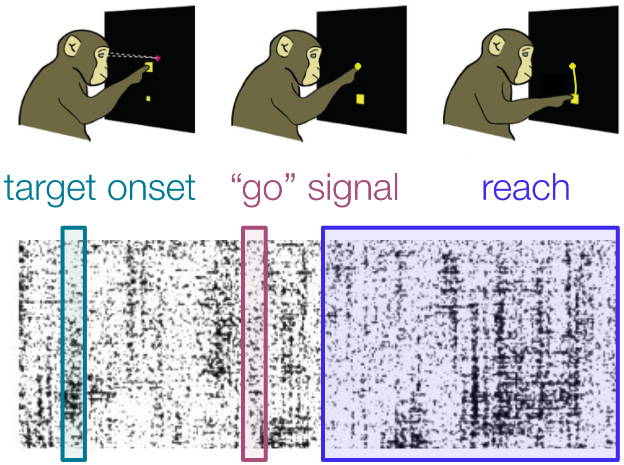
Latent variable models

Expectation maximization, mixture models, hidden Markov models

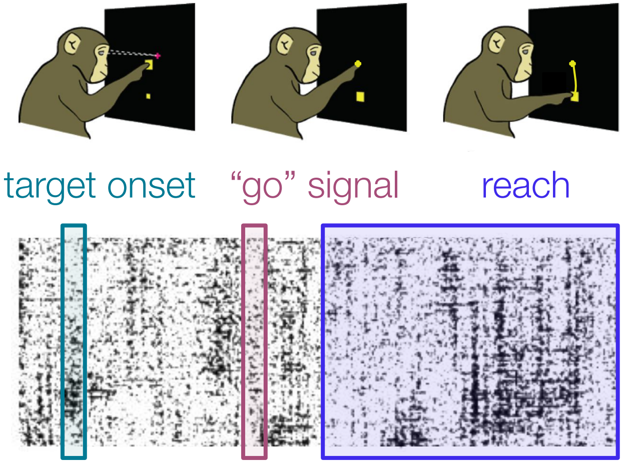
BAMB! ‘25
Heike Stein

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

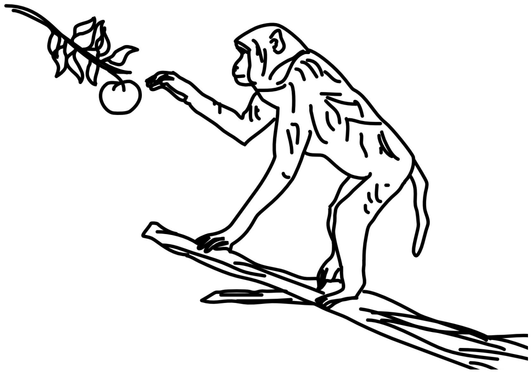


Behavior in the lab

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

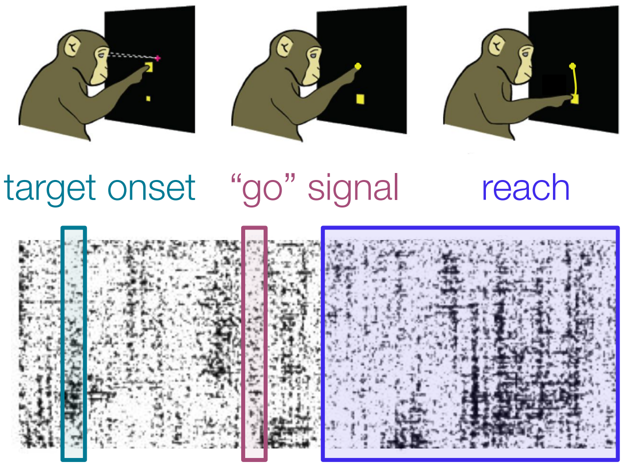


Behavior in the lab

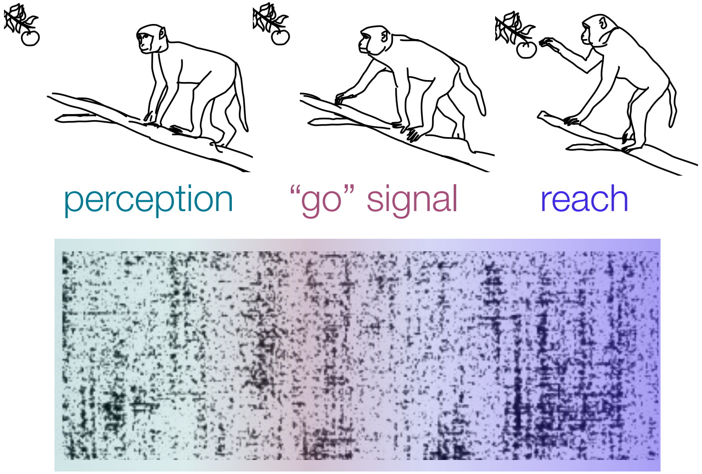


Natural behavior

Behavior in the lab vs. behavior "in the wild"

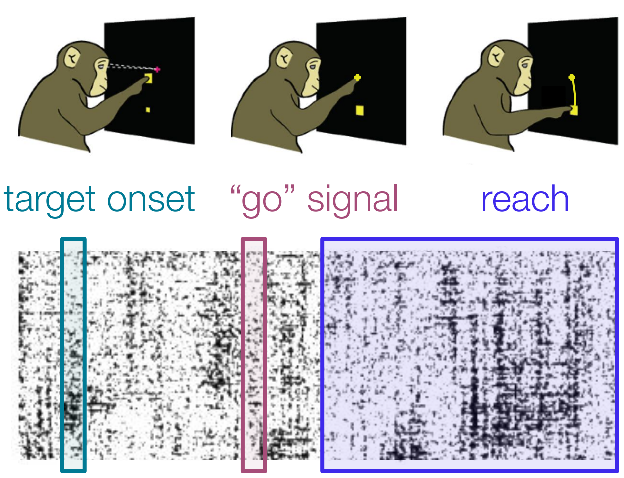


Behavior in the lab

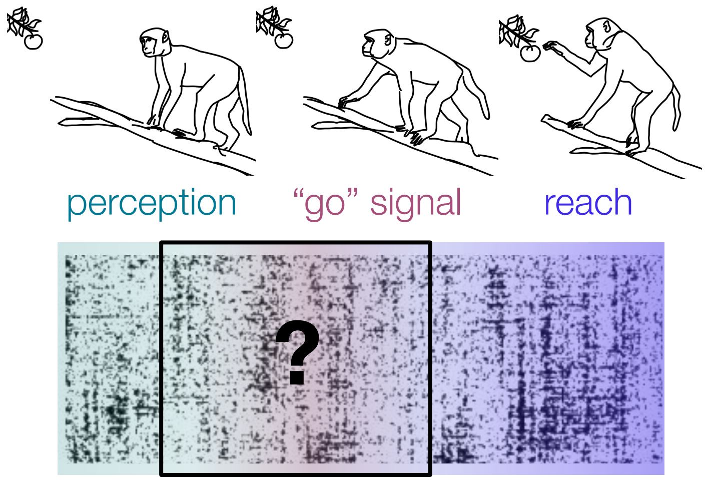


Natural behavior

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

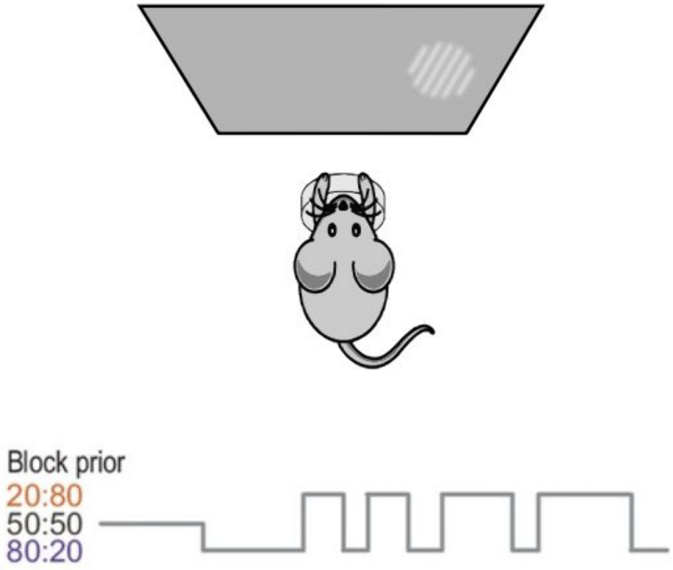


Behavior in the lab

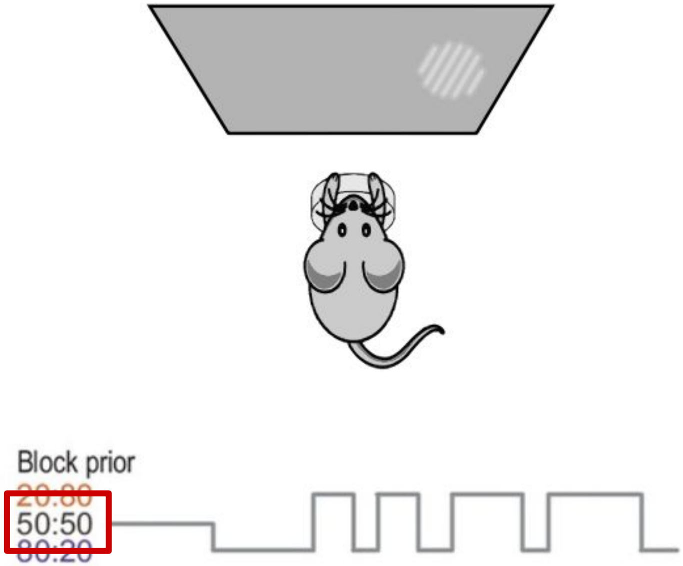


Natural behavior

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



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



Only 50:50 trials!

nature neuroscience

ARTICLES

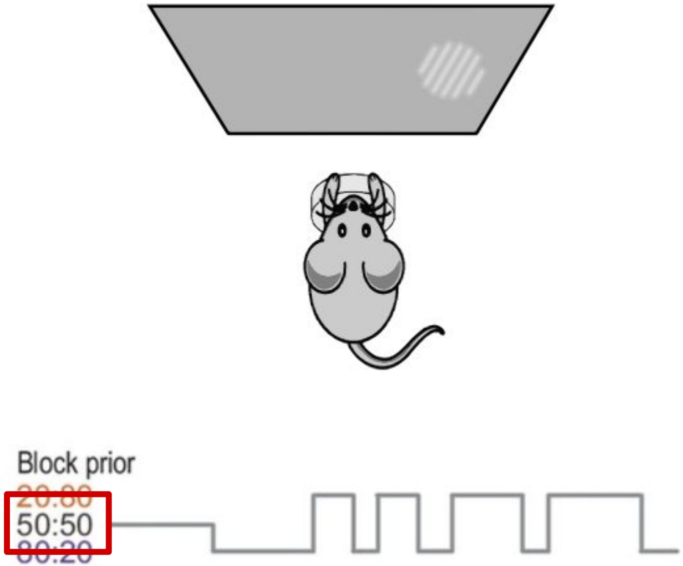
https://doi.org/10.1038/s41593-021-01007-z

Check for updates

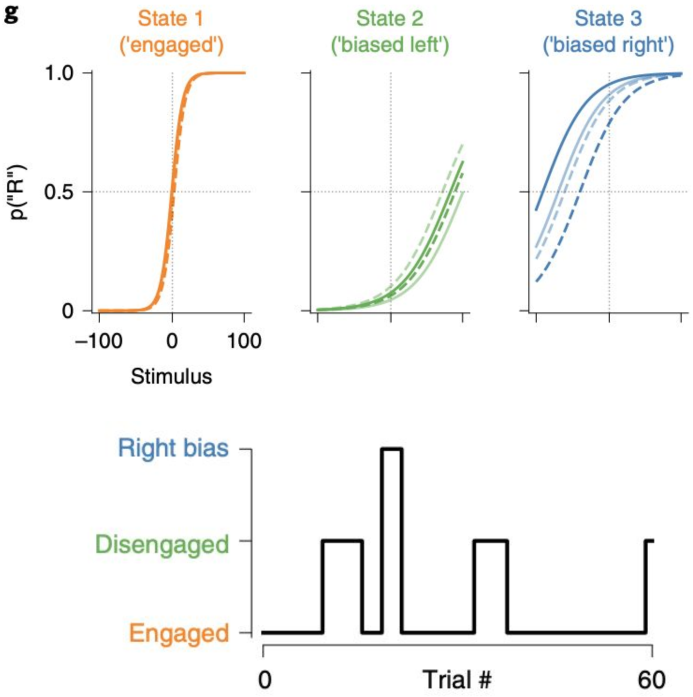
Mice alternate between discrete strategies during perceptual decision-making

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

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



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 unobserved “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 unobserved “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 unobserved “helper” variables (*latent variables*) to find structure underlying different behavioral patterns

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

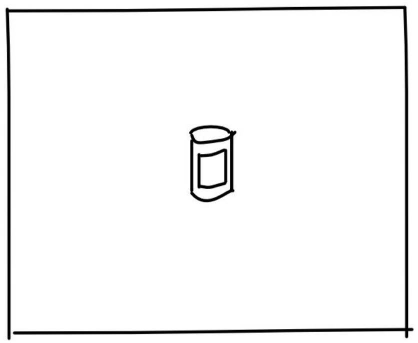
→ This will work if the data is explained by a mix of simpler models

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

General intro: Probabilistic models

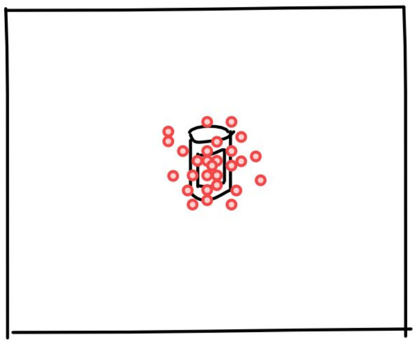
Stochasticity

Behavioral and neural measurements are inherently noisy.



Stochasticity

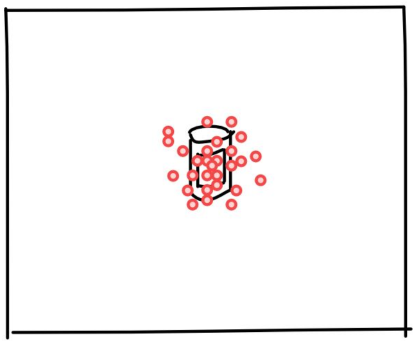
Behavioral and neural measurements are inherently noisy.



Stochasticity and probabilistic modeling

Behavioral and neural measurements are inherently noisy.

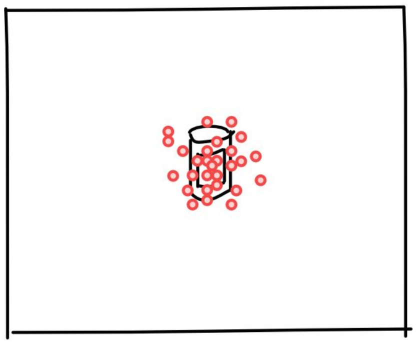
Probabilistic models specify a *noise model* that
(1) quantifies variability, and
(2) uses it for computation
under uncertainty



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

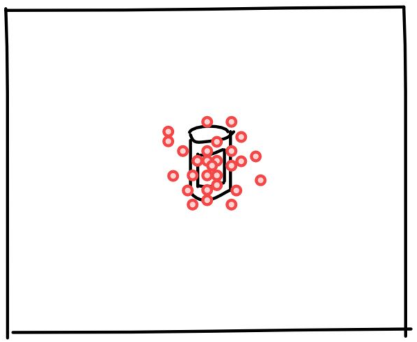


Inference in these models means estimating the value of a variable or parameter.

It is not assumed to map onto cognition/perception.

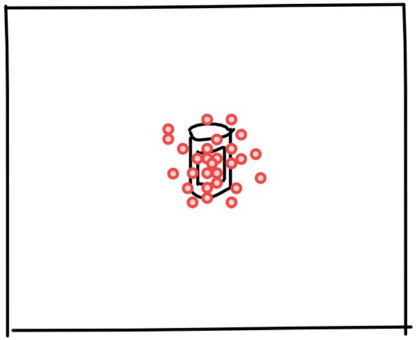
Random variables

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



Random variables

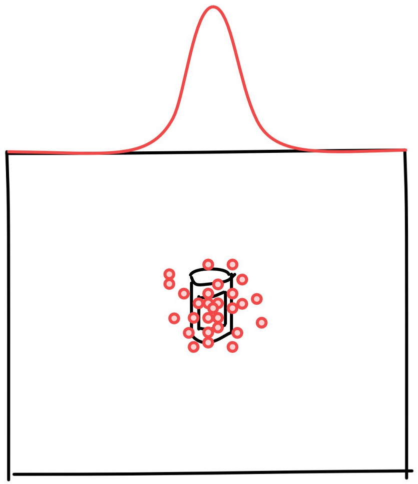
A variable X whose value is not deterministic. Its realizations are called *observations* x



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*

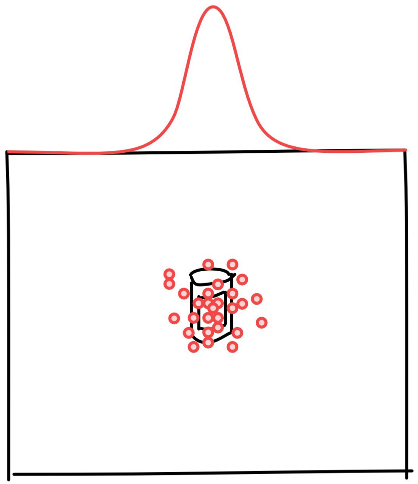


Observations $x \in \mathbb{R}^2$ are distributed according to a Gaussian probability density func.

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

Random variables and distributions

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

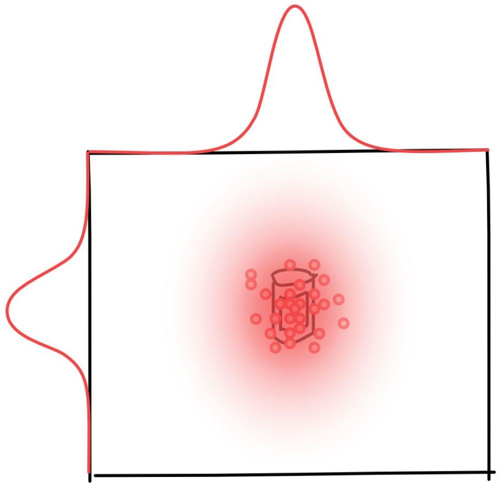


Observations $x \in \mathbb{R}^2$ are distributed according to a Gaussian probability density func.

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

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

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

Multivariate distributions

Multivariate distributions are *joint distributions* over random variables

$\boldsymbol{X} = (X_1, X_2, \ldots, X_d),$ with a joint PMF or PDF $\quad f_{X1,\, X2,\, \ldots,\, Xd}(x_1, x_2, \ldots, x_d)$

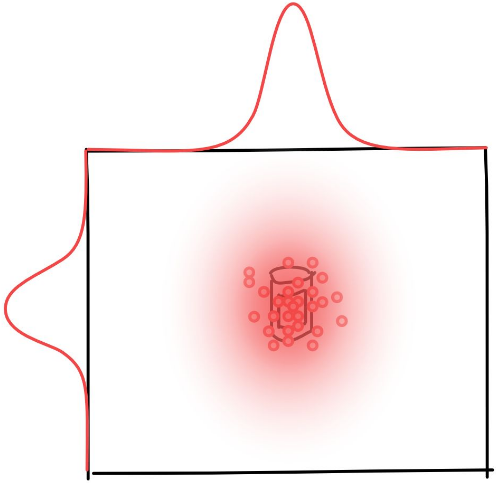
Multivariate distributions

Multivariate distributions are *joint distributions* over random variables $\boldsymbol{X} = (X_1, X_2, \dots, X_d)$, with a joint PMF or PDF $f_{X1, X2, \dots, Xd}(x_1, x_2, \dots, x_d)$

For the multivariate Gaussian

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

the covariance matrix Σ describes dependencies between variables



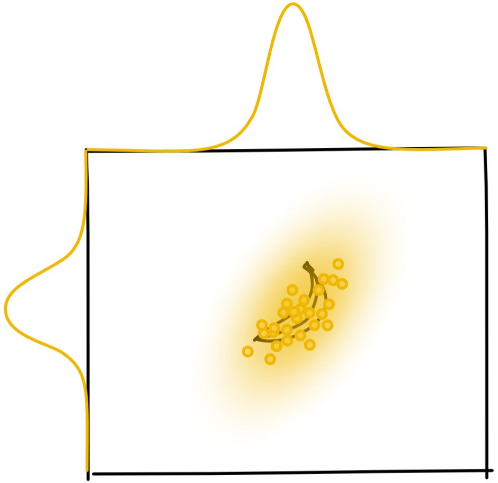
Multivariate distributions

Multivariate distributions are *joint distributions* over random variables $\boldsymbol{X} = (X_1, X_2, \ldots, X_d)$, with a joint PMF or PDF $f_{X1, X2, \ldots, Xd}(x_1, x_2, \ldots, x_d)$

For the multivariate Gaussian

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

the covariance matrix Σ describes dependencies between variables



Multivariate distributions

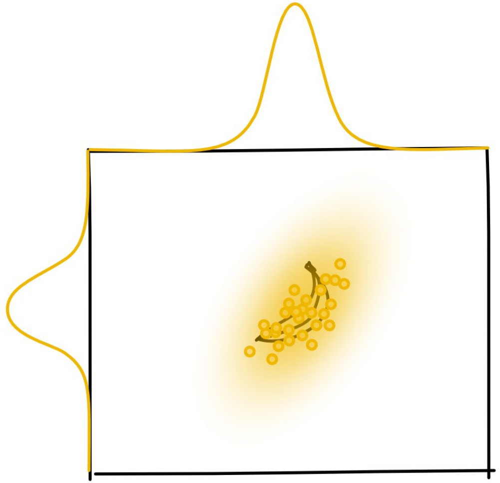
Multivariate distributions are *joint distributions* over random variables $\boldsymbol{X} = (X_1, X_2, \ldots, X_d)$, with a joint PMF or PDF $f_{X1, X2, \ldots, Xd}(x_1, x_2, \ldots, x_d)$

For the multivariate Gaussian

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

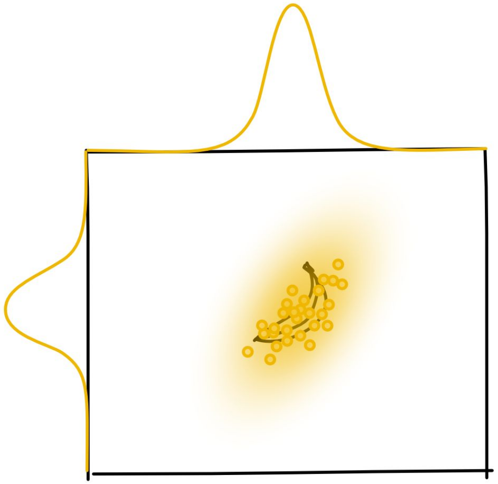
the covariance matrix Σ describes dependencies between variables

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



Inference

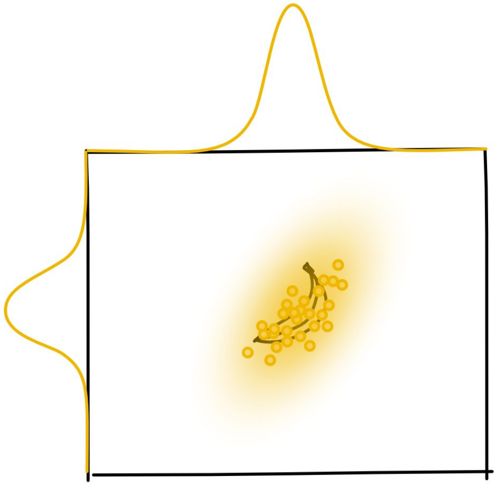
We have collected eye-tracking data. The saccade endpoint distribution might be approximated with a 2D Gaussian.



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

Inference

We have collected eye-tracking data. The saccade endpoint distribution might be approximated with a 2D Gaussian.

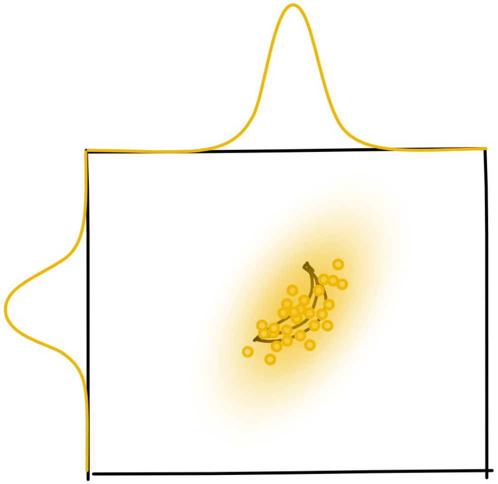


1) What is the target of saccades?
2) How much noise?
3) Is there structure in the noise?

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

Inference

We have collected eye-tracking data. The saccade endpoint distribution might be approximated with a 2D Gaussian.



1) What is the target of saccades?
2) How much noise?
3) Is there structure in the noise?

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

To answer these questions, we need to infer the values of the parameters.

Inference: Maximizing the likelihood function

Which are the parameter values that maximize the likelihood function?

→ We maximize

$$\ell(\theta \mid \mathbf{x_1}, \mathbf{x_2}, \ldots, \mathbf{x_n}) = \log L(\theta \mid \mathbf{x_1}, \mathbf{x_2}, \ldots, \mathbf{x_n})$$

Inference: Maximizing the likelihood function

Which are the parameter values that maximize the likelihood function?

→ We maximize

$$\ell(\theta \mid \mathbf{x_1}, \mathbf{x_2}, \ldots, \mathbf{x_n}) = \log L(\theta \mid \mathbf{x_1}, \mathbf{x_2}, \ldots, \mathbf{x_n})$$

That means deriving ℓ w.r.t. each of the parameters θ

E.g. for the multivariate Gaussian: $\frac{\partial \ell(\mu, \Sigma | \mathbf{x_1}, \mathbf{x_2}, \ldots, \mathbf{x_n})}{\partial \mu} = \sum_{i=1}^{n} \Sigma^{-1}(\mathbf{x_i} - \mu)$

Inference: Maximizing the likelihood function

Which are the parameter values that maximize the likelihood function?

→ We maximize

$$\ell(\theta \mid \mathbf{x_1}, \mathbf{x_2}, \ldots, \mathbf{x_n}) = \log L(\theta \mid \mathbf{x_1}, \mathbf{x_2}, \ldots, \mathbf{x_n})$$

That means deriving ℓ w.r.t. each of the parameters θ

E.g. for the multivariate Gaussian: $\frac{\partial \ell(\mu, \Sigma | \mathbf{x_1}, \mathbf{x_2}, \ldots, \mathbf{x_n})}{\partial \mu} = \sum_{i=1}^{n} \Sigma^{-1}(\mathbf{x_i} - \mu)$

→ set to zero, solve for μ . This is the *maximum likelihood estimate* (MLE).
→ same thing for Σ

The likelihood function in probabilistic terms

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

$L(\theta \mid \mathbf{x_1}, \mathbf{x_2}, \ldots, \mathbf{x_n})$

The likelihood function in probabilistic terms

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

$$L(\theta \mid \mathbf{x_1}, \mathbf{x_2}, \ldots, \mathbf{x_n})$$

What is the *probability of observations* ***x****, given the parameters* θ?

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

The likelihood function in probabilistic terms

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

$$L(\theta \mid \mathbf{x_1}, \mathbf{x_2}, \ldots, \mathbf{x_n})$$

What is the *probability of observations **x**, given the parameters* θ?

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

→ 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 θ

Bayesian inference: Key ingredients

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

Bayesian inference: Key ingredients

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

Bayesian inference: Key ingredients

- the *likelihood* $p(\mathbf{x} \mid \theta)$: evidence for our hypothesis about θ
- *the prior distribution* $p(\theta)$: before observing x, what's our belief and certainty about θ
- the posterior $p(\theta \mid \mathbf{x})$: after observing x, what's our updated belief about θ
- the marginal likelihood $p(\mathbf{x}) = \int p(\mathbf{x}|\theta)p(\theta)d\theta$: For any value θ 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 θ. We combine it with our *prior belief (prior distribution)* $p(\theta)$ about how the parameters are distributed to obtain the *posterior distribution* $p(\theta \mid \mathbf{x})$

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

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 θ. We combine it with our *prior belief (prior distribution)* $p(\theta)$ about how the parameters are distributed to obtain the *posterior distribution* $p(\theta \mid \mathbf{x})$

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

Bayes' theorem immediately follows from basic rules of probability,
specifically the product rule : p(**B**|**A**) = p(**A**,**B**) / p(**A**)
p(**A**,**B**) = p(**A**|**B**) / p(**B**)

Bayesian inference: Combining the likelihood with a prior

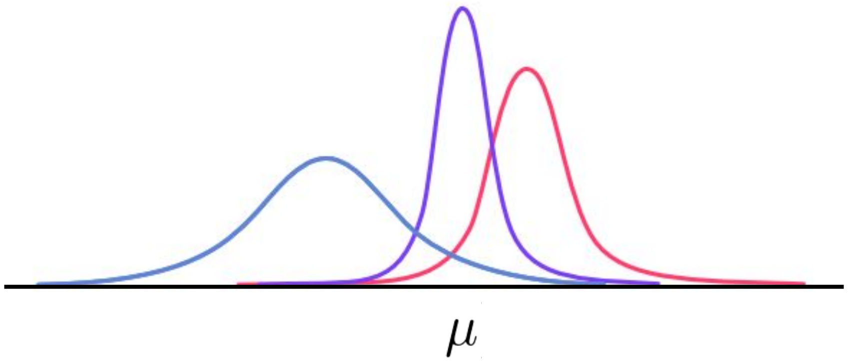
In Bayesian statistics, the *likelihood* $p(\mathbf{x} \mid \theta)$ is regarded as *evidence* in favor of a specific parameter set θ. We combine it with our *prior belief* (*prior distribution*) $p(\theta)$ about how the parameters are distributed to obtain the *posterior distribution* $p(\theta \mid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

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

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 θ. We combine it with our *prior belief* (*prior distribution*) $p(\theta)$ about how the parameters are distributed to obtain the *posterior distribution* $p(\theta \mid \mathbf{x})$

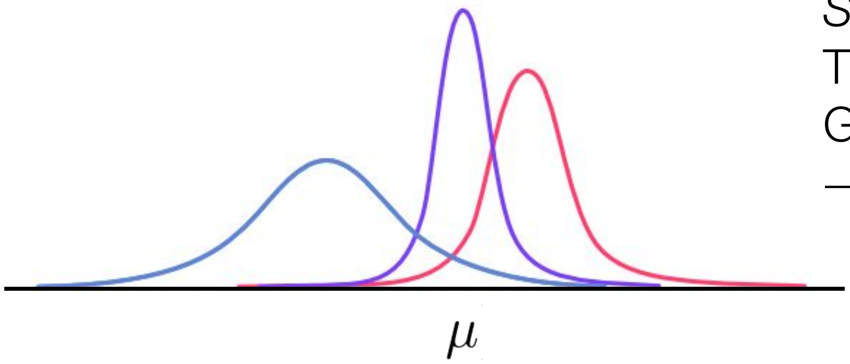
$$p(\theta \mid \mathbf{x}) = \frac{p(\mathbf{x}|\theta)p(\theta)}{p(\mathbf{x})}$$



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 θ. We combine it with our *prior belief* (prior distribution) $p(\theta)$ about how the parameters are distributed to obtain the *posterior distribution* $p(\theta \mid \mathbf{x})$

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



Side note:
The likelihood is not always Gaussian (e.g. for σ)
→ 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 θ. We combine it with our *prior belief* (*prior distribution*) $p(\theta)$ about how the parameters are distributed to obtain the *posterior distribution* $p(\theta \mid \mathbf{x})$

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

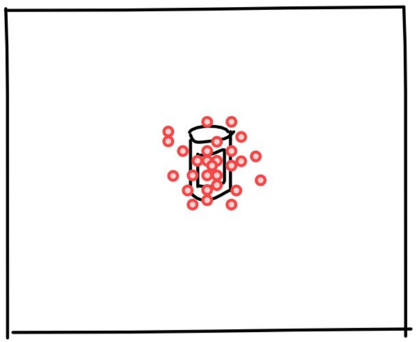
$p(\mathbf{x}) = \int p(\mathbf{x}|\theta)p(\theta)d\theta$ is a normalization constant to ensure that $p(\theta \mid \mathbf{x})$ integrates to 1. It is called the *marginal likelihood* (sometimes also called *evidence*).

Generative model

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

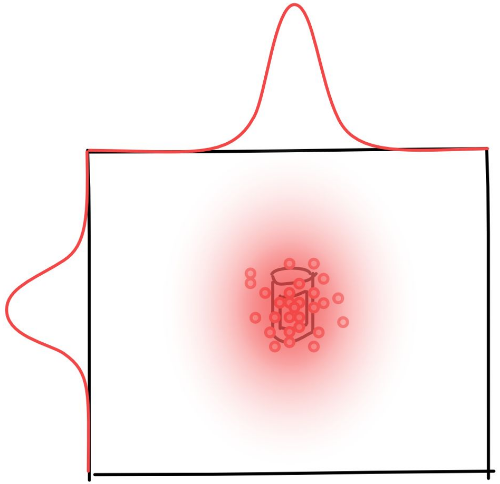
Generative model

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



Generative model

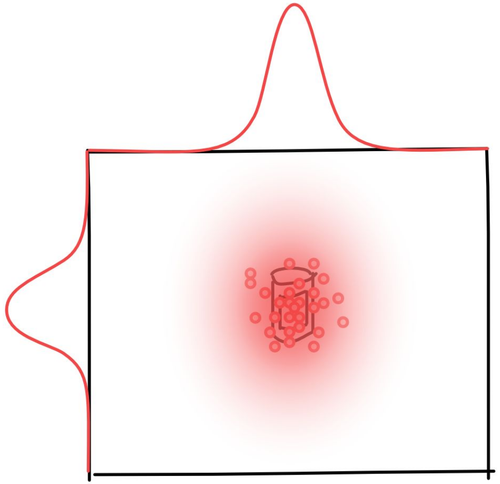
It formalizes our *knowledge* or our *hypothesis* about how data was generated.



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

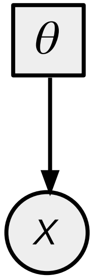
Generative model

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



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

θ



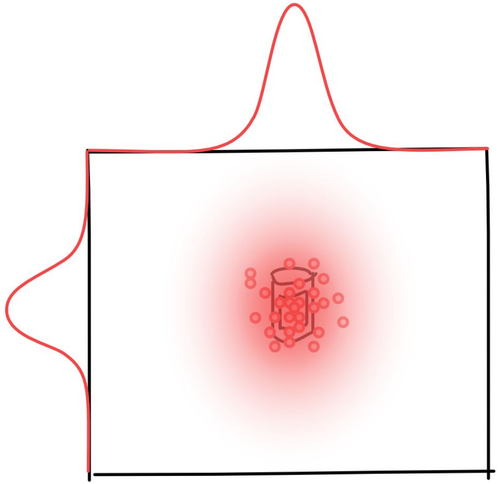
deterministic parameters

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

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

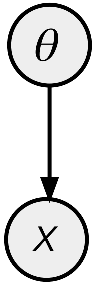
Generative model

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



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

$p(\theta)$



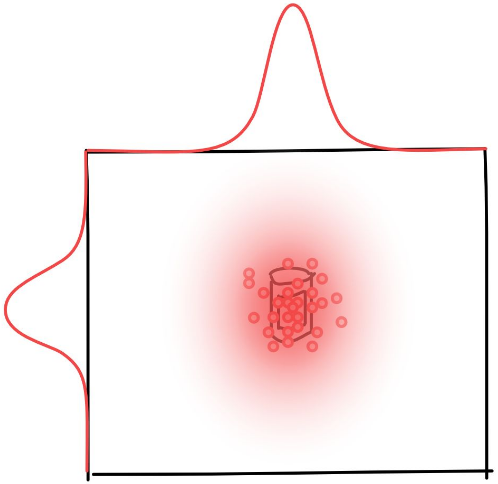
parameters as random variables

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

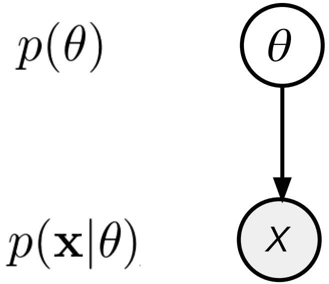
observed variables
$x = \{x_0, x_1, \dots, x_N\}$

Generative model

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



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



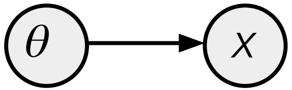
random variables
with unknown value
(:= *latent variable*)

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

Generative model: Joint distribution of the “complete dataset”

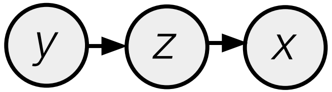
It formalizes our *knowledge* or our *hypothesis* about how data was generated.

(1)



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

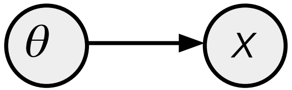
(2)



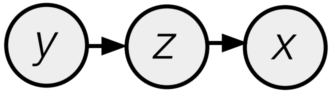
Generative model: Joint distribution of the "complete dataset"

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

(1)



(2)



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

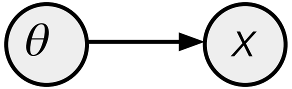
(1) $p(\mathbf{x}, \theta) = p(\mathbf{x}|\theta)p(\theta)$

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

Generative model: Sampling

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

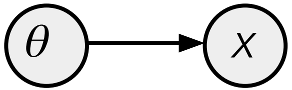
(1)



Generative model: Sampling

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

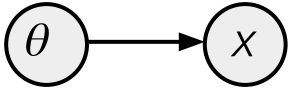
(1) $p(\mathbf{x}, \theta) = p(\mathbf{x}|\theta)p(\theta)$ with a Gaussian observation model:



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:

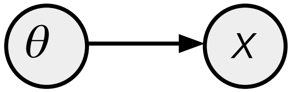


- sample $\theta = \{\mu, \Sigma\}$ from $p(\theta)$ (or assume fixed values)

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:

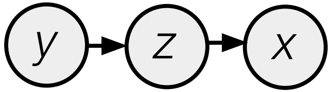


- sample $\theta = \{\mu, \Sigma\}$ from $p(\theta)$ (or assume fixed values)
- then, with fixed $\theta = \{\mu, \Sigma\}$, sample x from $p(\mathbf{x}|\theta) = \mathcal{N}(\mathbf{x}; \mu, \Sigma)$:
  $\mathbf{x}_i|\mu, \Sigma \sim \mathcal{N}(\mu, \Sigma)$

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



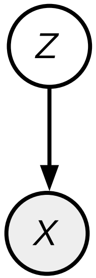
- sample *y*: $U \sim \mathrm{Uniform}(0,1)$
  $\mathbf{y} = 1_{\{U \leq p_{\mathbf{y}}\}}$

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

- sample *x* analogously

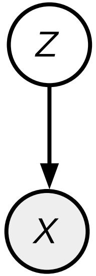
Latent variable models

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.

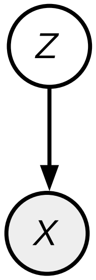


Two challenges:

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:

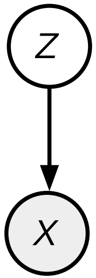
1) Infer the values of the latent variable
2) Estimate parameters of the model under uncertainty

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

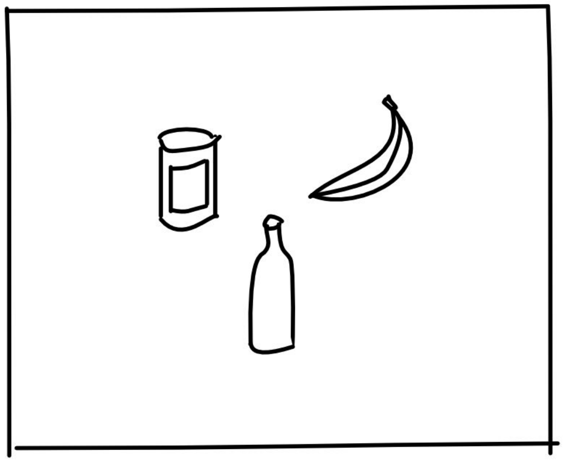
General recipe:

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

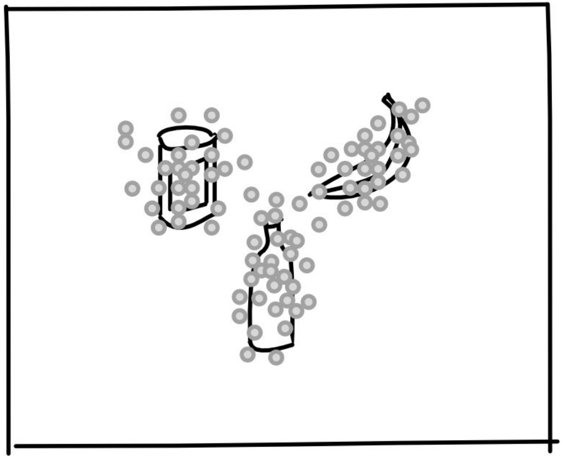
Mixture Models and Expectation Maximization

Mixture models

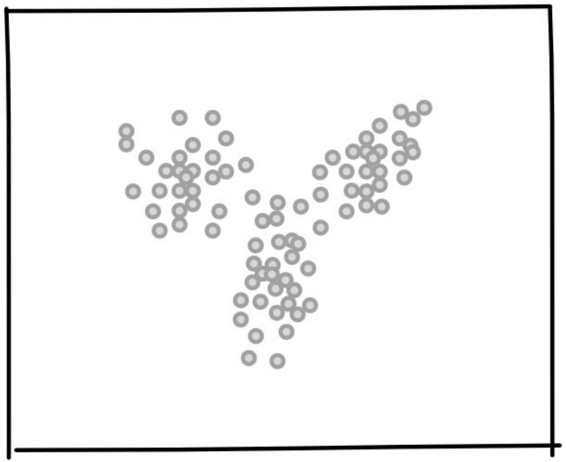
Experiment:
In each trial, subjects have to choose between three objects.



Mixture models



Mixture models

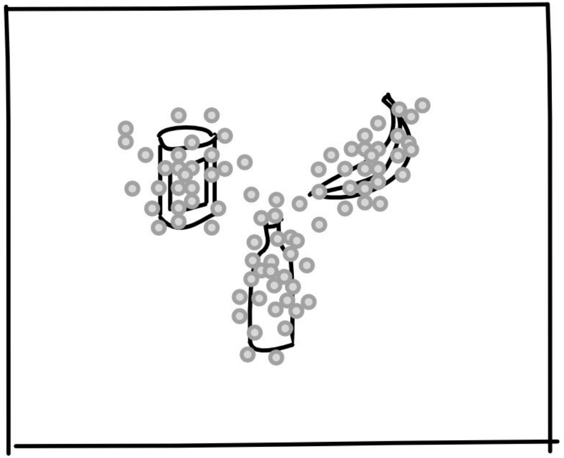


Mixture models

Hypothesis:
For each trial, subjects

(1) choose one of three objects, and

(2) make a noisy saccade to the object's center

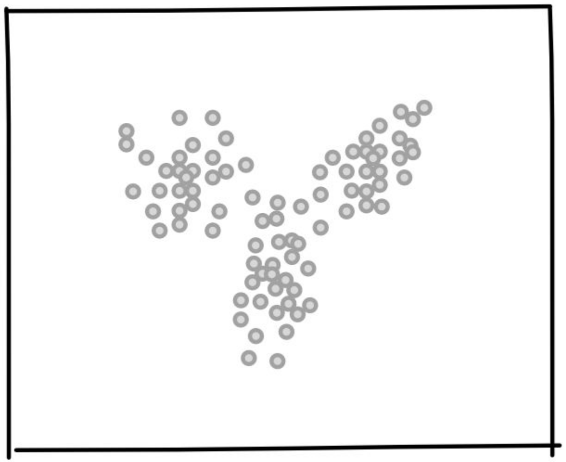


Mixture models

Hypothesis:
For each trial, subjects

(1) choose one of three objects, and

(2) make a noisy saccade to the object's center



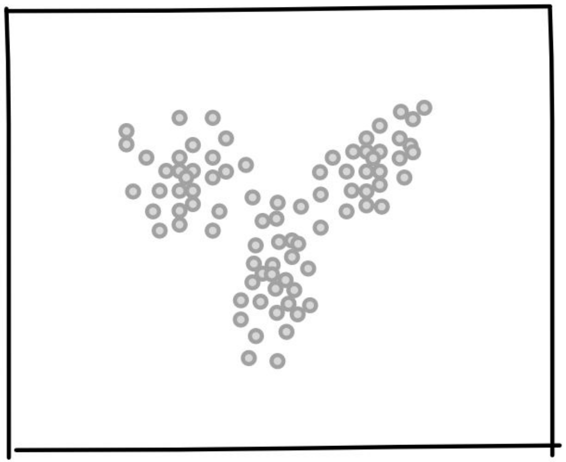
We have only observed saccades $\mathbf{x} \in \mathbb{R}^{N \times D}$

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

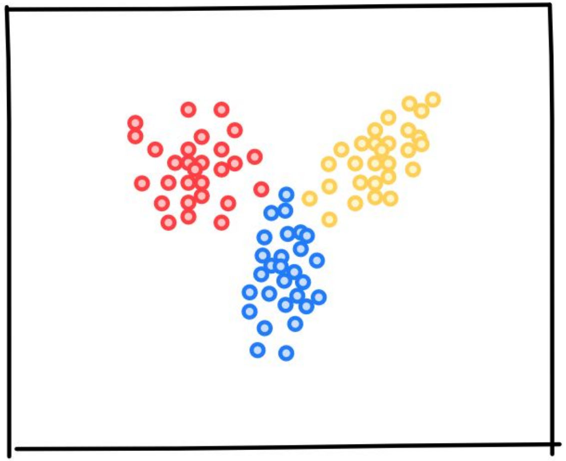
We have only observed saccades $\mathbf{x} \in \mathbb{R}^{N \times D}$

Mixture models

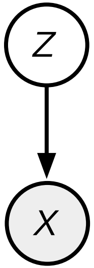
Hypothesis:
For each trial, subjects

(1) choose one of three objects, and

(2) make a noisy saccade to the object's center



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



$z = (1,0,0)$
$z = (0,1,0)$
$z = (0,0,1)$

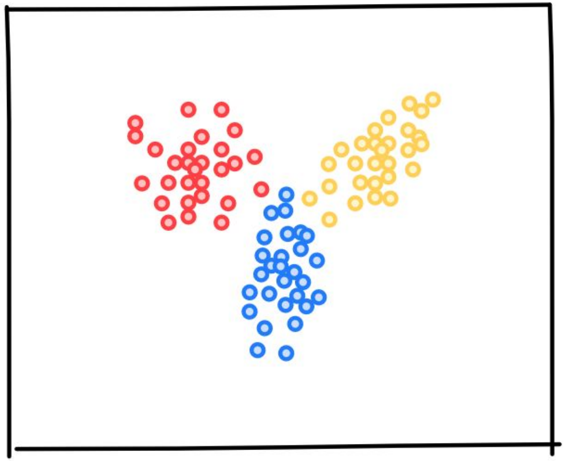
We have only observed saccades $\mathbf{x} \in \mathbb{R}^{N \times D}$

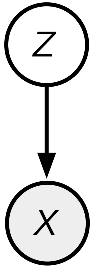
Mixture models

Hypothesis:
For each trial, subjects

(1) choose one of three objects, and

(2) make a noisy saccade to the object's center





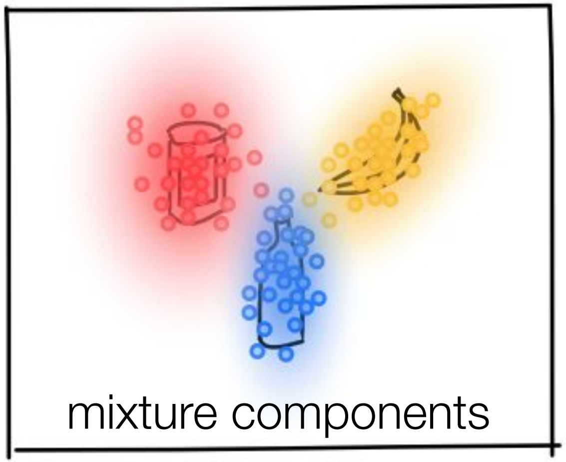
Joint distribution:

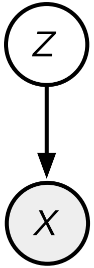
$$p(\mathbf{x}, \mathbf{z}) = p(\mathbf{x}|\mathbf{z})p(\mathbf{z})$$

Mixture models

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

For each class *k*, observations ***x*** are distributed as a class-specific Gaussian.





Joint distribution:

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

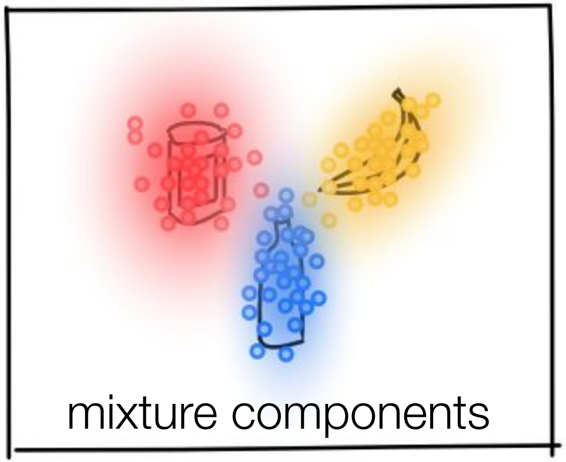
Mixture models

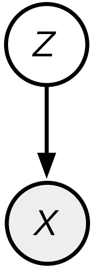
$p(\mathbf{x}|z_k = 1) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$

For each class *k*, observations ***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:

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

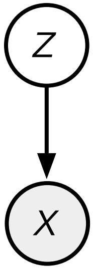
Mixture models

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

For each class *k*, observations ***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:

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

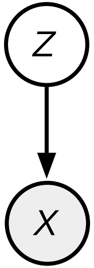
Mixture models

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

For each class *k*, observations ***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:

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

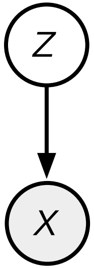
Mixture models

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

For each class *k*, observations ***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:

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

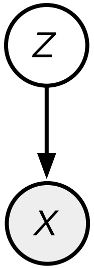
Mixture models

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

For each class *k*, observations ***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.



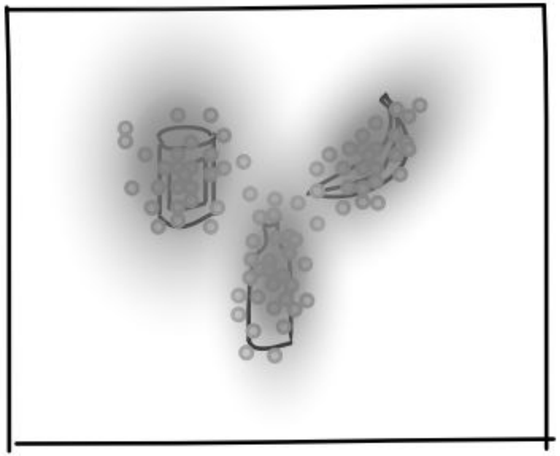
Joint distribution:

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

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

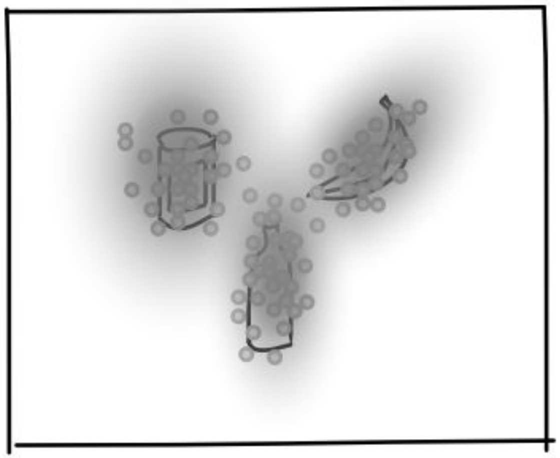


$$p(\mathbf{x}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

Mixture models

How do we estimate class-specific parameters?

Which data point belongs to which class?

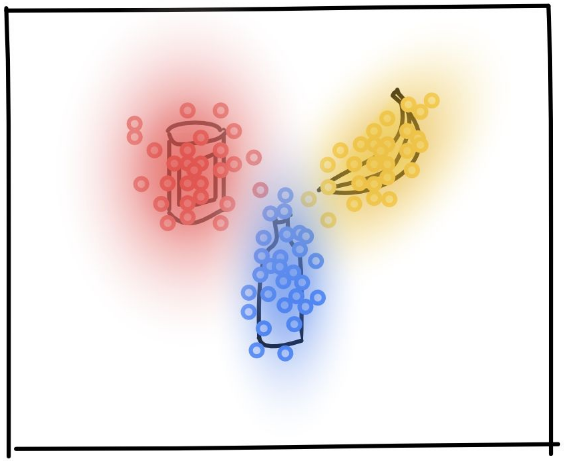


$$p(\mathbf{x}) = \sum_{k=1}^{K} \boxed{\pi_k} \mathcal{N}(\mathbf{x} \boxed{\boldsymbol{\mu}_k}, \boxed{\boldsymbol{\Sigma}_k})$$

Mixture models

How do we estimate class-specific parameters?

Which data point belongs to which class?



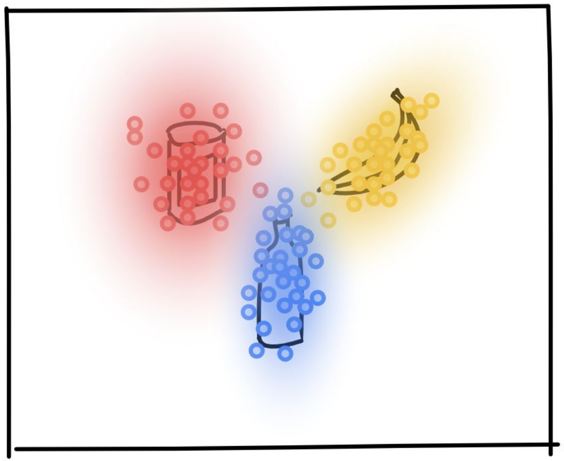
$$p(\mathbf{x}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

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

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

Mixture models

0) initialize θ randomly



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

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

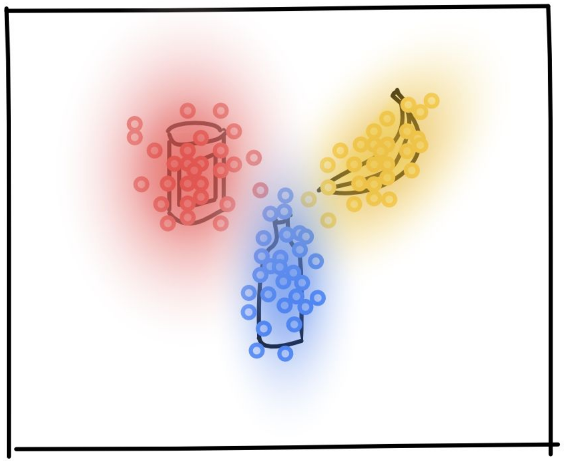
Mixture models

0) initialize θ randomly

1) calculate *responsibilities*

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

posterior probabilities that datapoint *i* belongs to class *k*



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

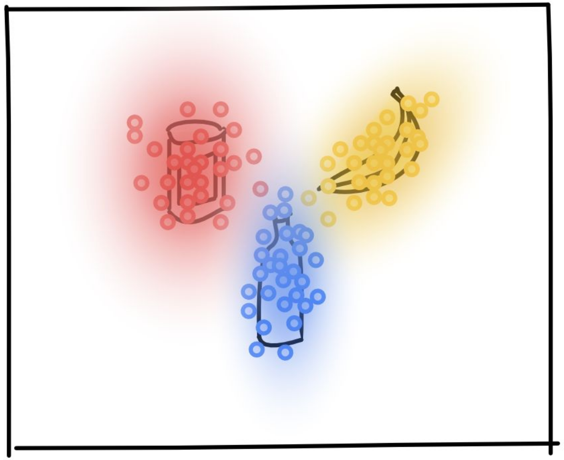
$$p(\mathbf{x}, \mathbf{z} | \theta) = p(\mathbf{x} | \mathbf{z}, \theta) p(\mathbf{z} | \theta)$$

Mixture models

0) initialize θ randomly

1) calculate *responsibilities* $\gamma_{ik} = p(z_{ik} = 1 | x_i, \theta)$, *posterior probabilities* that datapoint *i* belongs to class *k*

2) update θ by maximizing a LL where data points are weighted by γ_{ik}



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

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

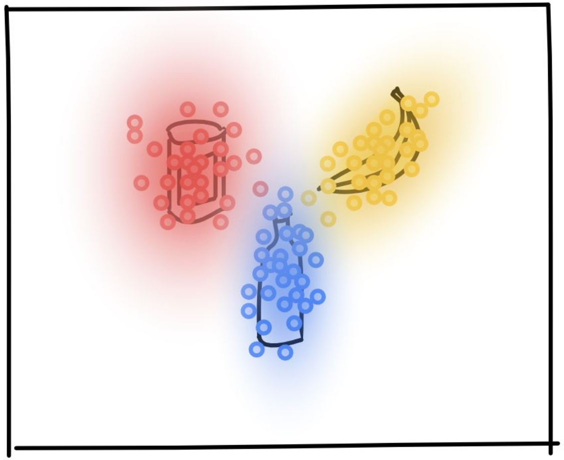
Mixture models

0) initialize θ randomly

1) calculate *responsibilities* $\gamma_{ik} = p(z_{ik} = 1 | x_i, \theta)$, *posterior probabilities* that datapoint *i* belongs to class *k*

2) update θ by maximizing a LL where data points are weighted by γ_{ik}

Iterate 1) and 2) til convergence



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

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

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 ***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***

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

$$= \frac{\pi_k \mathcal{N}(\mathbf{x};\mu_k,\Sigma_k)}{\sum_K \pi_k \mathcal{N}(\mathbf{x};\mu_k,\Sigma_k)}$$

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 ***z***

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

$$= \frac{\pi_k \mathcal{N}(\mathbf{x};\mu_k,\Sigma_k)}{\sum_K \pi_k \mathcal{N}(\mathbf{x};\mu_k,\Sigma_k)}$$

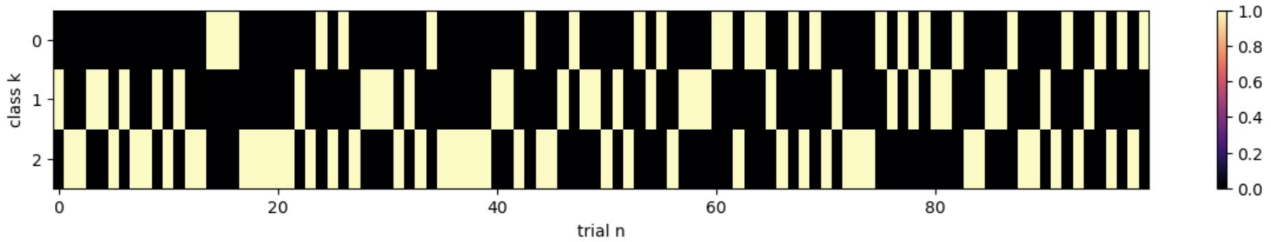
Then: plug in current parameter estimates, evaluate PDF for each datapoint and each class, normalize

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

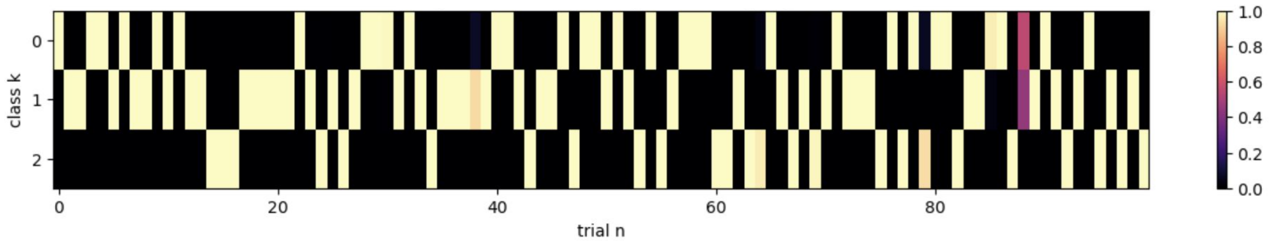
Expectation maximization (EM)

EM tackles two problems:

E-Step: Determine which data point belongs to which class
→ Inference of the **posterior** over the latent indicator variable ***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** $p(\boldsymbol{z}|\boldsymbol{x}, \theta)$ over the latent indicator variable ***z***

M-Step: Fit class-specific parameters

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

Expectation maximization (EM)

EM tackles two problems:

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

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

M-Step: Fit class-specific parameters

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

→ Problem: We don't know the value of ***z***

Expectation maximization (EM)

EM tackles two problems:

E-Step: Determine which data point belongs to which class
→ Inference of the **posterior** $p(\boldsymbol{z}|\boldsymbol{x}, \theta)$ over the latent indicator variable ***z***

M-Step: Fit class-specific parameters
→ Maximize the *complete-data log LL* $\ln p(\boldsymbol{x}, \boldsymbol{z}|\theta)$
→ Problem: We don't know the value of ***z***

Solution: Optimize expected value under the posterior distribution of ***z***

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

Expectation maximization (EM)

EM tackles two problems:

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

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

M-Step: Fit class-specific parameters

→ Find parameters that optimize expected complete-data log likelihood

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

Expectation maximization (EM)

EM tackles two problems:

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

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

M-Step: Fit class-specific parameters

→ Find parameters that optimize expected complete-data log likelihood

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

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

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** $p(\boldsymbol{z}|\boldsymbol{x},\theta)$ over the latent indicator variable ***z***

M-Step: Fit class-specific parameters

→ Find parameters that optimize expected complete-data log likelihood

$$\mathbb{E}_{\mathbf{z}}\left[\ln p(\boldsymbol{x},\boldsymbol{z}|\theta')\right] = \sum_{z} \underbrace{p(\boldsymbol{z}|\boldsymbol{x},\theta)}_{\text{Posterior probabilities of each class for each datapoint}} \ln \underbrace{p(\boldsymbol{x},\boldsymbol{z}|\theta')}_{p(\mathbf{x}|\mathbf{z},\theta')p(\mathbf{z}|\theta')}$$

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** $p(\boldsymbol{z}|\boldsymbol{x}, \theta)$ over the latent indicator variable ***z***

M-Step: Fit class-specific parameters

→ Find parameters that optimize expected complete-data log likelihood

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

Class-specific likelihoods are summed over all possible values of ***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** $p(\boldsymbol{z}|\boldsymbol{x},\theta)$ over the latent indicator variable ***z***

M-Step: Fit class-specific parameters

→ Find parameters that optimize expected complete-data log likelihood

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

Iterate

Expectation maximization (EM)

EM tackles two problems *for any model with observations **x** depending on latents **z**:*

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

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

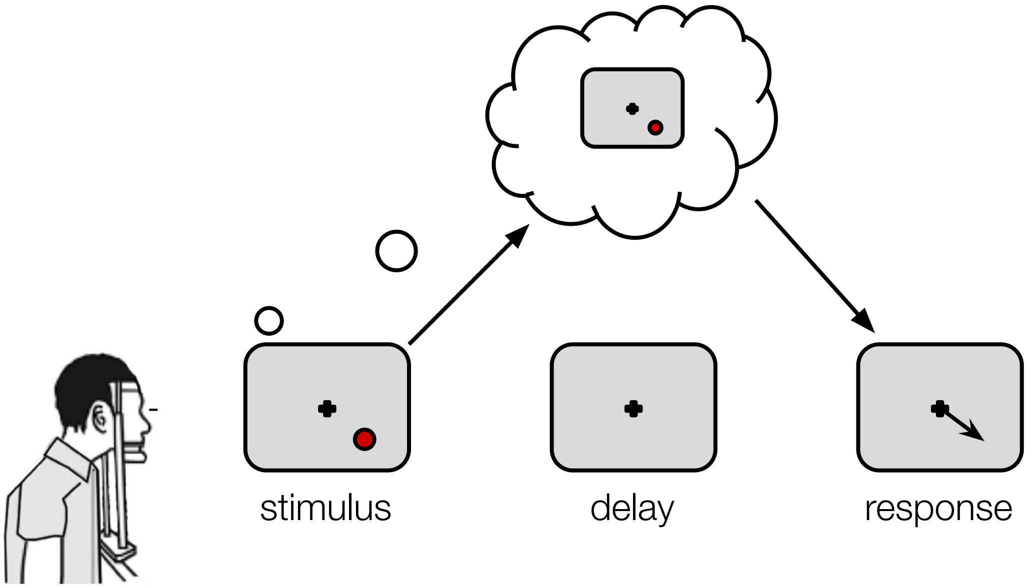
M-Step: Fit class-specific parameters

→ Find parameters that optimize expected complete-data log likelihood

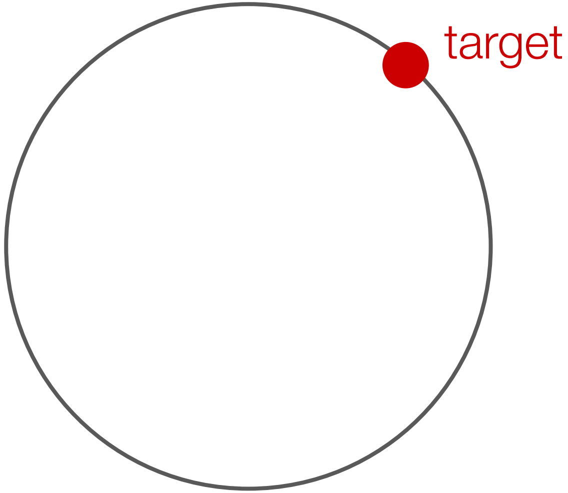
$$\mathbb{E}_{\mathbf{z}}\left[\ln p(\boldsymbol{x},\boldsymbol{z}|\theta')\right] = \sum_{z} p(\boldsymbol{z}|\boldsymbol{x},\theta) \ln p(\boldsymbol{x},\boldsymbol{z}|\theta')$$

Iterate

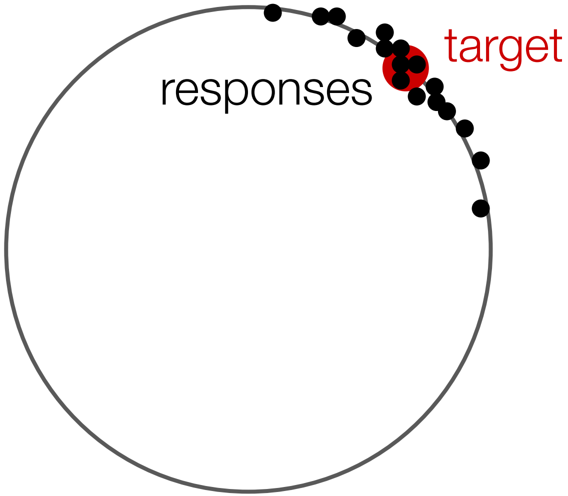
Example of mixture models: classic WM task



Example of mixture models: classic WM task

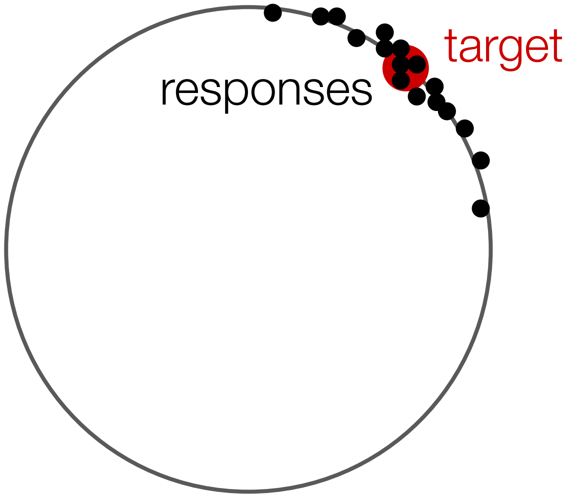


Example of mixture models: classic WM task

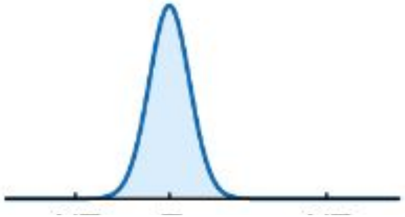


How precise is the working memory representation?

Example of mixture models: classic WM task

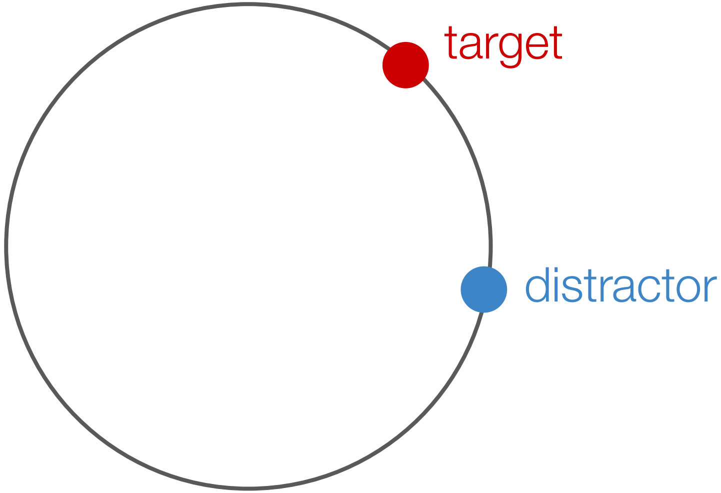


How precise is the working memory representation?

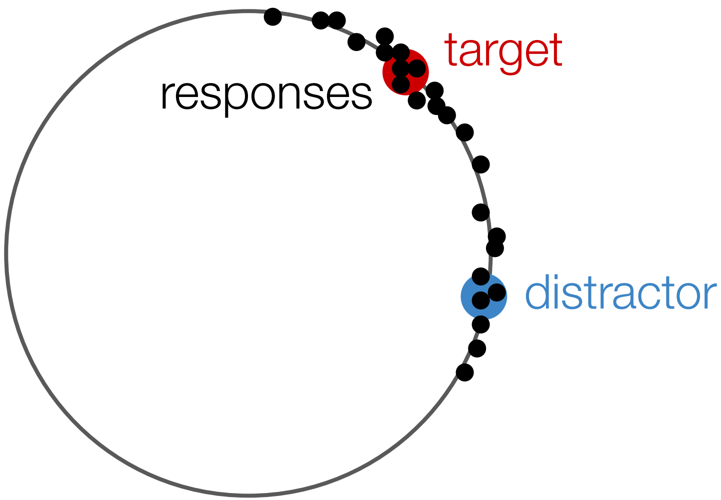


$$\theta = \{\mu, \sigma^2\}$$

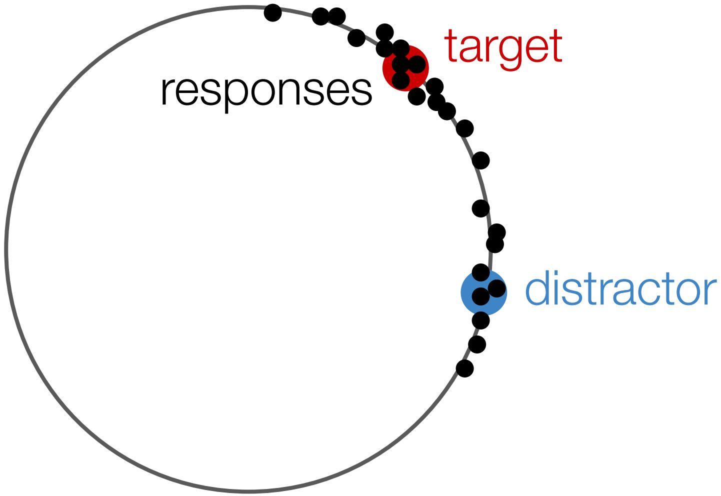
Example of mixture models: classic WM task



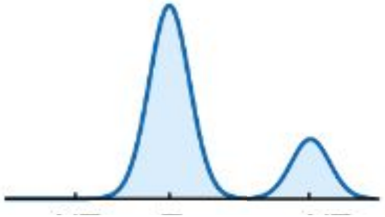
Example of mixture models: classic WM task



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

$$x^{(1)} = x_{saccade} - x_{target} \qquad x^{(2)} = x_{saccade} - x_{distractor}$$

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

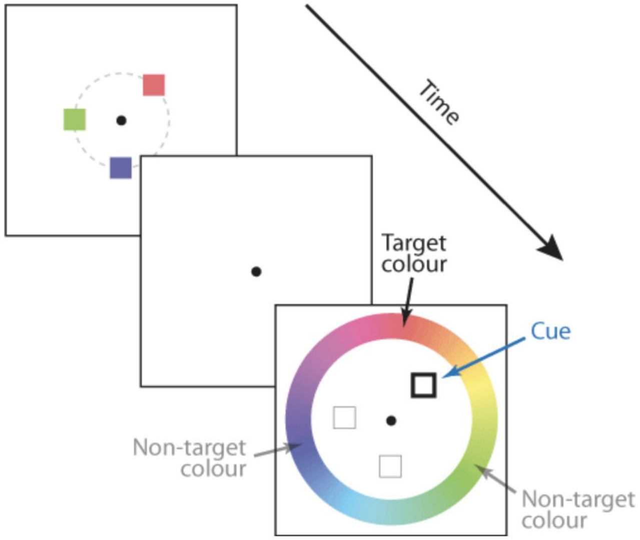
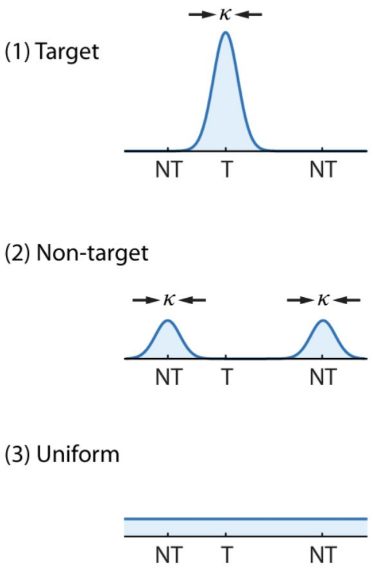


Figure 1 | The colour report task.

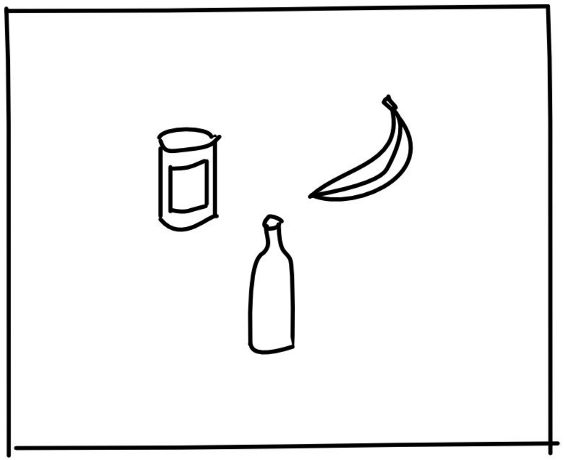


Bays, Catalao & Husain, *J Vis* (2009); Schneegans & Bays, *Cortex* (2016)

Hidden Markov Models

Hidden Markov models

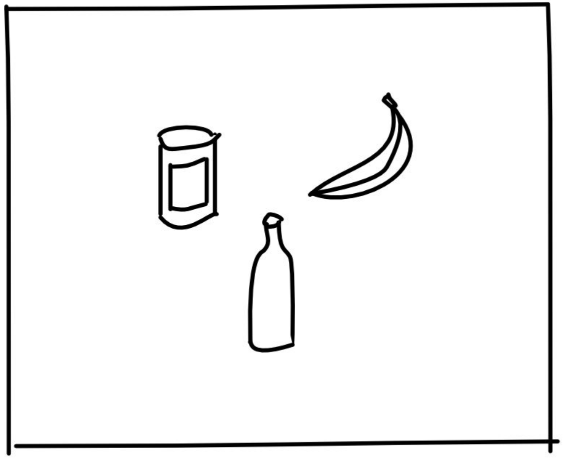
Experiment:
In each trial, subjects have to choose between three objects.

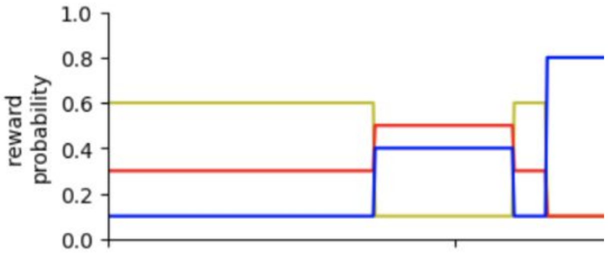


Hidden Markov models

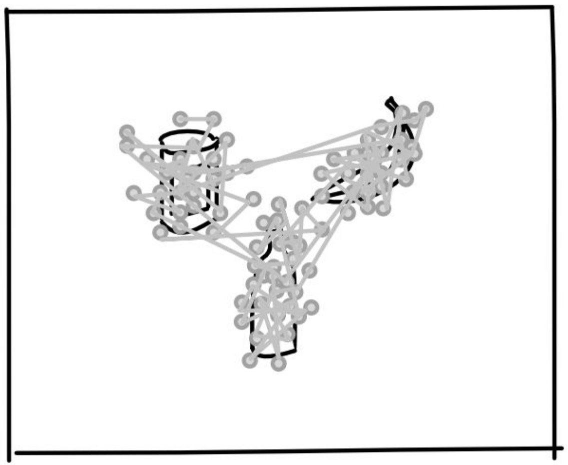
Experiment:
In each trial, subjects have to choose between three objects.

The relative value of objects fluctuates.

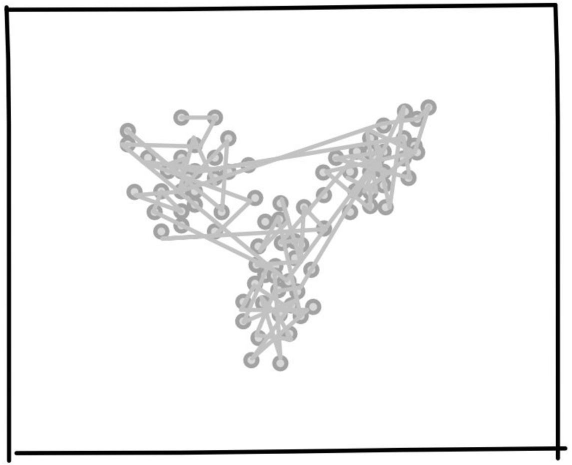




Hidden Markov models



Hidden Markov models



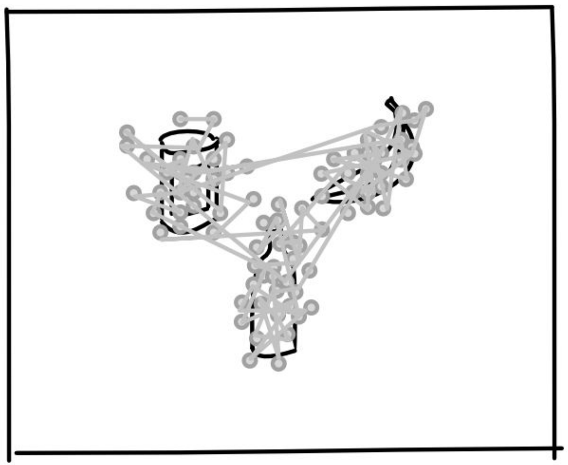
Hidden Markov models

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



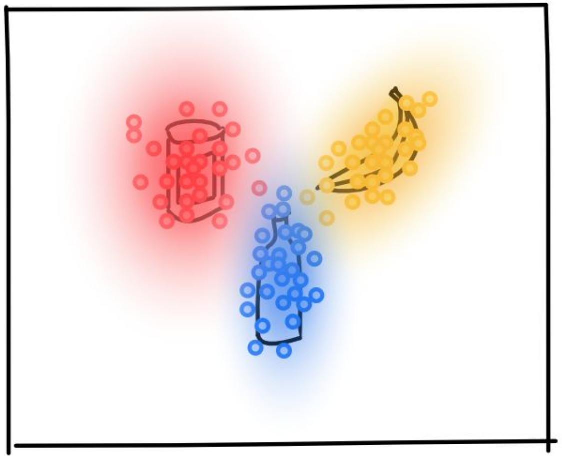
Hidden Markov models

Hypothesis:
For each trial, subjects

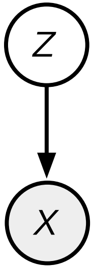
(1) choose one of three objects

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

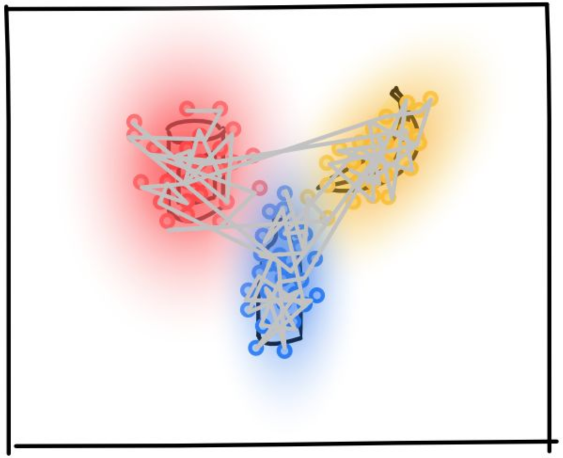
Hidden Markov models

Hypothesis:
For each trial, subjects

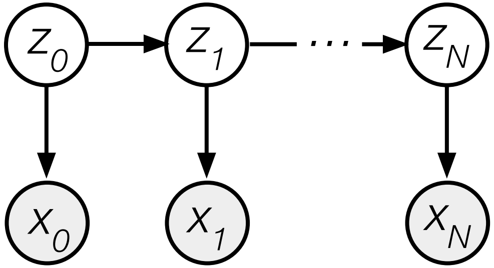
(1) choose one of three objects

(2) which object they choose depends on their previous choice

(3) make a noisy saccade to the object's center



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

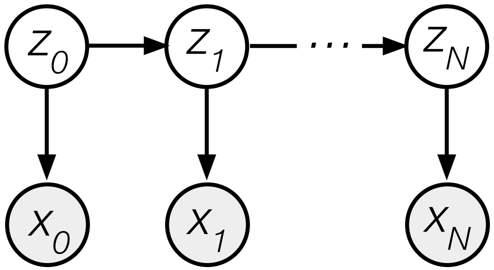


observations $\mathbf{x} \in \mathbb{R}^{N \times D}$

→ temporal dependencies in the latent variable

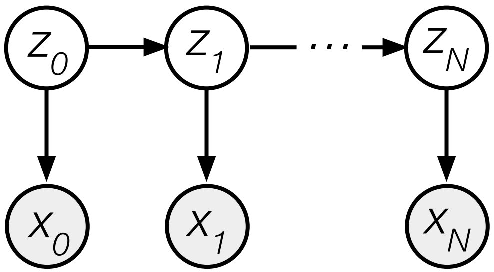
Hidden Markov models

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



Hidden Markov models

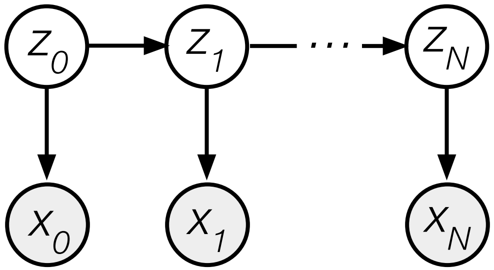
To capture sequential dependencies in choice behavior, we need an explicit model of transition structure of a latent variable



Markov property: $p(\mathbf{z_t}|\mathbf{z_0}, \mathbf{z_1}, ...\mathbf{z_{t-1}}) = p(\mathbf{z_t}|\mathbf{z_{t-1}})$

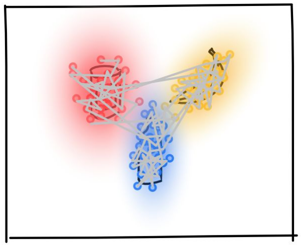
Hidden Markov models

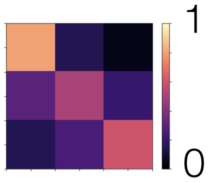
To capture sequential dependencies in choice behavior, we need an explicit model of transition structure of a latent variable



Markov property: $p(\mathbf{z_t}|\mathbf{z_0}, \mathbf{z_1}, ...\mathbf{z_{t-1}}) = p(\mathbf{z_t}|\mathbf{z_{t-1}})$

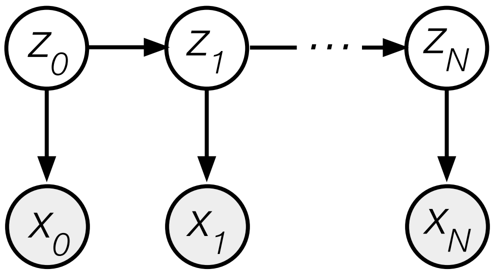
→ We can summarize transition structure in the transition matrix ***A***, with $A_{jk} \equiv p(z_{nk} = 1|z_{n-1,j} = 1)$





Hidden Markov models

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

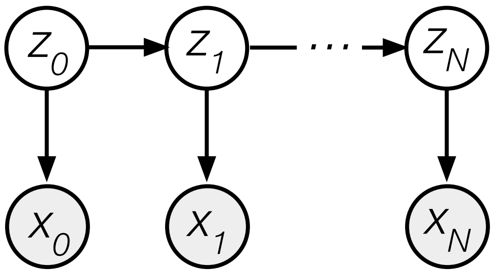


Joint distribution:

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

Hidden Markov models

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

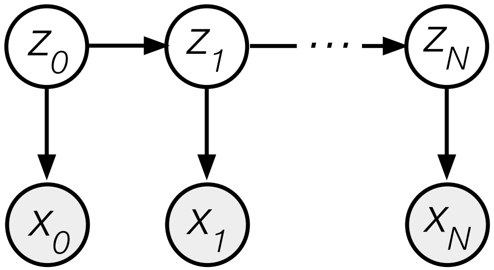


Joint distribution:

$$p(\boldsymbol{x}, \boldsymbol{z}|\theta) = \underbrace{p(\boldsymbol{z}_0)}_{\text{initial state}} \underbrace{\prod_{n=1}^{N} p(\boldsymbol{z}_n|\boldsymbol{z}_{n-1})}_{\text{transition matrix}} \underbrace{\prod_{n=0}^{N} p(\boldsymbol{x}_n|\boldsymbol{z}_n, \theta)}_{\text{likelihoods}}$$

Hidden Markov models

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

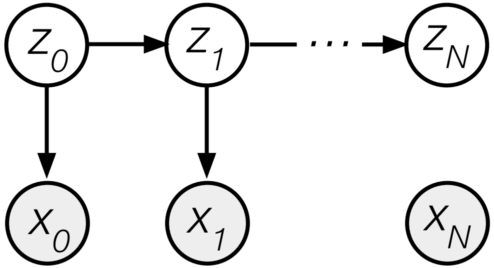


Joint distribution:

$$p(\boldsymbol{x}, \boldsymbol{z}|\theta) = \underbrace{p(\boldsymbol{z}_0)}_{\text{initial state}} \underbrace{\prod_{n=1}^{N} p(\boldsymbol{z}_n|\boldsymbol{z}_{n-1})}_{\text{transition matrix}} \underbrace{\prod_{n=0}^{N} p(\boldsymbol{x}_n|\boldsymbol{z}_n, \theta)}_{\text{likelihoods}}$$

π $\quad A \quad$ $\mathcal{N}(\boldsymbol{x}_n; \mu_k, \Sigma_k)$

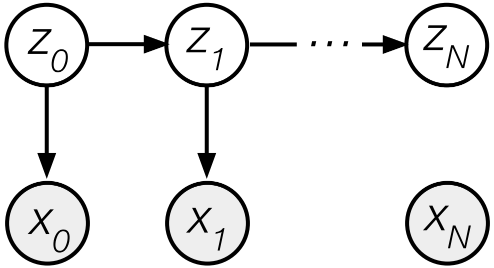
EM for Hidden Markov Models



Joint distribution:

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

EM for Hidden Markov Models

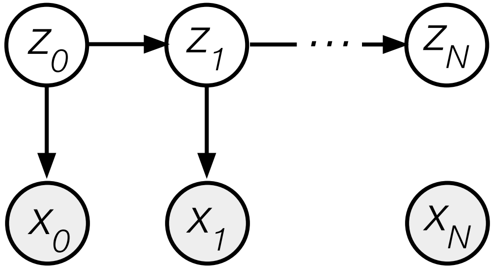


Joint distribution:

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

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

EM for Hidden Markov Models



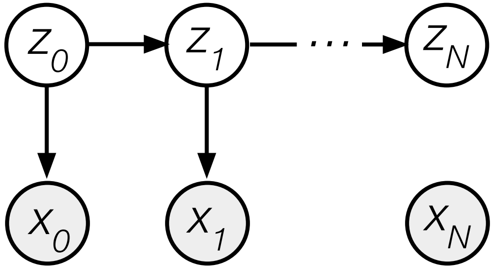
Joint distribution:

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

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

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

EM for Hidden Markov Models



Joint distribution:

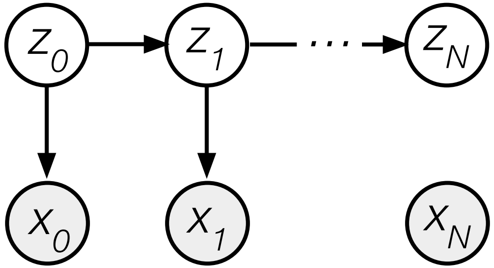
$$p(\boldsymbol{x}, \boldsymbol{z}|\theta) = p(\boldsymbol{z}_0) \prod_{n=1}^{N} p(\boldsymbol{z}_n|\boldsymbol{z}_{n-1}) \prod_{n=0}^{N} p(\boldsymbol{x}_n|\boldsymbol{z}_n, \theta)$$

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

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

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

EM for Hidden Markov Models



Joint distribution:

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

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

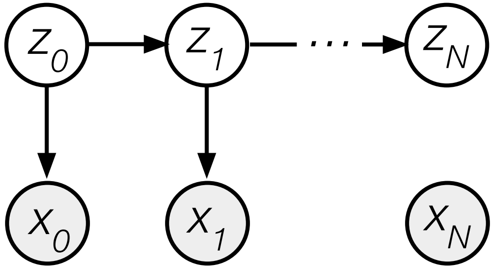
E-Step: infer posteriors, and calculate initial and state transition probabilities $\boldsymbol{\pi}, \boldsymbol{A}$

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

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

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

EM for Hidden Markov Models

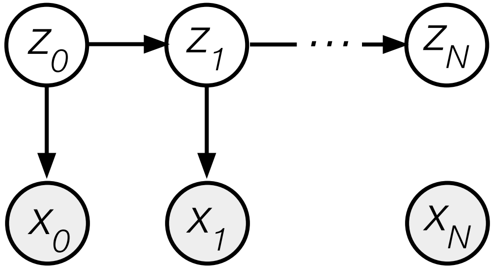


Joint distribution:

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

E-Step: infer posteriors

EM for Hidden Markov Models



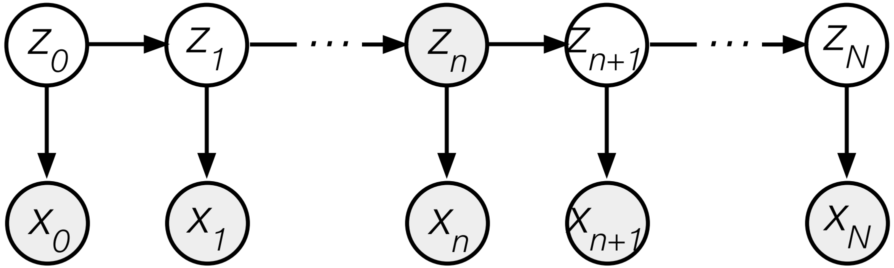
Joint distribution:

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

E-Step: infer posteriors
→ In HMMs, there are two posteriors over $\boldsymbol{z}$:

$p(\boldsymbol{z}_n \vert \boldsymbol{x}, \theta)$	probability of each latent state value, given observations
$p(\boldsymbol{z}_n, \boldsymbol{z}_{n-1} \vert \boldsymbol{x}, \theta)$	probability of observing a pair of subsequent states, – “ –

EM for Hidden Markov Models

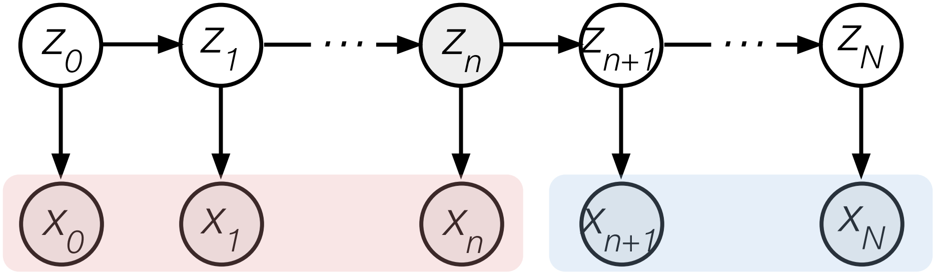


E-Step: infer posteriors

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

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

EM for Hidden Markov Models



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

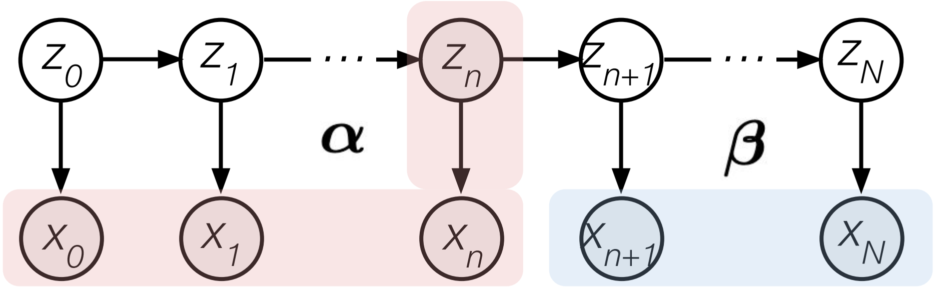
$$p(\boldsymbol{x}_{0:n}|\boldsymbol{z}_n, \theta), \quad p(\boldsymbol{x}_{n+1:N}|\boldsymbol{z}_n, \theta)$$

E-Step: infer posteriors

→ again, we start from Bayes theorem

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

EM for Hidden Markov Models



We include $p(\boldsymbol{z}_n)$:

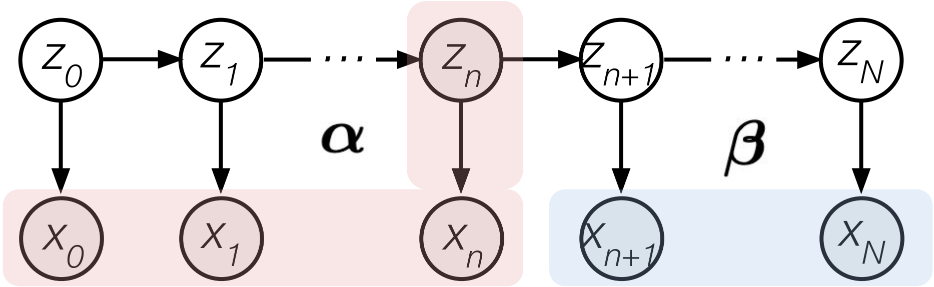
$$\underbrace{p(\boldsymbol{x}_{0:n}, \boldsymbol{z}_n|\theta)}_{\boldsymbol{\alpha}}, \underbrace{p(\boldsymbol{x}_{n+1:N}|\boldsymbol{z}_n, \theta)}_{\boldsymbol{\beta}}$$

E-Step: infer posteriors

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

→ Then the posterior becomes $$p(\boldsymbol{z}_n|\boldsymbol{x}_{0:N}, \theta) = \frac{\boldsymbol{\alpha}(\boldsymbol{z}_n)\boldsymbol{\beta}(\boldsymbol{z}_n)}{p(\boldsymbol{x}_{0:N})}$$

EM for Hidden Markov Models

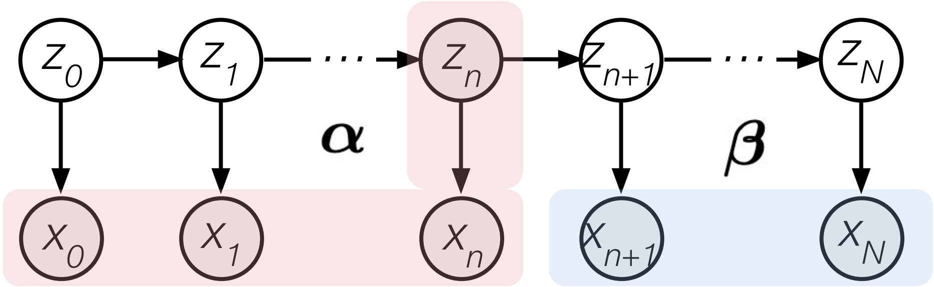


E-Step: infer posteriors

→ **Good news:**

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

EM for Hidden Markov Models

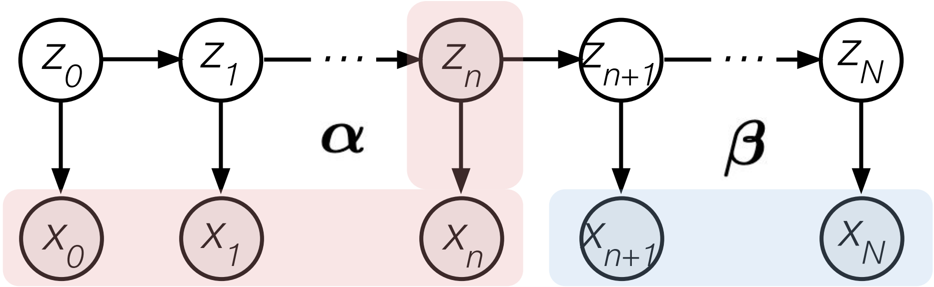


E-Step: infer posteriors

→ **Good news:**

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

EM for Hidden Markov Models



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

E-Step:

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

M-Step:

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

Remarks on emission models

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

Other (including more complex) emission models are possible:

- Student-t, etc… (continuous observations ***x***)
- Categorical emissions, Poisson emissions etc (discrete observations ***x***)

Remarks on emission models

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

Other (including more complex) emission models are possible:

- Student-t, etc… (continuous observations ***x***)
- Categorical emissions, Poisson emissions etc (discrete observations ***x***)

- Linear models of input ***u***: $p(\boldsymbol{x}|z_k = 1, \theta_k) = \mathcal{N}(\boldsymbol{x}|\boldsymbol{W}_k\boldsymbol{u} + c_k, \Sigma_k)$

- Autoregressive models: $p(\boldsymbol{x}_n|z_k = 1, \theta_k) = \mathcal{N}(\boldsymbol{x}_n|\boldsymbol{A}_k\boldsymbol{x}_{n-1} + d_k, \Sigma_k)$

Remarks on emission models

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

Other (including more complex) emission models are possible:

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

- Linear models of input $\boldsymbol{u}$: $p(\boldsymbol{x}|z_k = 1, \theta_k) = \mathcal{N}(\boldsymbol{x}|\boldsymbol{W}_k\boldsymbol{u} + c_k, \Sigma_k)$

- Autoregressive models: $p(\boldsymbol{x}_n|z_k = 1, \theta_k) = \mathcal{N}(\boldsymbol{x}_n|\boldsymbol{A}_k\boldsymbol{x}_{n-1} + d_k, \Sigma_k)$

- Combine them, you get switching drift diffusion models

-

Remarks on emission models

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

Other (including more complex) emission models are possible:

- Student-t, etc… (continuous observations ***x***)
- Categorical emissions, Poisson emissions etc (discrete observations ***x***)

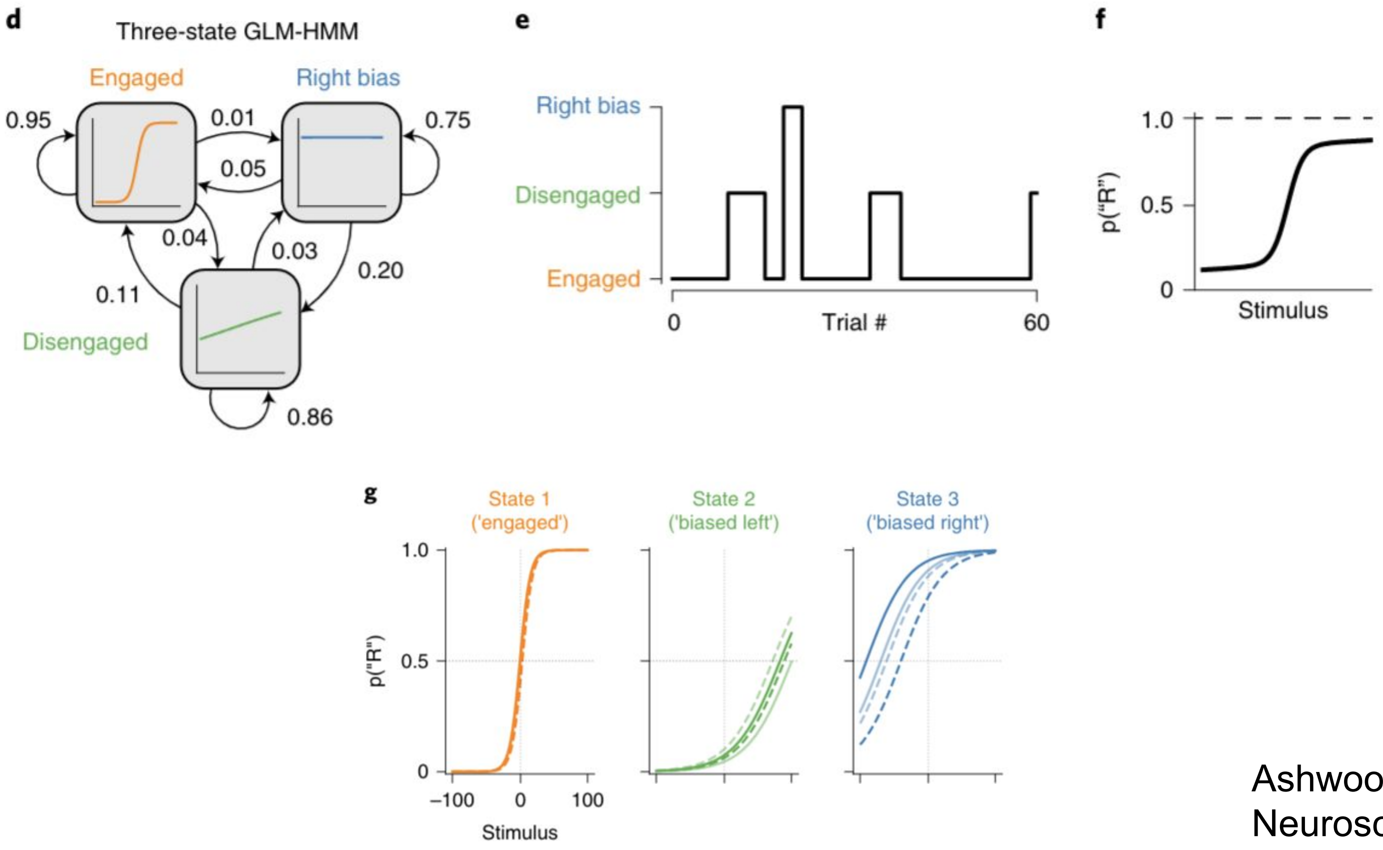
- Linear models of input ***u***: $p(\boldsymbol{x}|z_k = 1, \theta_k) = \mathcal{N}(\boldsymbol{x}|\boldsymbol{W}_k\boldsymbol{u} + c_k, \Sigma_k)$

- Autoregressive models: $p(\boldsymbol{x}_n|z_k = 1, \theta_k) = \mathcal{N}(\boldsymbol{x}_n|\boldsymbol{A}_k\boldsymbol{x}_{n-1} + d_k, \Sigma_k)$

- Combine them, you get switching drift diffusion models

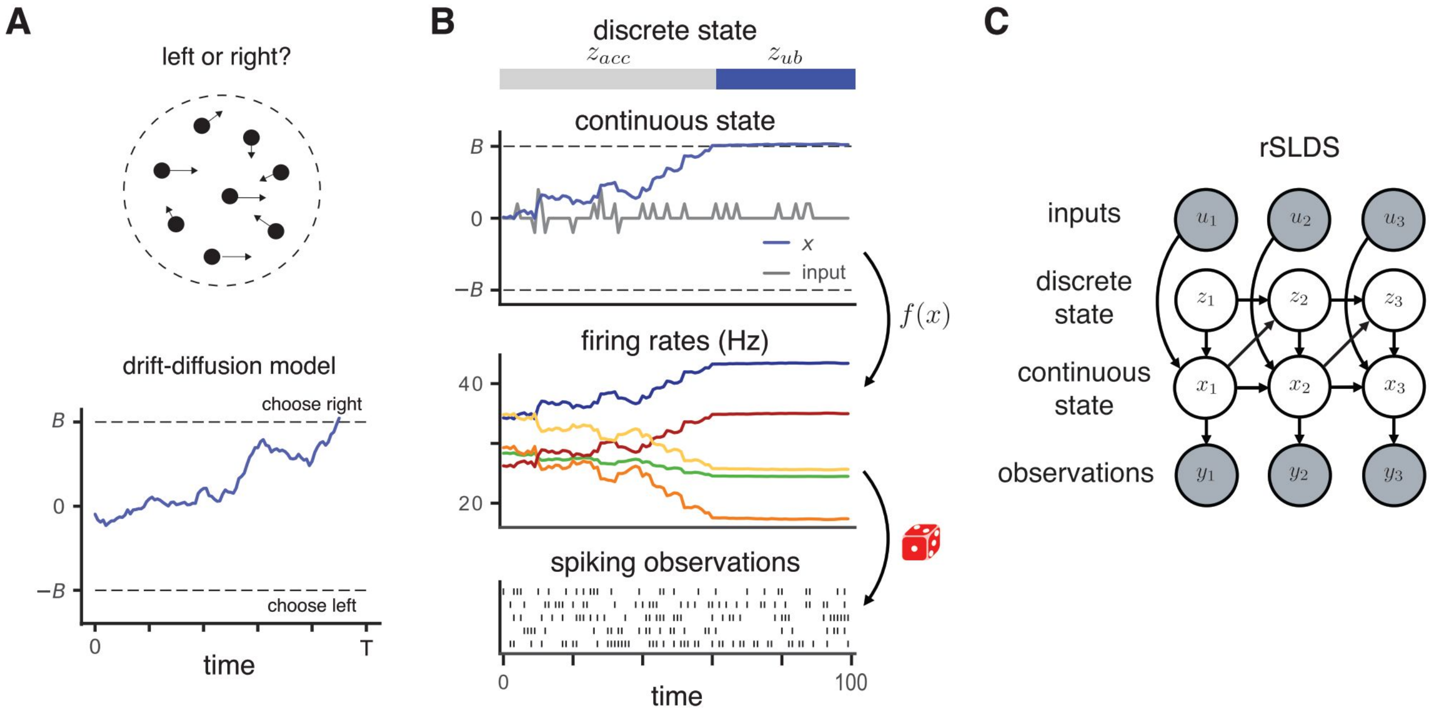
- Switching factor analysis (if you include an extra set of continuous latents that depend on the state)

Examples of HMMs in neuroscience



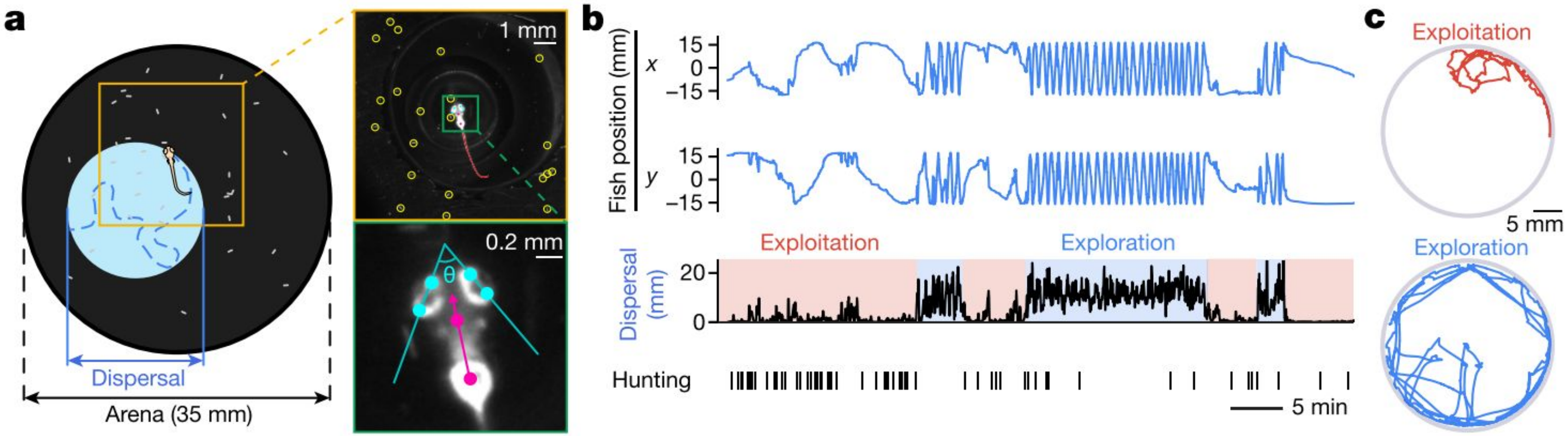
Ashwood et al., Nat Neurosci, 2022

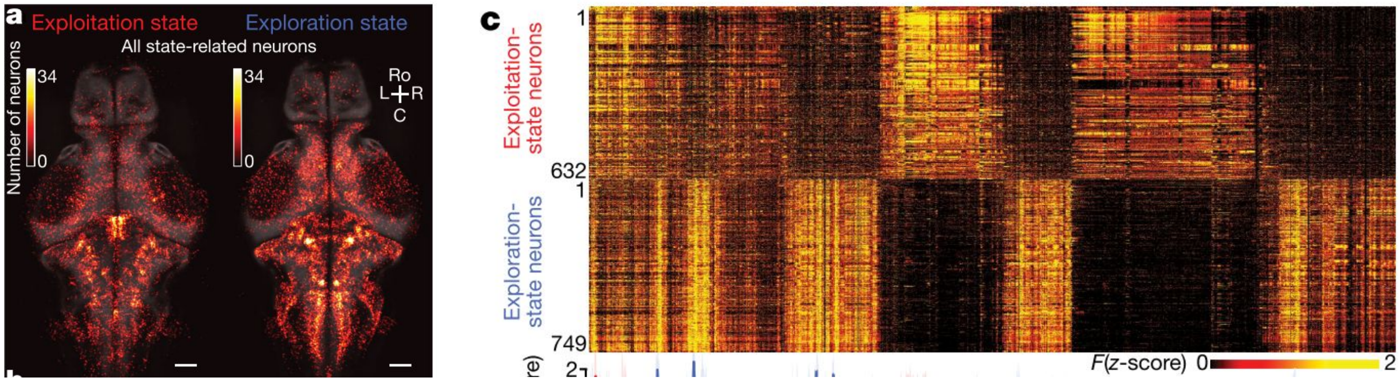
Examples of HMMs in neuroscience



Zoltowski et al., ICML, 2020

Examples of HMMs in neuroscience

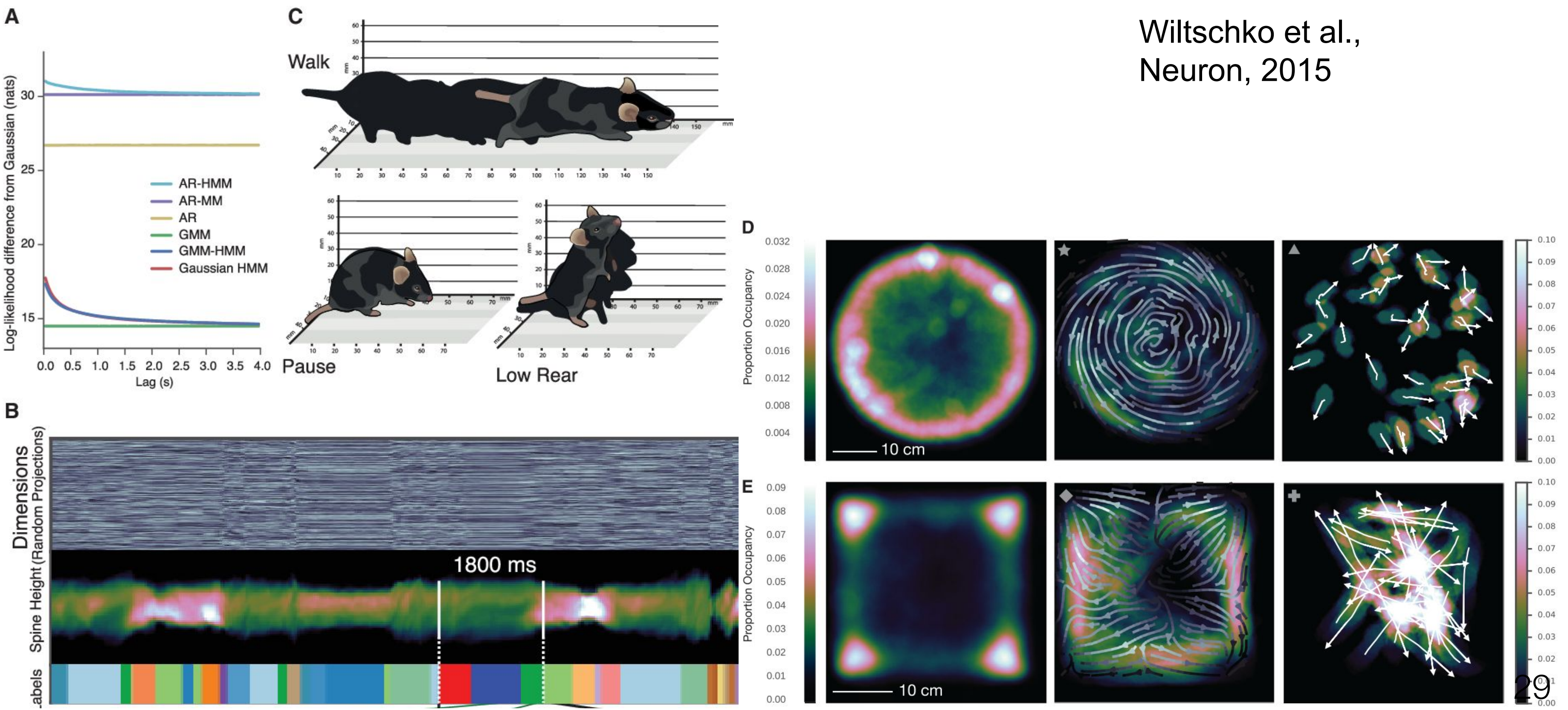




Marques et al., Nature, 2019

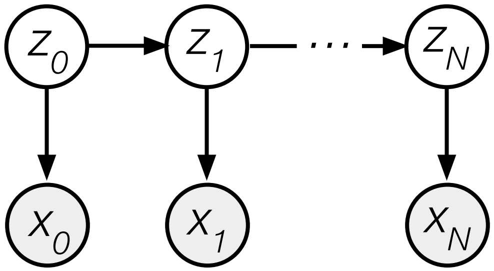
Examples of HMMs in neuroscience

Wiltschko et al., Neuron, 2015



Kalman filter

The generative model is the same as for the HMM

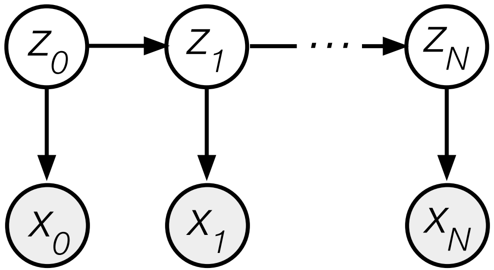


Joint distribution:

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

Kalman filter

The generative model is the same as for the HMM



Joint distribution:

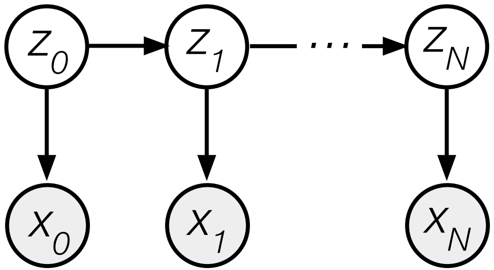
$$p(\mathbf{x}, \mathbf{z}|\theta) = p(\mathbf{z}_0|\theta) \prod_{n=1}^{N} p(\mathbf{z}_n|\mathbf{z}_{n-1}, \theta) p(\mathbf{x}_n|\mathbf{z}_n, \theta)$$

But now the latent ***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:

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

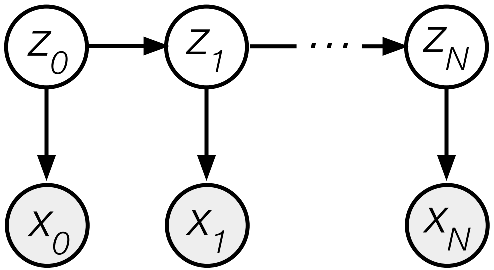
But now the latent $\boldsymbol{z}$ is continuous-valued, with transition dynamics determined by

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

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

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

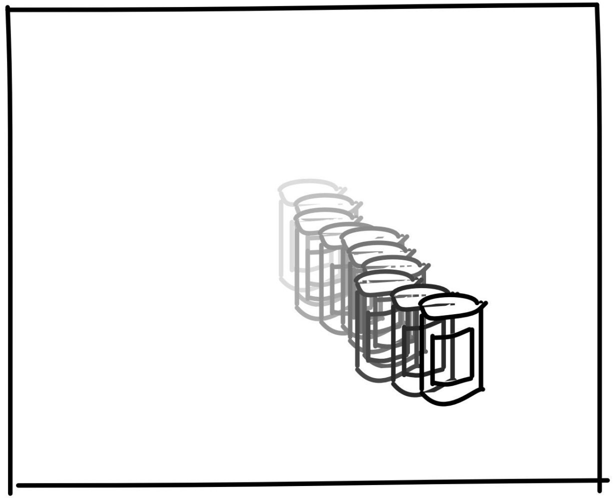
Kalman filter



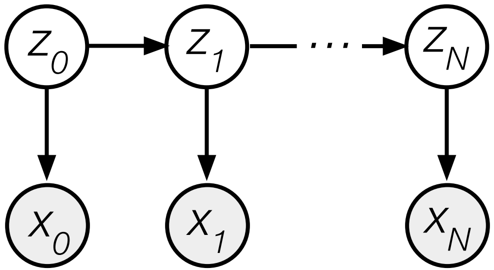
$$p(\mathbf{z}_n|\mathbf{z}_{n-1},\theta) = \mathcal{N}(\mathbf{z}_n;\mathbf{A}\mathbf{z}_{n-1},\mathbf{Q})$$

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

Dynamics of the latent ***z***



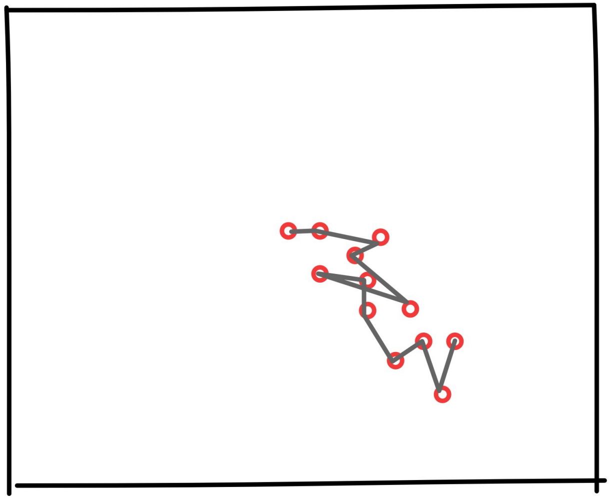
Kalman filter



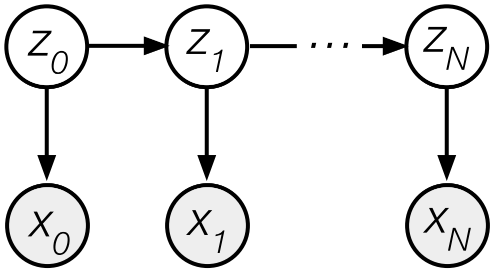
$$p(\mathbf{z}_n | \mathbf{z}_{n-1}, \theta) = \mathcal{N}(\mathbf{z}_n; \mathbf{A}\mathbf{z}_{n-1}, \mathbf{Q})$$

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

Observations ***x***



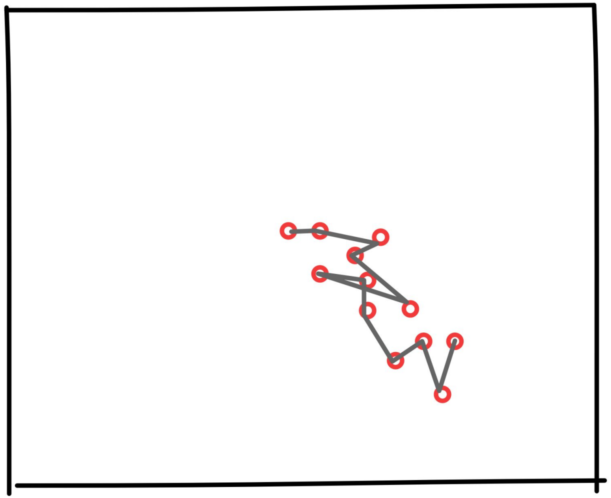
Kalman filter



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

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

Observations ***x***



→ Inference: Posterior for state ***z*** and parameters ***A, C, Q, R***

Kalman filter

Inference: Posterior for state ***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 ***A, Q, C, R*** based on complete-data log likelihood $\mathbb{E}_{\mathbf{z}}\left[\ln p(\boldsymbol{x},\boldsymbol{z}|\theta')\right]$

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