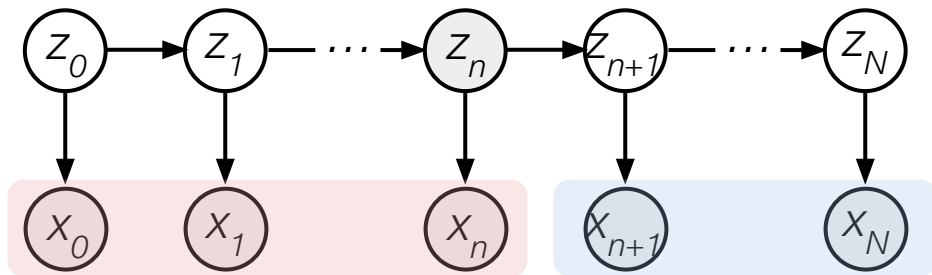


**E-Step:** infer posteriors

→ again, we start from Bayes theorem 
$$p(\mathbf{z}_n | \mathbf{x}_{0:N}, \theta) = \frac{p(\mathbf{x}_{0:N} | \mathbf{z}_n, \theta) p(\mathbf{z}_n)}{p(\mathbf{x}_{0:N})}$$

(and equivalent for  $p(\mathbf{z}_n, \mathbf{z}_{n-1} | \mathbf{x}, \theta)$ )

# EM for Hidden Markov Models



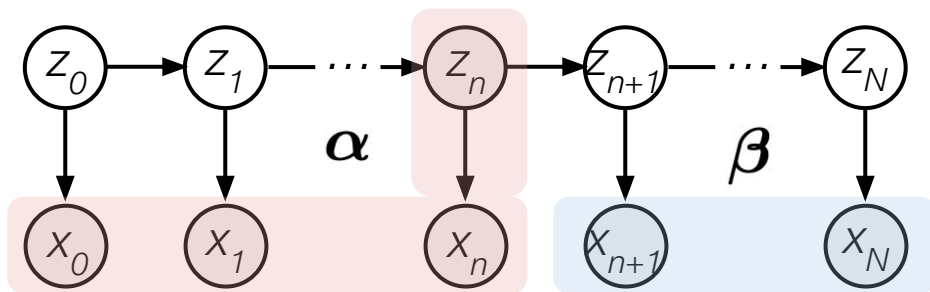
For a given state  $\mathbf{z}_n$  for sample  $n$ , we can split the likelihood in two terms:

$$p(\mathbf{x}_{0:n} | \mathbf{z}_n, \theta), \quad p(\mathbf{x}_{n+1:N} | \mathbf{z}_n, \theta)$$

**E-Step:** infer posteriors

→ again, we start from Bayes theorem 
$$p(\mathbf{z}_n | \mathbf{x}_{0:N}, \theta) = \frac{p(\mathbf{x}_{0:N} | \mathbf{z}_n, \theta) p(\mathbf{z}_n)}{p(\mathbf{x}_{0:N})}$$

# EM for Hidden Markov Models



We include  $p(z_n)$ :

$$\underbrace{p(\mathbf{x}_{0:n}, \mathbf{z}_n | \theta)}_{\alpha}, \underbrace{p(\mathbf{x}_{n+1:N} | \mathbf{z}_n, \theta)}_{\beta}$$

**E-Step:** infer posteriors

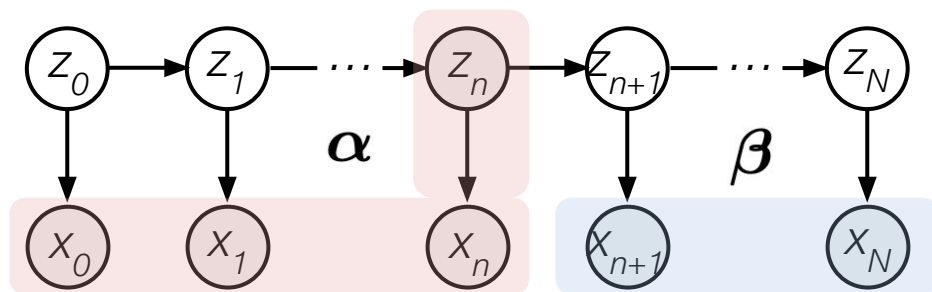
→ again, we start from Bayes theorem

$$p(\mathbf{z}_n | \mathbf{x}_{0:N}, \theta) = \frac{p(\mathbf{x}_{0:N} | \mathbf{z}_n, \theta) p(\mathbf{z}_n)}{p(\mathbf{x}_{0:N})}$$

→ Then the posterior becomes

$$p(\mathbf{z}_n | \mathbf{x}_{0:N}, \theta) = \frac{\alpha(\mathbf{z}_n) \beta(\mathbf{z}_n)}{p(\mathbf{x}_{0:N})}$$

# EM for Hidden Markov Models

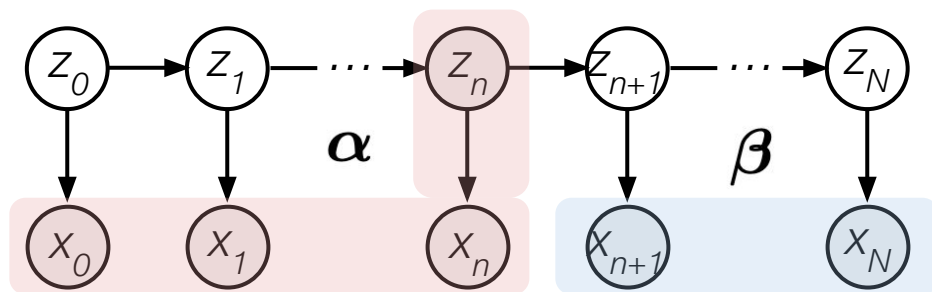


**E-Step:** infer posteriors

→ **Good news:**

1. There is an efficient algorithm for calculating  $\alpha$  and  $\beta$  (the Baum-Welch / forward-backward algorithm)

# EM for Hidden Markov Models

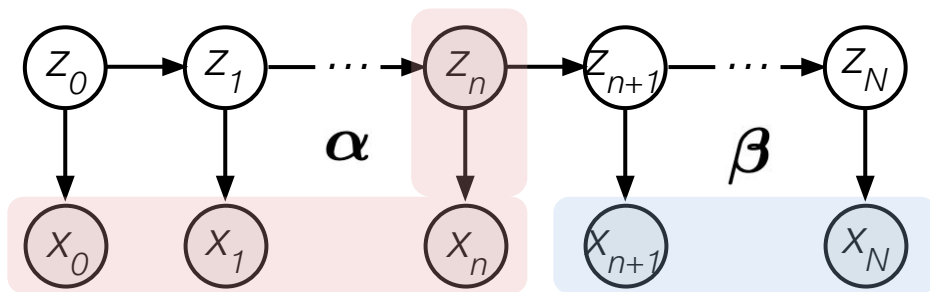


**E-Step:** infer posteriors

→ **Good news:**

1. There is an efficient algorithm for calculating  $\alpha$  and  $\beta$  (the Baum-Welch / forward-backward algorithm)
2. Both posteriors (  $p(\mathbf{z}_n|\mathbf{x}, \theta)$  and  $p(\mathbf{z}_n, \mathbf{z}_{n-1}|\mathbf{x}, \theta)$  ) can be calculated from  $\alpha$  and  $\beta$

# EM for Hidden Markov Models



$$p(\mathbf{x}, \mathbf{z} | \theta) = p(z_0) \prod_{n=1}^N p(z_n | z_{n-1}) \prod_{n=0}^N p(x_n | z_n, \theta)$$

## E-Step:

Baum-Welch algorithm to infer posteriors  $p(z_n | \mathbf{x}, \theta)$  and  $p(z_n, z_{n-1} | \mathbf{x}, \theta)$   
→ this also gives us probabilities  $\pi, \mathbf{A}$

## M-Step:

update parameters  $\mu, \Sigma$  based on complete-data log likelihood  $\mathbb{E}_{\mathbf{z}} [\ln p(\mathbf{x}, \mathbf{z} | \theta')]$

# Remarks on emission models

Gaussian emission models:  $p(\mathbf{x}|z_k = 1, \theta_k) = \mathcal{N}(\mathbf{x}|\mu_k, \Sigma_k)$

Other (including more complex) emission models are possible:

- Student-t, etc... (continuous observations  $\mathbf{x}$ )
- Categorical emissions, Poisson emissions etc (discrete observations  $\mathbf{x}$ )

# Remarks on emission models

Gaussian emission models:  $p(\mathbf{x}|z_k = 1, \theta_k) = \mathcal{N}(\mathbf{x}|\mu_k, \Sigma_k)$

Other (including more complex) emission models are possible:

- Student-t, etc... (continuous observations  $\mathbf{x}$ )
- Categorical emissions, Poisson emissions etc (discrete observations  $\mathbf{x}$ )
- Linear models of input  $\mathbf{u}$ :  $p(\mathbf{x}|z_k = 1, \theta_k) = \mathcal{N}(\mathbf{x}|\mathbf{W}_k\mathbf{u} + c_k, \Sigma_k)$
- Autoregressive models:  $p(\mathbf{x}_n|z_k = 1, \theta_k) = \mathcal{N}(\mathbf{x}_n|\mathbf{A}_k\mathbf{x}_{n-1} + d_k, \Sigma_k)$



# Remarks on emission models

Gaussian emission models:  $p(\mathbf{x}|z_k = 1, \theta_k) = \mathcal{N}(\mathbf{x}|\mu_k, \Sigma_k)$

Other (including more complex) emission models are possible:

- Student-t, etc... (continuous observations  $\mathbf{x}$ )
- Categorical emissions, Poisson emissions etc (discrete observations  $\mathbf{x}$ )
- Linear models of input  $\mathbf{u}$ :  $p(\mathbf{x}|z_k = 1, \theta_k) = \mathcal{N}(\mathbf{x}|\mathbf{W}_k\mathbf{u} + c_k, \Sigma_k)$
- Autoregressive models:  $p(\mathbf{x}_n|z_k = 1, \theta_k) = \mathcal{N}(\mathbf{x}_n|\mathbf{A}_k\mathbf{x}_{n-1} + d_k, \Sigma_k)$
- Combine them, you get switching drift diffusion models
-

# Remarks on emission models

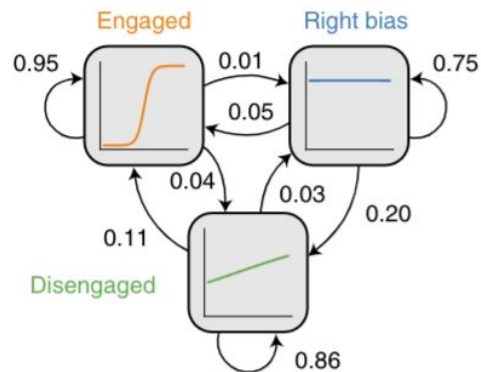
Gaussian emission models:  $p(\mathbf{x}|z_k = 1, \theta_k) = \mathcal{N}(\mathbf{x}|\mu_k, \Sigma_k)$

Other (including more complex) emission models are possible:

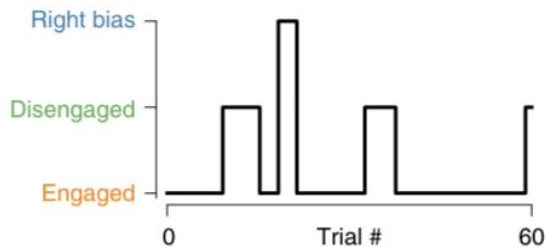
- Student-t, etc... (continuous observations  $\mathbf{x}$ )
- Categorical emissions, Poisson emissions etc (discrete observations  $\mathbf{x}$ )
- Linear models of input  $\mathbf{u}$ :  $p(\mathbf{x}|z_k = 1, \theta_k) = \mathcal{N}(\mathbf{x}|\mathbf{W}_k\mathbf{u} + c_k, \Sigma_k)$
- Autoregressive models:  $p(\mathbf{x}_n|z_k = 1, \theta_k) = \mathcal{N}(\mathbf{x}_n|\mathbf{A}_k\mathbf{x}_{n-1} + d_k, \Sigma_k)$
- Combine them, you get switching drift diffusion models
- Switching factor analysis (if you include an extra set of continuous latents that depend on the state)

# Examples of HMMs in neuroscience

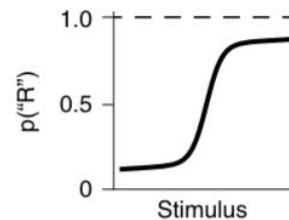
**d** Three-state GLM-HMM



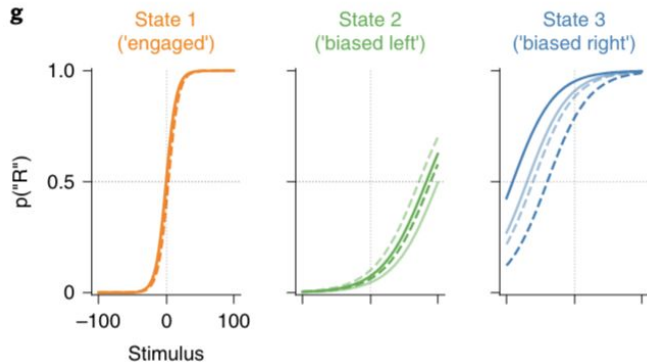
**e**



**f**

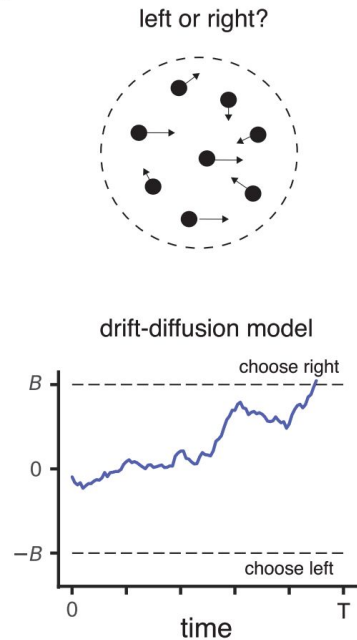


**g**

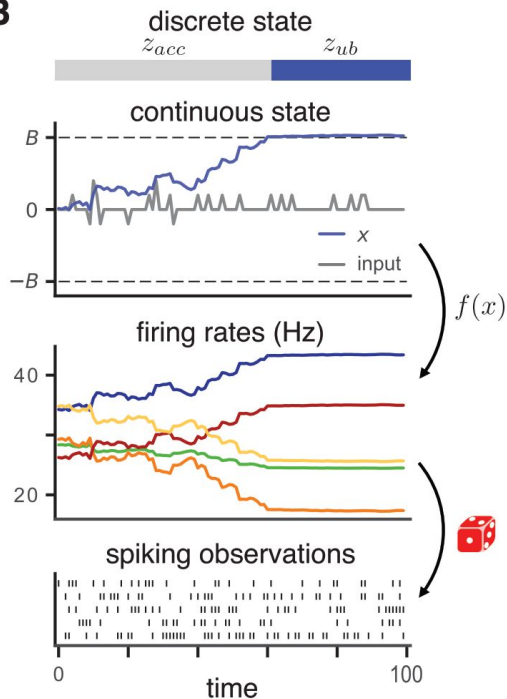


# Examples of HMMs in neuroscience

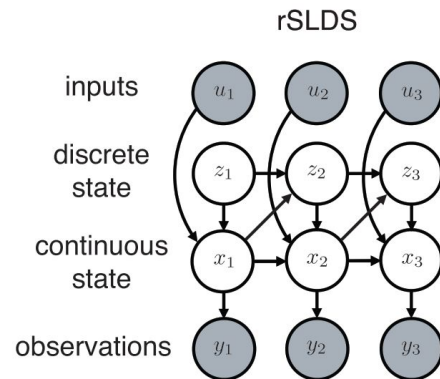
**A**



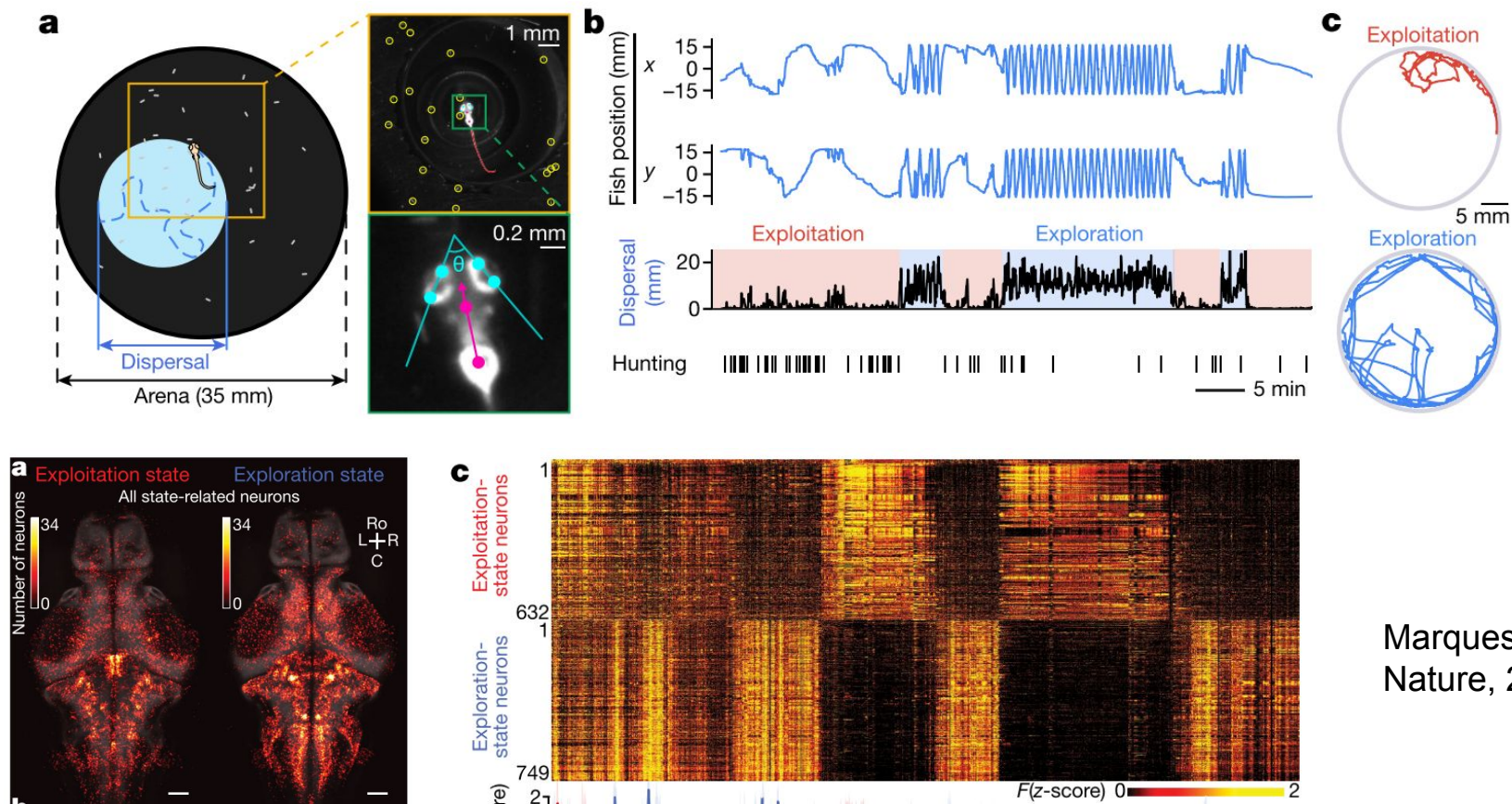
**B**



**C**



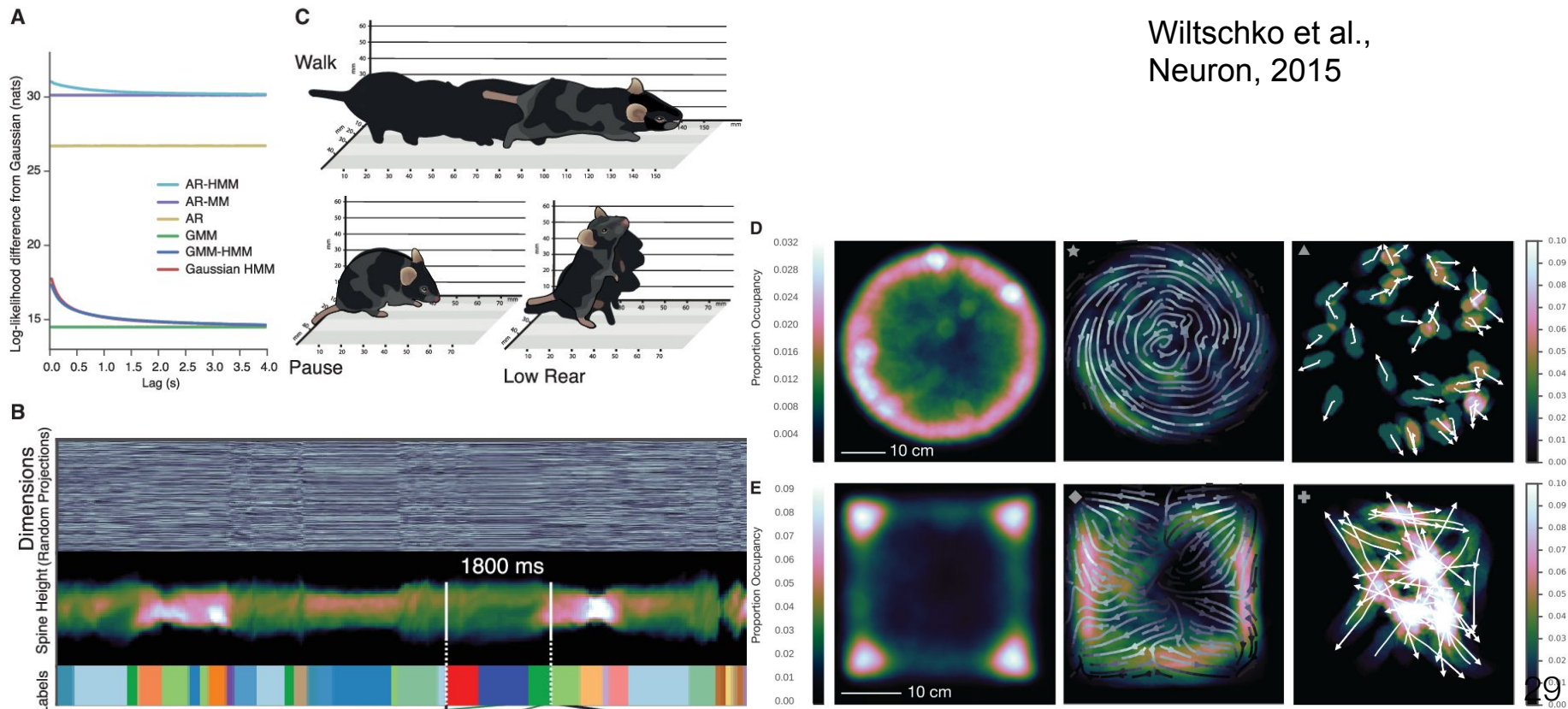
# Examples of HMMs in neuroscience



Marques et al.,  
Nature, 2019

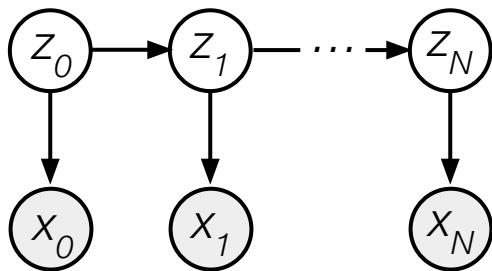
# Examples of HMMs in neuroscience

Wiltchko et al.,  
Neuron, 2015



# Kalman filter

The generative model is the same as for the HMM

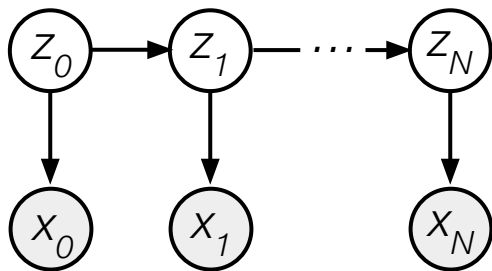


Joint distribution:

$$p(\mathbf{x}, \mathbf{z} | \theta) = p(\mathbf{z}_0 | \theta) \prod_{n=1}^N p(\mathbf{z}_n | \mathbf{z}_{n-1}, \theta) p(\mathbf{x}_n | \mathbf{z}_n, \theta)$$

# Kalman filter

The generative model is the same as for the HMM



Joint distribution:

$$p(\mathbf{x}, \mathbf{z} | \theta) = p(\mathbf{z}_0 | \theta) \prod_{n=1}^N p(\mathbf{z}_n | \mathbf{z}_{n-1}, \theta) p(\mathbf{x}_n | \mathbf{z}_n, \theta)$$

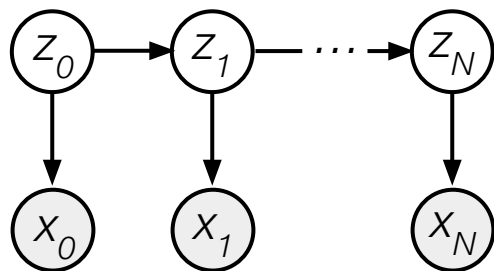
But now the latent  $\mathbf{z}$  is continuous-valued, with transition dynamics determined by

$$p(\mathbf{z}_n | \mathbf{z}_{n-1}, \theta) = \mathcal{N}(\mathbf{z}_n; \mathbf{A}\mathbf{z}_{n-1}, \mathbf{Q})$$



# Kalman filter

The generative model is the same as for the HMM



Joint distribution:

$$p(\mathbf{x}, \mathbf{z} | \theta) = p(\mathbf{z}_0 | \theta) \prod_{n=1}^N p(\mathbf{z}_n | \mathbf{z}_{n-1}, \theta) p(\mathbf{x}_n | \mathbf{z}_n, \theta)$$

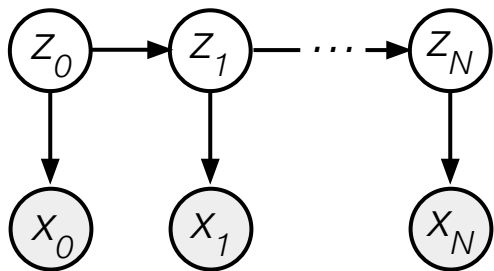
But now the latent  $\mathbf{z}$  is continuous-valued, with transition dynamics determined by

$$p(\mathbf{z}_n | \mathbf{z}_{n-1}, \theta) = \mathcal{N}(\mathbf{z}_n; \mathbf{A}\mathbf{z}_{n-1}, \mathbf{Q})$$

Observations depend linearly on the state  $\mathbf{z}$

$$p(\mathbf{x}_n | \mathbf{z}_n, \theta) = \mathcal{N}(\mathbf{x}_n; \mathbf{C}\mathbf{z}_n, \mathbf{R})$$

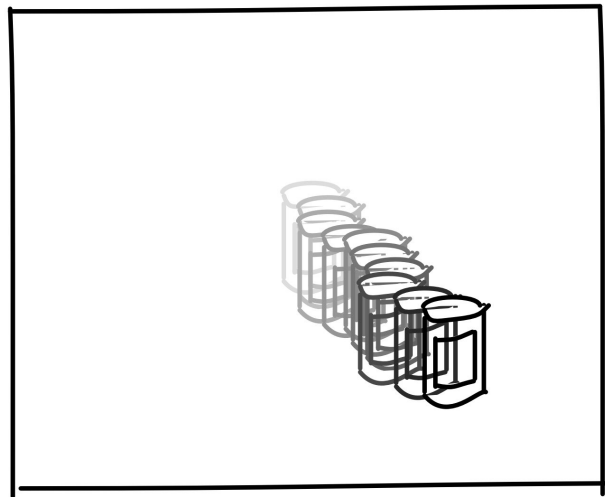
# Kalman filter



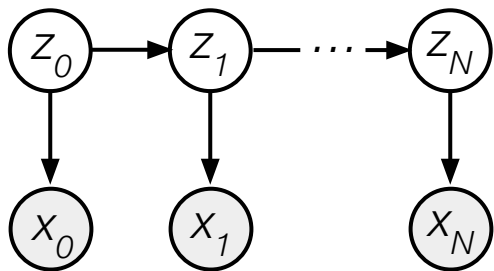
$$p(\mathbf{z}_n | \mathbf{z}_{n-1}, \theta) = \mathcal{N}(\mathbf{z}_n; \mathbf{A}\mathbf{z}_{n-1}, \mathbf{Q})$$

$$p(\mathbf{x}_n | \mathbf{z}_n, \theta) = \mathcal{N}(\mathbf{x}_n; \mathbf{C}\mathbf{z}_n, \mathbf{R})$$

Dynamics of the latent  $\mathbf{z}$



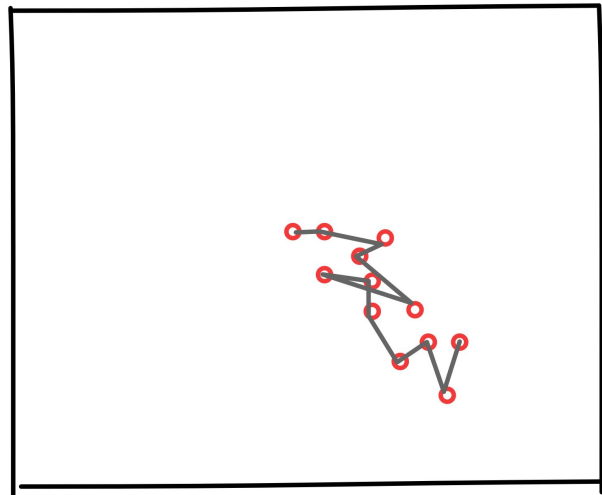
# Kalman filter



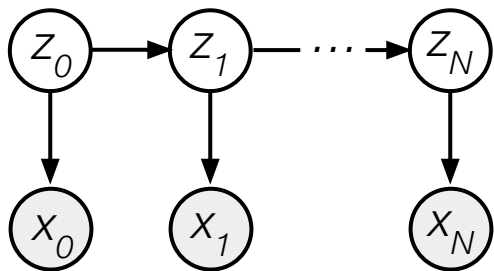
$$p(\mathbf{z}_n | \mathbf{z}_{n-1}, \theta) = \mathcal{N}(\mathbf{z}_n; \mathbf{A}\mathbf{z}_{n-1}, \mathbf{Q})$$

$$p(\mathbf{x}_n | \mathbf{z}_n, \theta) = \mathcal{N}(\mathbf{x}_n; \mathbf{C}\mathbf{z}_n, \mathbf{R})$$

Observations  $\mathbf{x}$



# Kalman filter

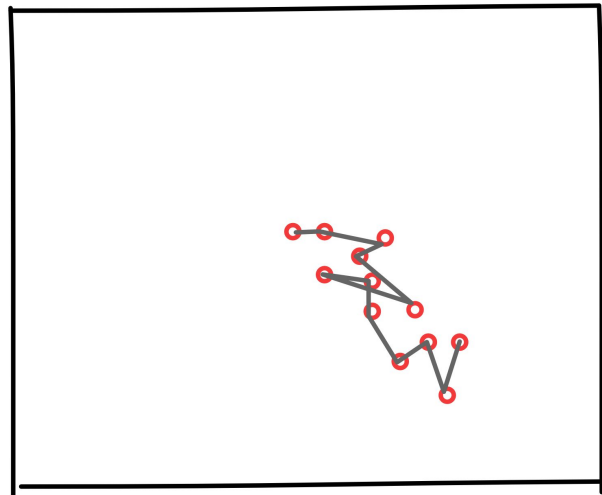


$$p(\mathbf{z}_n | \mathbf{z}_{n-1}, \theta) = \mathcal{N}(\mathbf{z}_n; \mathbf{A}\mathbf{z}_{n-1}, \mathbf{Q})$$

$$p(\mathbf{x}_n | \mathbf{z}_n, \theta) = \mathcal{N}(\mathbf{x}_n; \mathbf{C}\mathbf{z}_n, \mathbf{R})$$

→ Inference: Posterior for state  $\mathbf{z}$  and parameters  $\mathbf{A}, \mathbf{C}, \mathbf{Q}, \mathbf{R}$

Observations  $\mathbf{x}$



# Kalman filter

**Inference:** Posterior for state  $\mathbf{z}$  and parameters in  $p(\mathbf{z}_n | \mathbf{z}_{n-1}, \theta) = \mathcal{N}(\mathbf{z}_n; \mathbf{A}\mathbf{z}_{n-1}, \mathbf{Q})$   
 $p(\mathbf{x}_n | \mathbf{z}_n, \theta) = \mathcal{N}(\mathbf{x}_n; \mathbf{C}\mathbf{z}_n, \mathbf{R})$

## E-Step:

Forward-backward algorithm to obtain smoothing posterior  $p(\mathbf{z}_n | \mathbf{x}_{0:N}, \theta)$   
and state transition posterior  $p(\mathbf{z}_n | \mathbf{z}_{n-1}, \mathbf{x}_{1:N}, \theta)$

## M-Step:

update parameters  $\mathbf{A}, \mathbf{Q}, \mathbf{C}, \mathbf{R}$  based on complete-data log likelihood  $\mathbb{E}_{\mathbf{z}} [\ln p(\mathbf{x}, \mathbf{z} | \theta')]$

## Further reading & materials

- Textbook on probabilistic statistics and machine learning:  
Christopher Bishop's "Pattern Recognition and Machine Learning" (2006)
- Library for HMMs with all types of emission models:  
Scott Linderman's SSM library <https://github.com/lindermanlab/ssm>