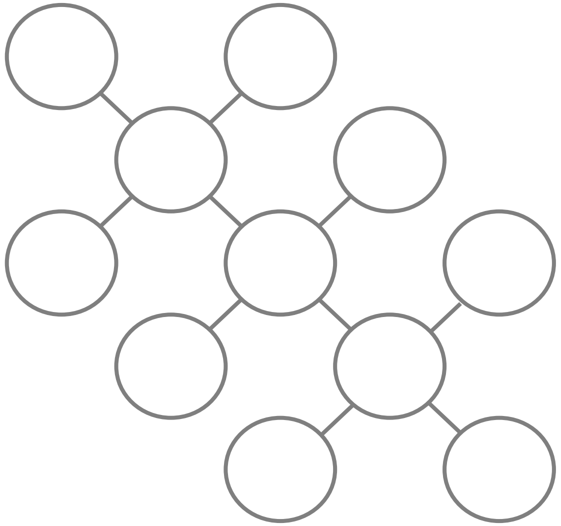


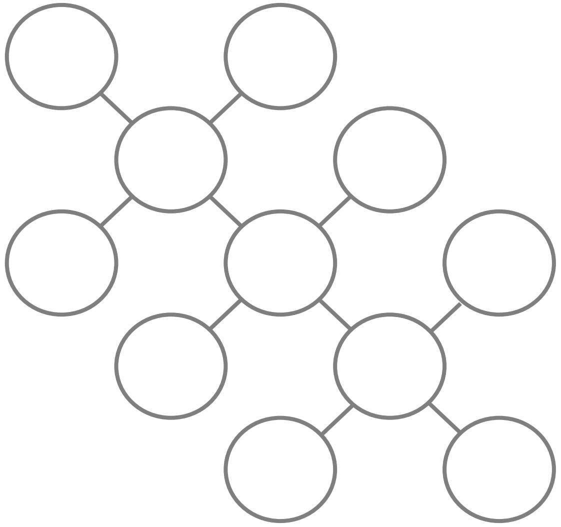
BAYESIAN INFERENCE ON A GRAPH

John Whitamore

Bayes' Mixer

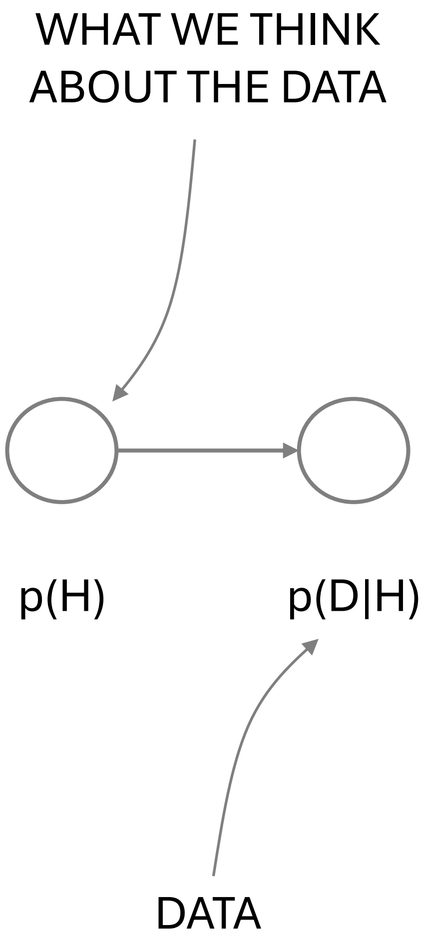
February 2025

WHAT IS THIS FOR?



DATA AND THOUGHTS

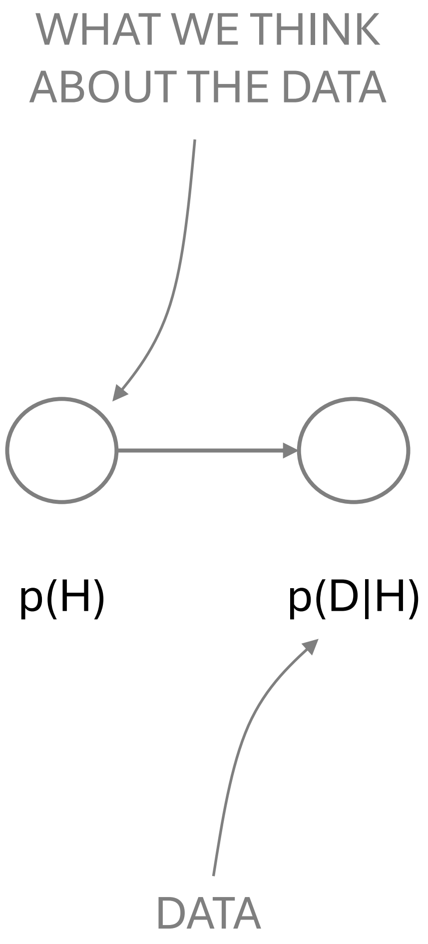
Distinguish between
- Data
- What we think about the data



INDIVIDUAL MODEL: STATIC

Model data using known root causes

Identify root causes driving data



INDIVIDUAL MODEL: DYNAMIC

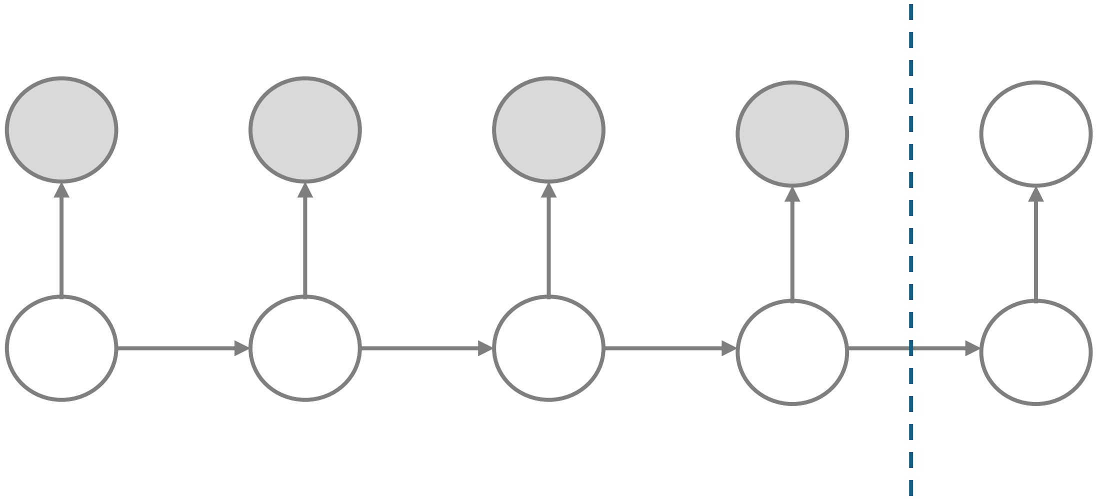
Model data using known root causes

Identify root causes driving data

Track behaviour dynamically through time

Make predictions about the future

Better understand the past



INDIVIDUAL MODEL: HIERARCHICAL

Model data using known root causes

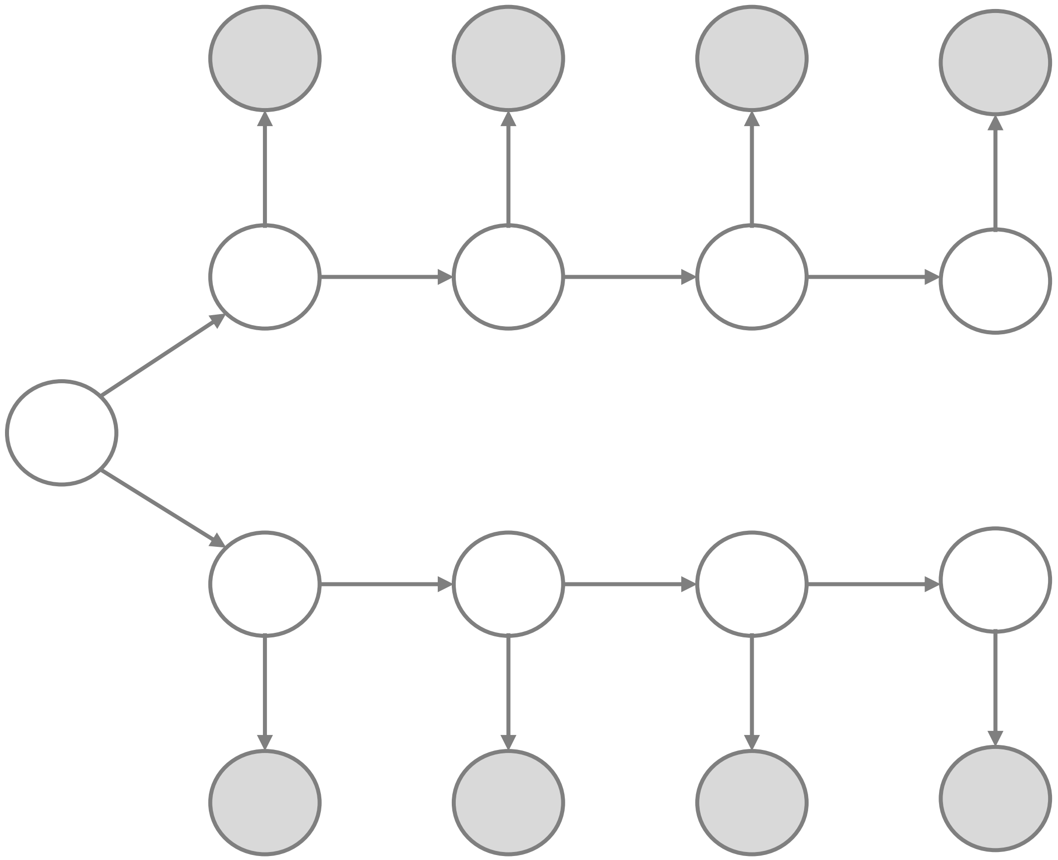
Identify root causes driving data

Track behaviour dynamically through time

Make predictions about the future

Better understand the past

Use hierarchies to share inferences



SYSTEMS OF MODELS

Model data using known root causes

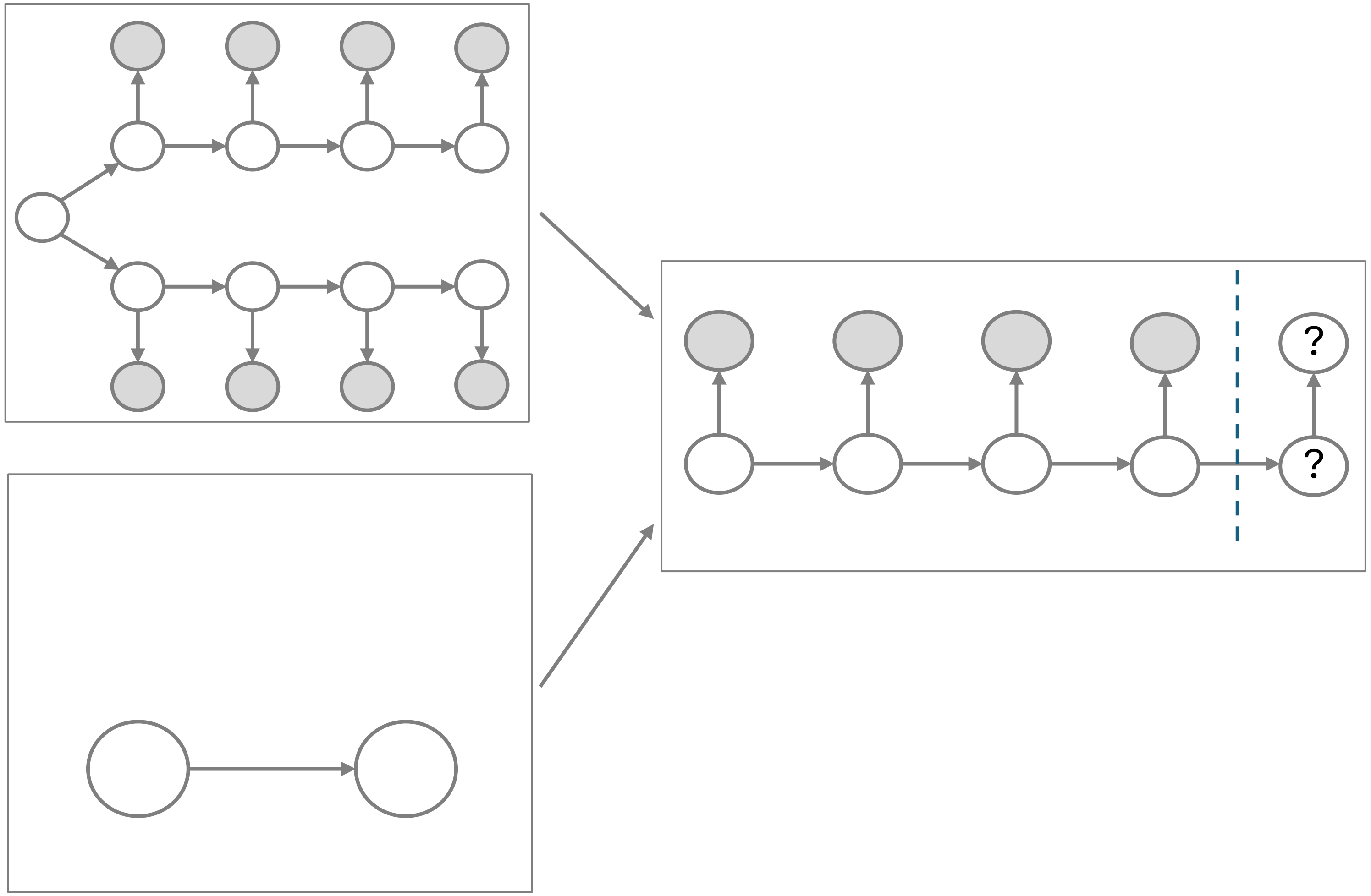
Identify root causes driving data

Track behaviour dynamically through time

Make predictions about the future

Better understand the past

Connect models together into a system



DO THINGS!

Model data using known root causes

Identify root causes driving data

Track behaviour dynamically through time

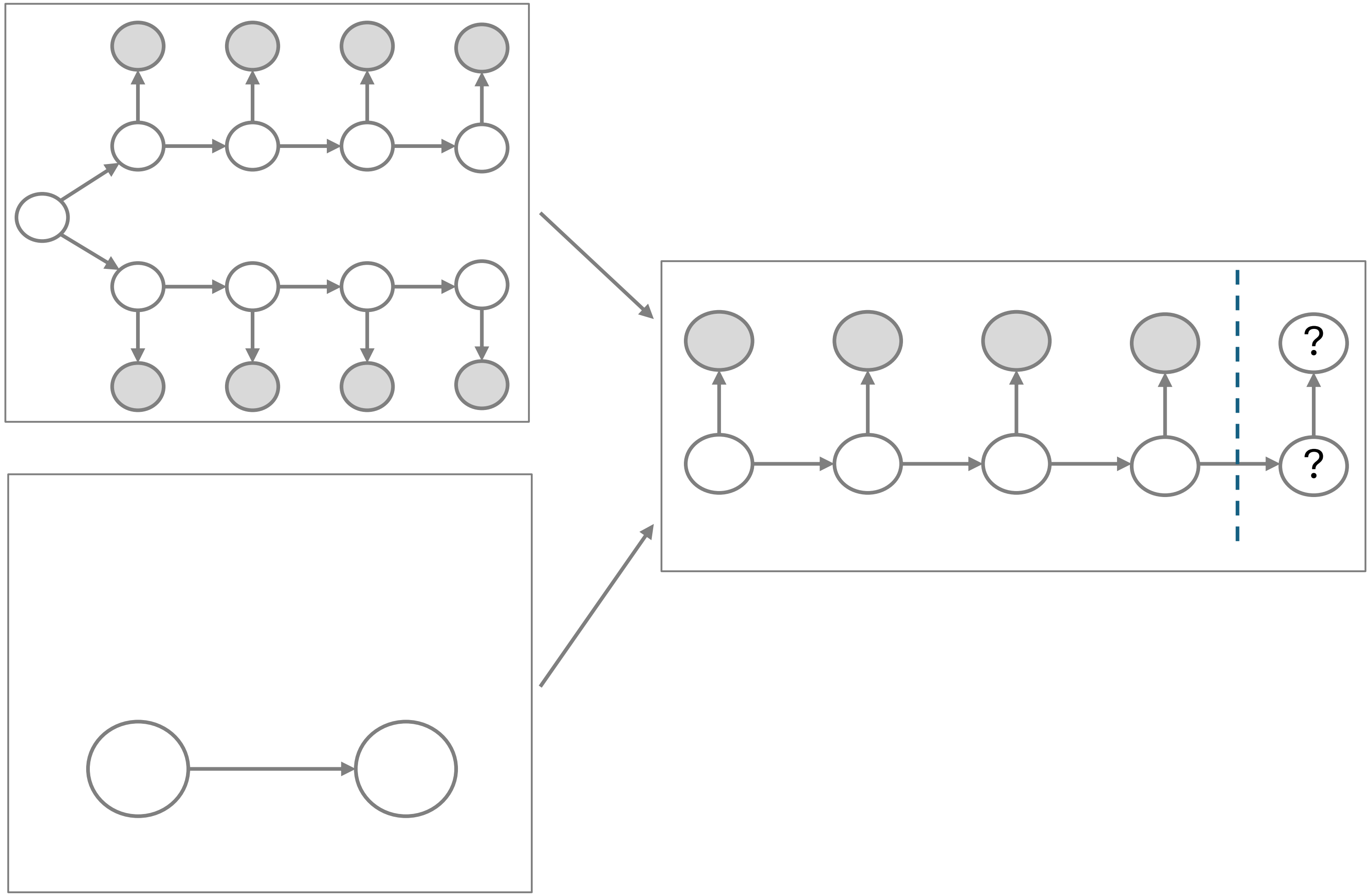
Make predictions about the future

Better understand the past

Connect models together into a system

Enable the system to act

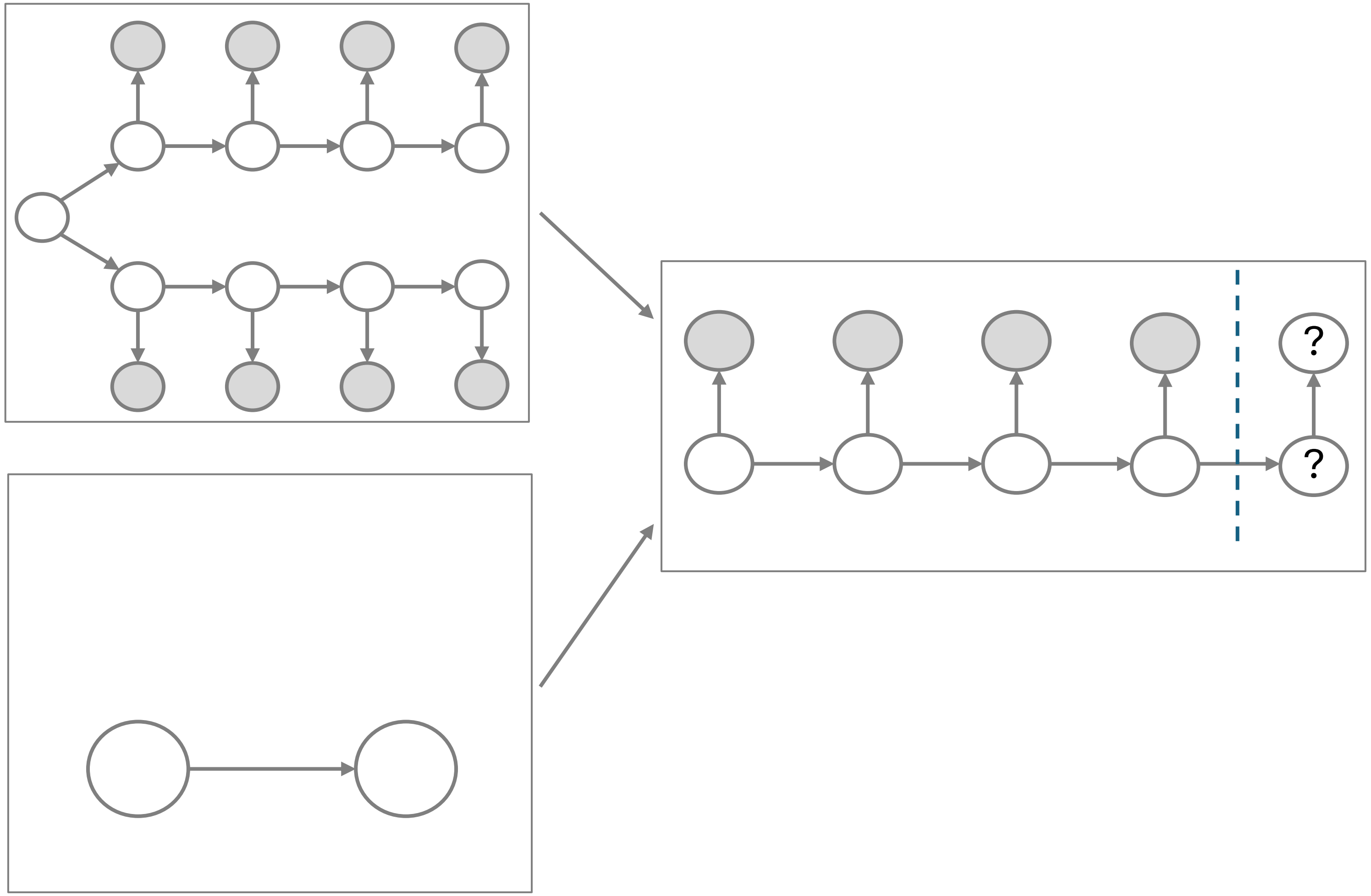
NOT IN SCOPE



APPLICATIONS

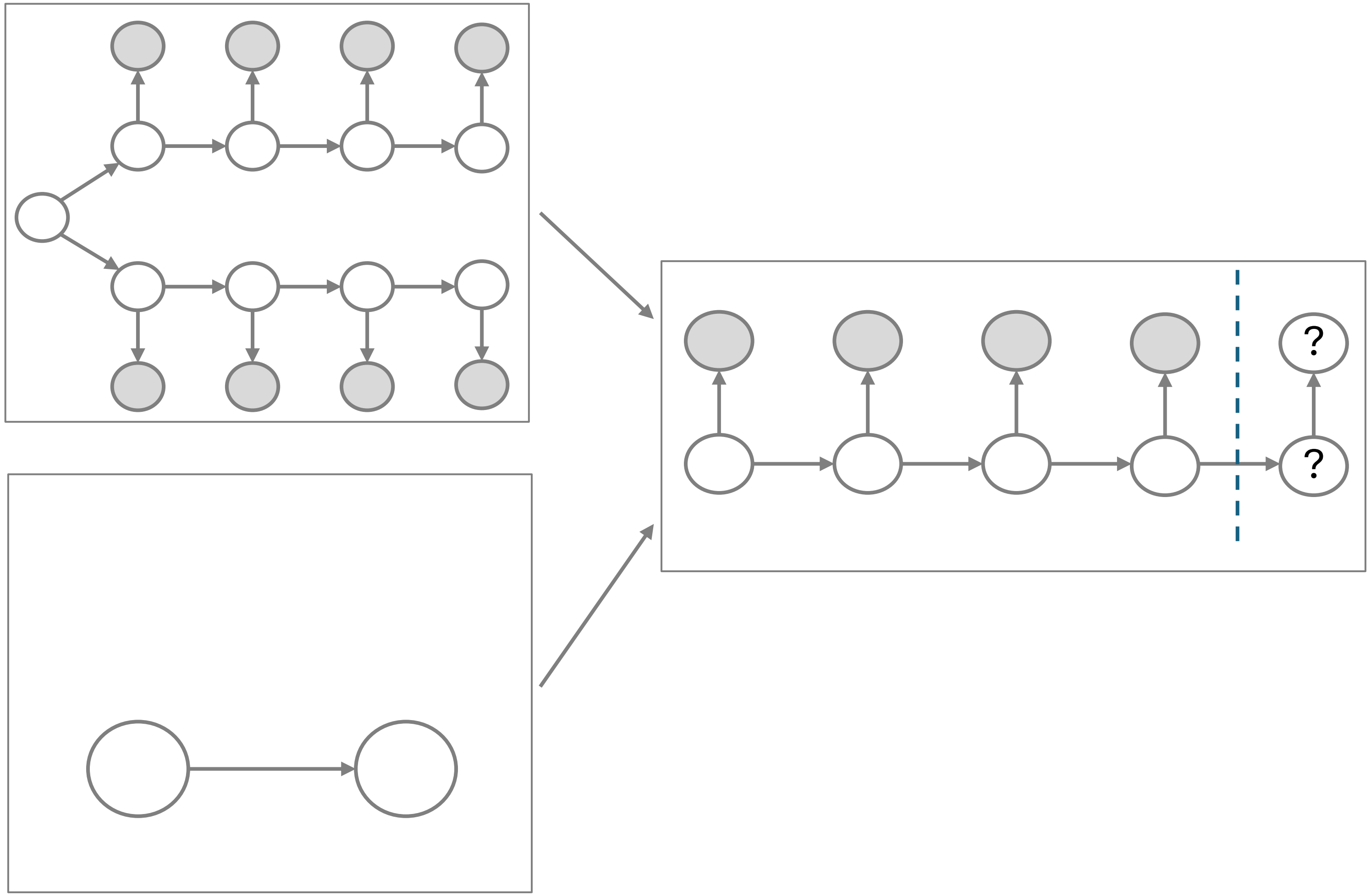
Organisations

- Commerce
- Finance
- Governments
- Defence



PARALLELS

Organisms
- Individual organisms
- Ecosystems

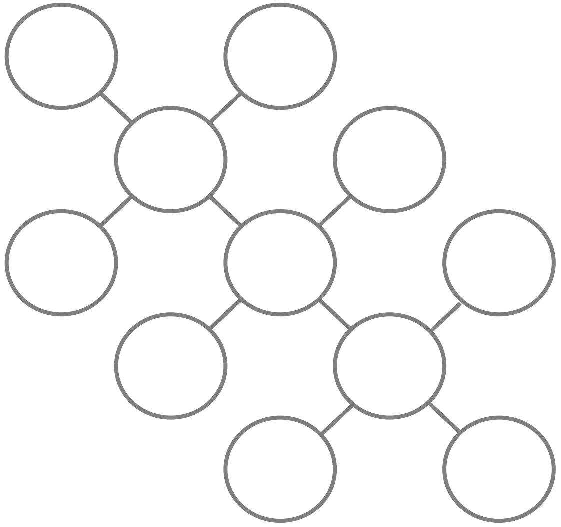


MANIFESTO

- ✓ Localised information sources
- ✓ Localised inference
- ✓ Efficient sharing of inferences
- ✓ Global consistency of inferences
- ✓ Automated calibration
- ✓ Scenario analysis
- ✓ Prediction

- ✓ Time efficiency
- ✓ Robustness
- ✓ Scalability

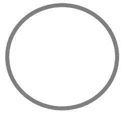
- ✓ Bayesian inferences
- ✓ Message-passing
- ✓ On a graph



CONSTRUCTING A GRAPH

A NODE

A node represents a probability distribution



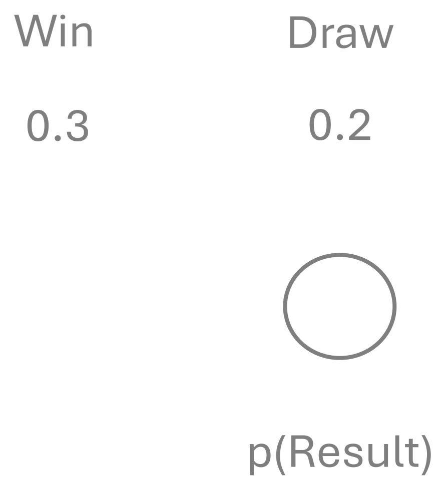
p(A)

A NODE

A node represents a probability distribution
- Probability is a non-negative real value
- Assigned to outcomes of an event

Sum rule
- add up probabilities of mutually exclusive outcomes
- Probabilities of partition sum to one

Probability distributions can be
- Parametric or non-parametric
- Discrete or continuous
- Univariate, multivariate, process



Event	Result of match
Outcome space	{Win, Draw, Lose}

A NODE

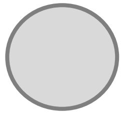
A node represents a probability distribution
- Probability is a non-negative real value

Sum rule
- add up probabilities of mutually exclusive outcomes
- Probabilities of partition sum to one

Probability distributions can be
- Parametric or non-parametric
- Discrete or continuous
- Univariate, multivariate, process

A node can represent an observed value
- p(Observed value) = 1
- Node is “clamped”
- Represent by shading in the node

A win!



p(Result = Win) = 1

AN EDGE

An edge represents conditioning

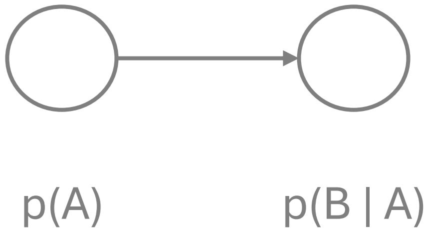
The table represents joint probabilities
- p(A, B)
- p(Home, Win) = 0.2

Marginal probabilities
- p(Win) = p(Win, Home) + p(Win, Away) = 0.3

Conditional probabilities
- p(B | A) = p(B, A) / p(A)
- P(Win | Home) = p(Win, Home) / p(Home)
  = 0.2 / 0.5
  = 0.4

B

A	WIN	DRAW	LOSE	
HOME	0.2	0.1	0.2	0.5
AWAY	0.1	0.1	0.3	0.5
	0.3	0.2	0.5	



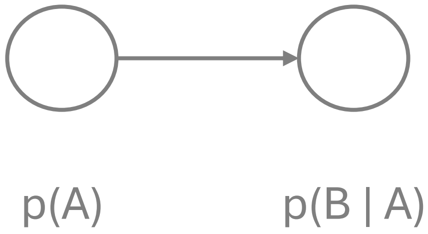
A GRAPH

Conditional probability $p(B|A) = p(B, A) / p(A)$

Rearrange $p(B, A) = p(A) \times p(B|A)$

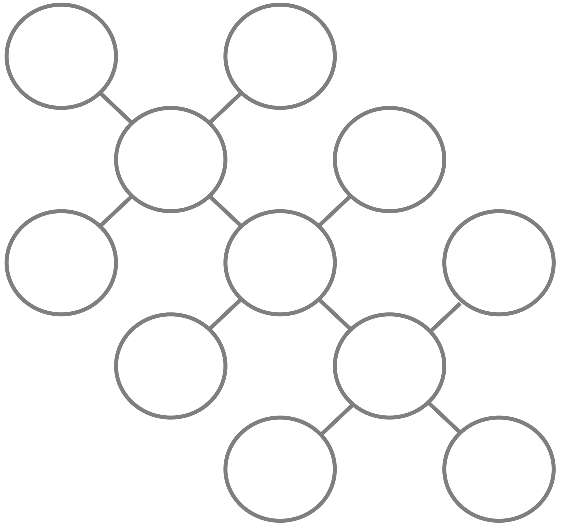
Product Law
- Multiply conditionally independent distributions
- Independence $p(B, A) = p(A) \times p(B)$

The graph
- Represents the joint distribution
- In factorised form
- Exploiting conditional independence relationships



Manage complexity
- Factorise the problem into many small problems

CORE IDEA!



FAMILIAR MODELS IN GRAPH FORM

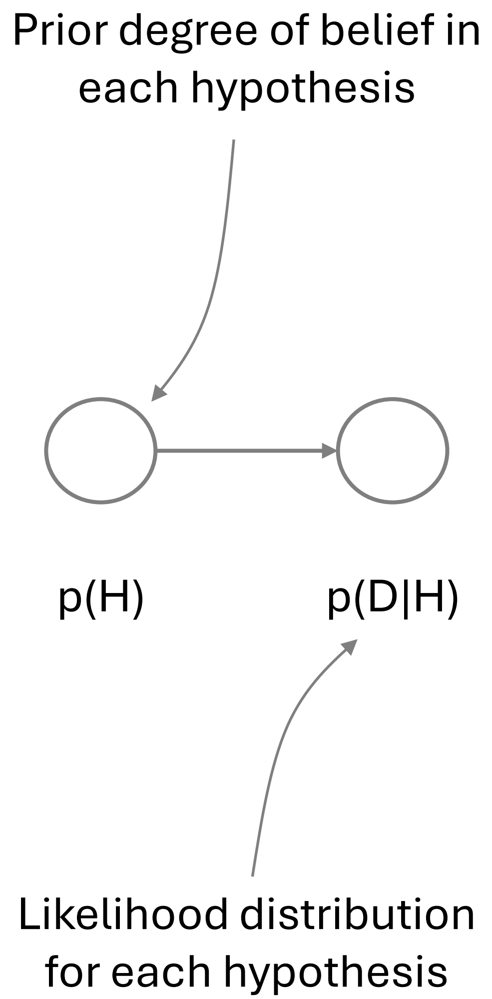
TRUST MODEL

A hypothesis H is an idea or potential explanation.

I want to know if I should trust a particular person. My hypotheses are YES or NO.

I use Bayes' Law to update my degrees of belief in those hypotheses according to how I observe that person behaving.

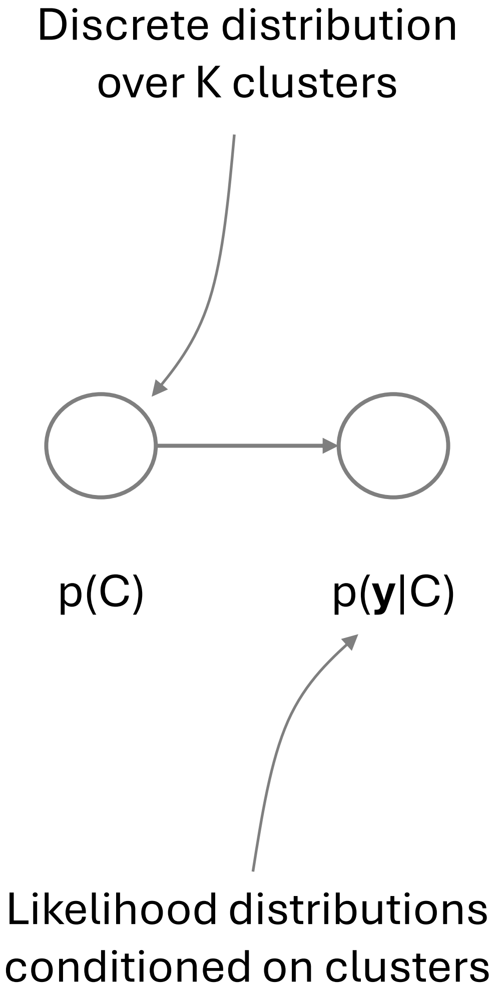
Prior distribution: $p(H)$
Likelihood distribution: $p(D \mid H)$
Posterior distribution: $p(H \mid D)$



MIXTURE MODEL

Discrete mixture of Gaussians, Poissons, …

$P(C, \mathbf{y}) = \mathrm{Disc}(C_1, \ldots C_K)\ \mathrm{Normal}(\mathbf{y}_n | \mathrm{Mean}_k, \mathrm{Cov}_k)$

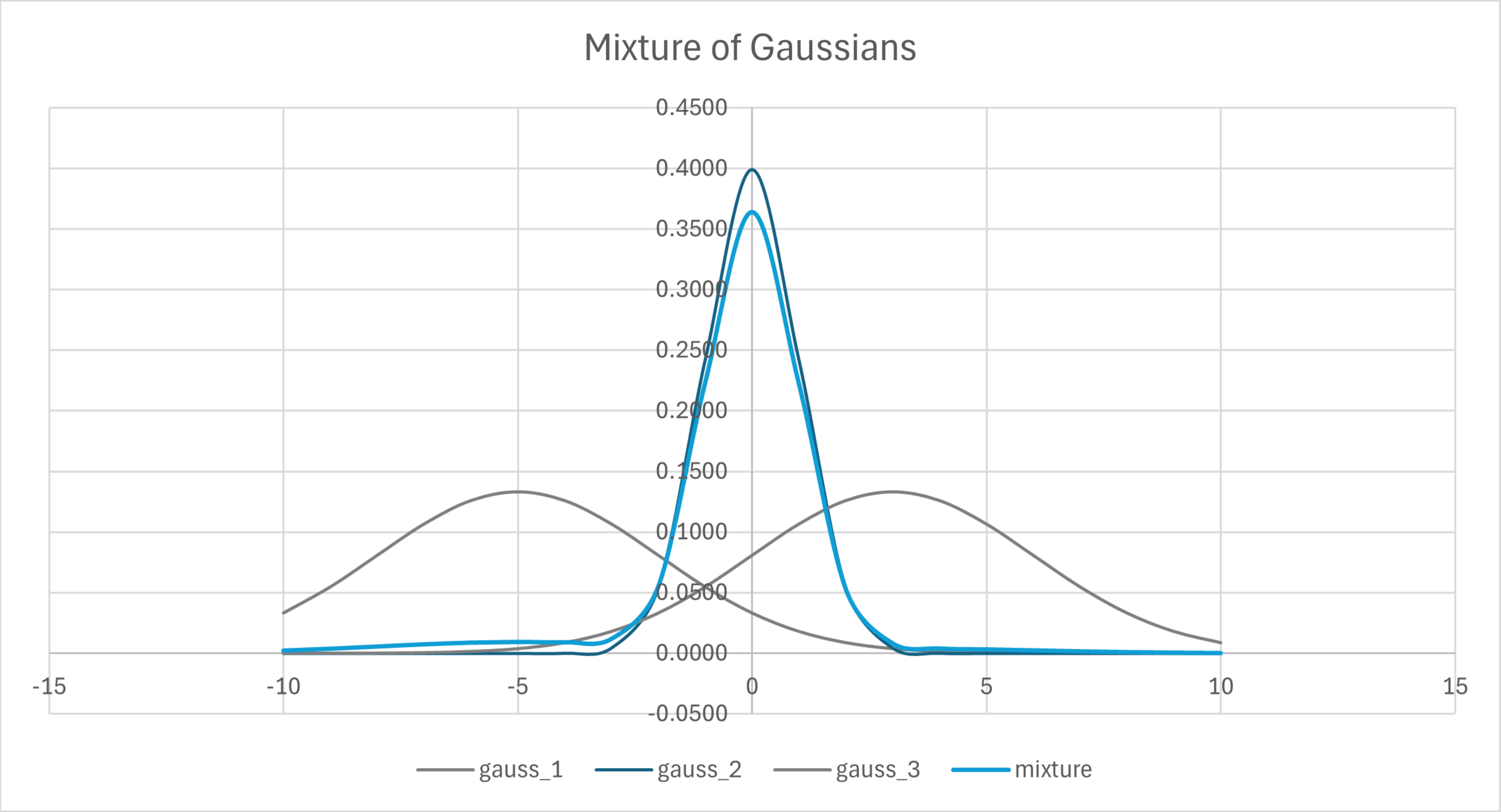


Remarks
1. A cluster is a hypothesis about the data
2. A prior is a mixing weight (and vice versa)
3. MoG is like a Taylor series for distributions
4. Continuous mixtures also useful e.g. Gamma mixture of Gaussians is (a type of) Student T
5. Stochastic Volatility models (Heston etc)

MIXTURE MODEL

Discrete mi

P(C, **y**) = Dis



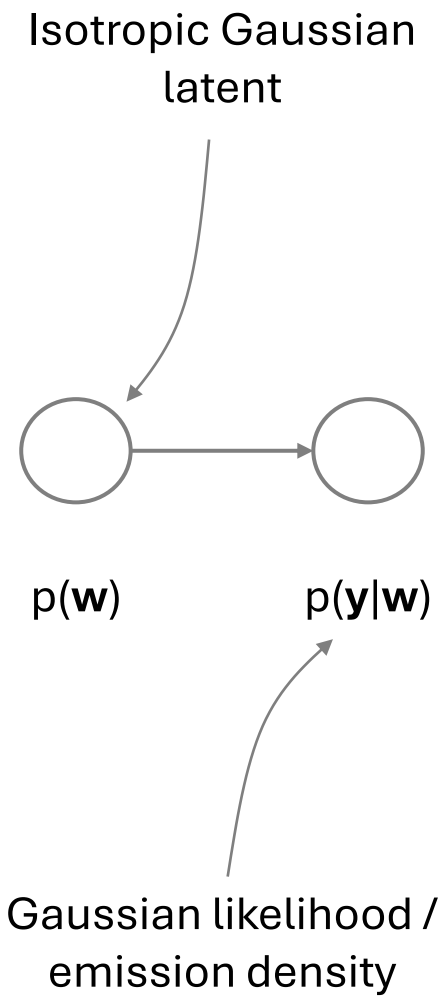
BAYESIAN REGRESSION

Supervised learning

P(**w**, **y**) = Normal(**w**|0, alpha x I) Normal(**y**|X**w**, Variance)

X is the (fixed) design matrix

- Each column represents a hypothesis
- Hypotheses populated using basis functions

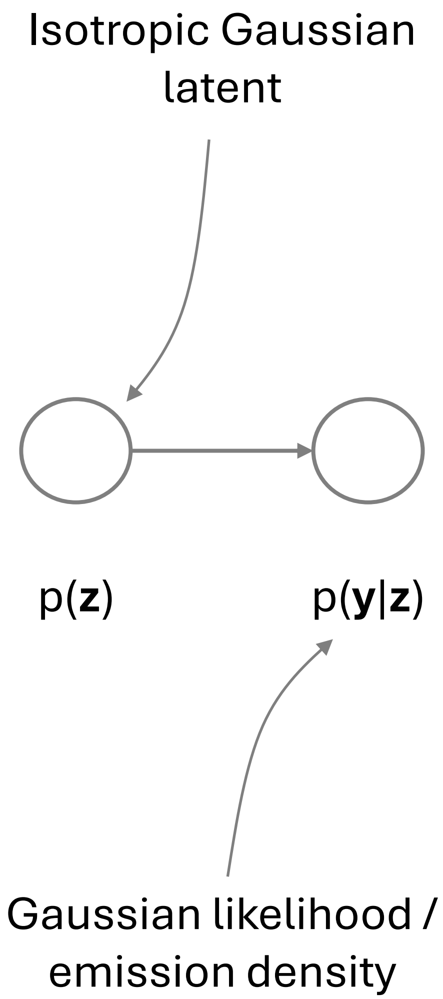


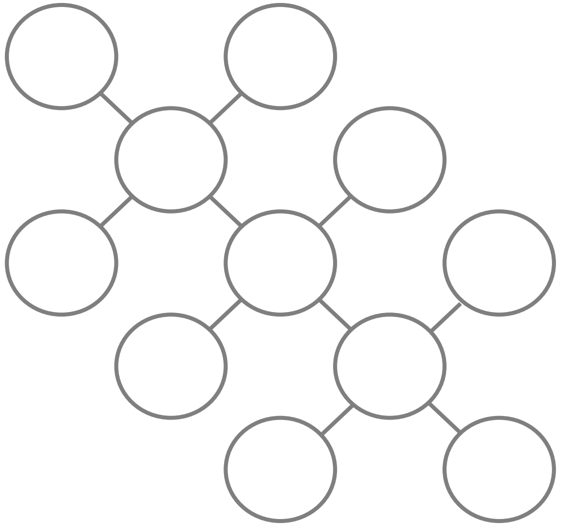
FACTOR ANALYSIS

Unsupervised learning

P(**z**, **x**) = Normal(**z**|0, I) Normal(**y**|A**z** + **m**, Cov)

Factor analysis is a regression model with
- Emission matrix A instead of Design Matrix X
- Offset vector **m**
- Both learned from the data

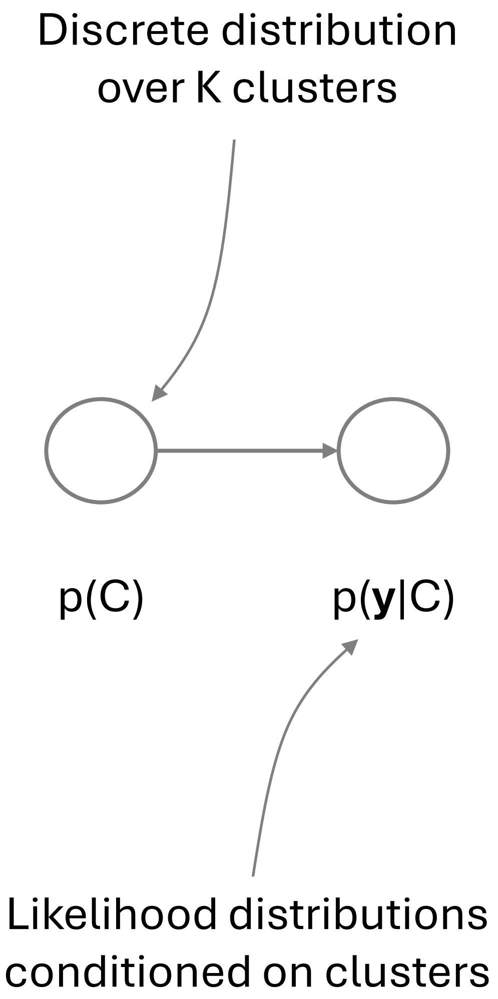




GENERATIVE MODELLING

GENERATIVE MODELLING

Generate synthetic data by sampling from graph

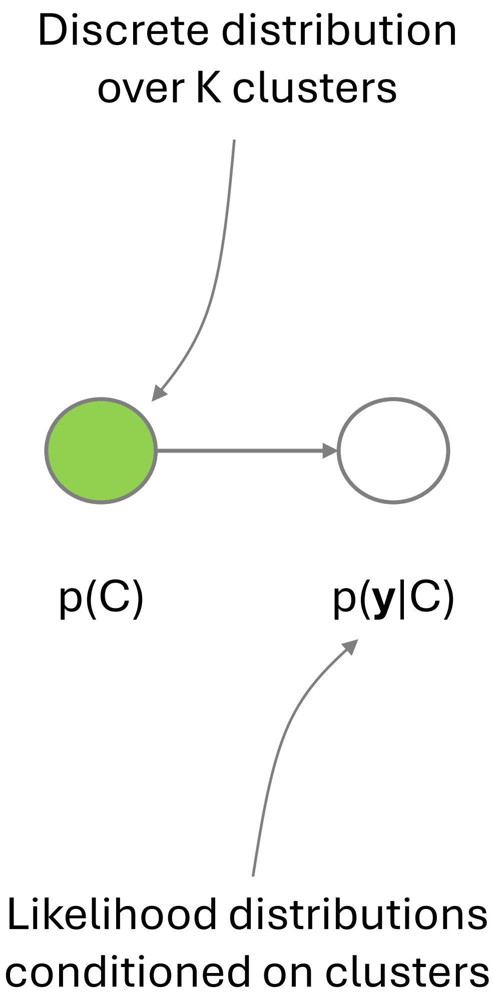


GENERATIVE MODELLING

Generate synthetic data by sampling from graph

1. Sample from prior

➢ Randomly selects a cluster

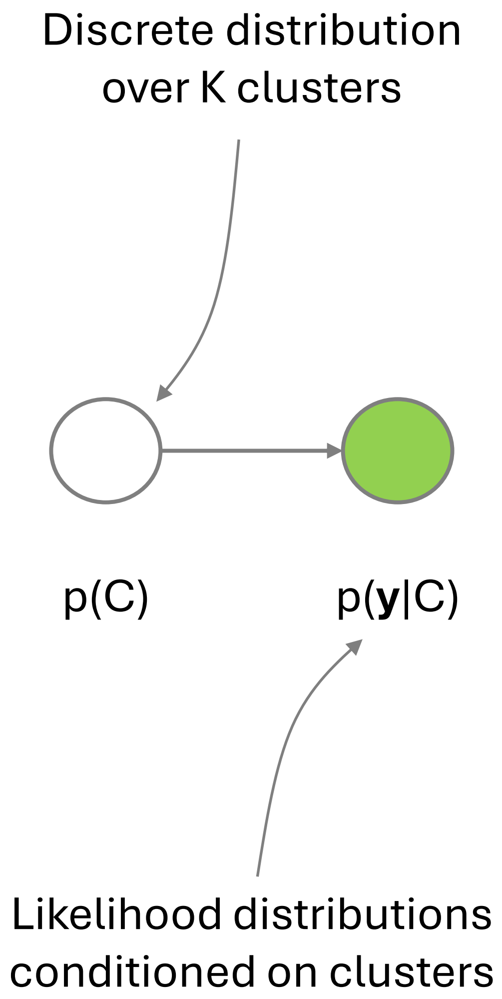


GENERATIVE MODELLING

Generate synthetic data by sampling from graph

1. Sample from prior
2. Use sample from prior to sample from likelihood

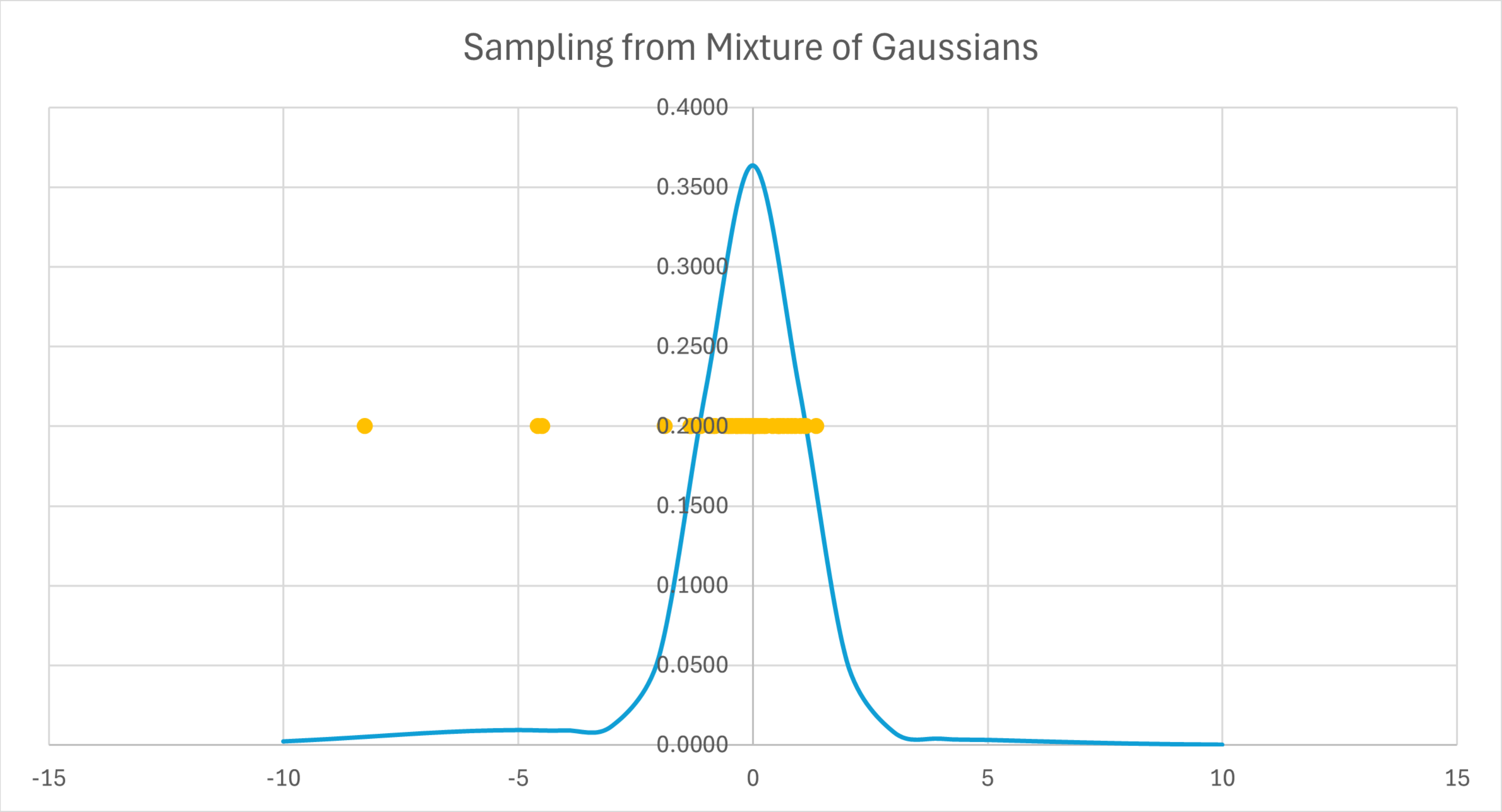
➢ Sample from Gaussian from the selected cluster



GENERATIVE MODELLING

Generate sy

1. Sample
2. Use sam



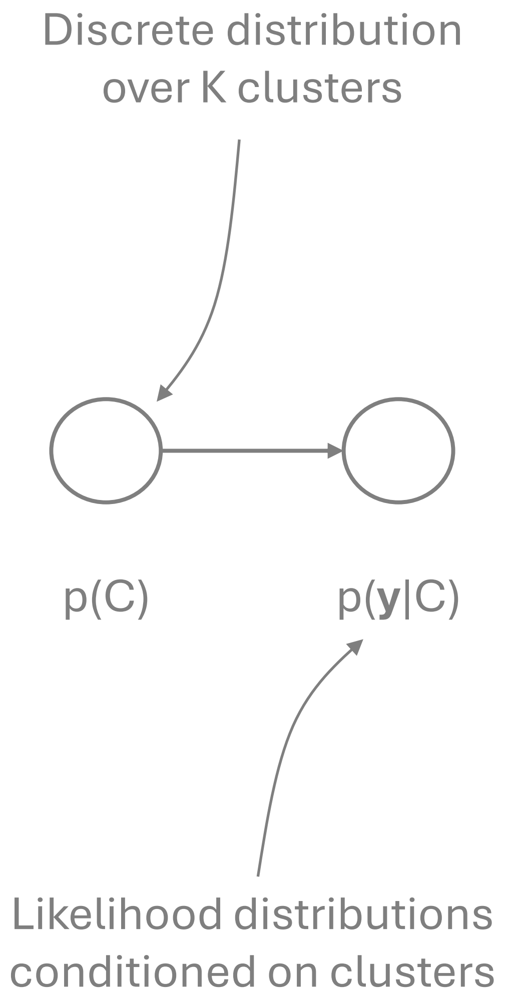
GENERATIVE MODELLING

Generate synthetic data by sampling from graph

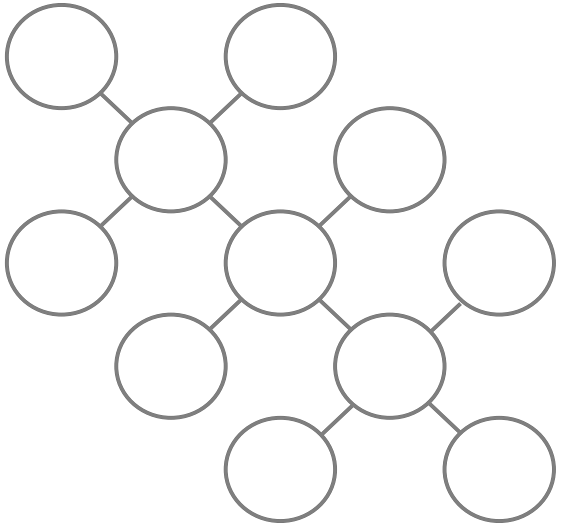
1. Sample from prior
2. Use sample from prior to sample from likelihood

In general
1. Sample from parents
2. Pass samples to children
3. Continue through the graph

Ancestral Sampling



MARKOV CHAIN

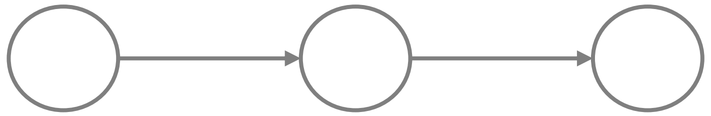


MESSAGE PASSING

Markov chain: $p(A, B, C) = p(C|B)p(B|A)p(A)$

C is conditionally independent of A, given B

- Modelling decision
- Other factorisations are available



p(A) p(B|A) p(C|B)

MESSAGE PASSING

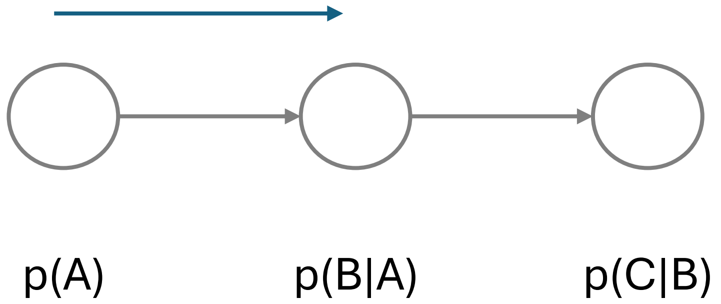
Markov chain: $p(A, B, C) = p(C|B)p(B|A)p(A)$

C is conditionally independent of A, given B

Message passing

$$
\begin{aligned}
p(C) &= \int\int p(A, B, C)\, dA\, dB \\
&= \int\int p(C|B)p(B|A)p(A)\, dA\, dB
\end{aligned}
$$

Pass p(A) forwards



MESSAGE PASSING

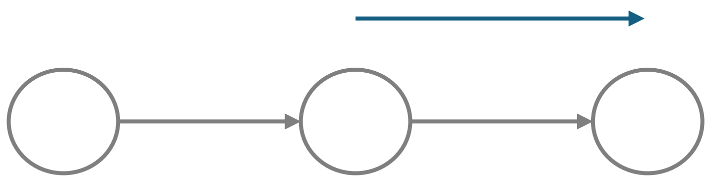
Markov chain: $p(A, B, C) = p(C|B)p(B|A)p(A)$

C is conditionally independent of A, given B

Message passing

$$
\begin{aligned}
p(C) &= \int\int p(A, B, C)\, dA\, dB \\
&= \int\int p(C|B)p(B|A)p(A)\, dA\, dB \\
&= \int p(C|B) \left[\int p(B|A)p(A)\, dA \right] dB
\end{aligned}
$$

Marginalise
Pass p(B) forwards



p(A) p(B|A) p(C|B)

MESSAGE PASSING

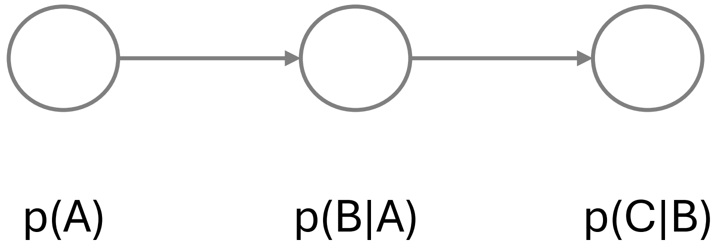
Markov chain: $p(A, B, C) = p(C|B)p(B|A)p(A)$

C is conditionally independent of A, given B

Message passing

$$
\begin{aligned}
p(C) &= \int\int p(A, B, C)\, dA\, dB \\
&= \int\int p(C|B)p(B|A)p(A)\, dA\, dB \\
&= \int p(C|B) \left[\int p(B|A)p(A)\, dA\right] dB \\
&= \int p(C|B)p(B)\, dB \\
&= p(C)
\end{aligned}
$$

Marginalise
Obtain p(C)



MESSAGE PASSING

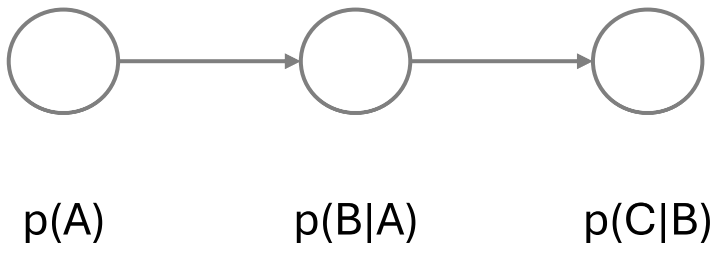
Summary

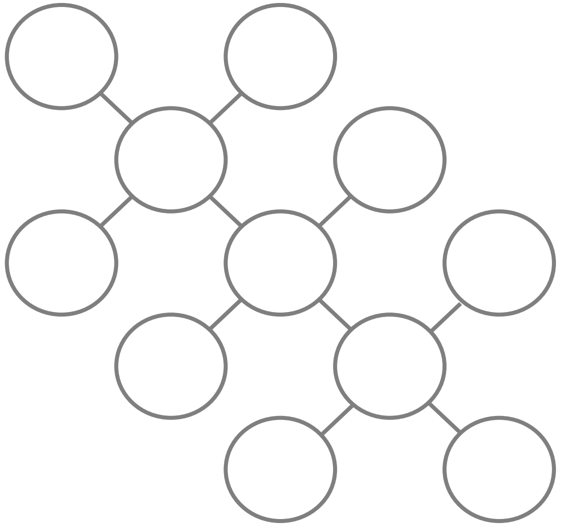
1. Summarise inferences at a node: marginalisation
2. Pass marginal distribution forwards as a message

Architecture

1. Allows inferences to be made locally
2. Allows inferences to be shared across the graph
3. Can implement as distributed architecture

Note use of message passing in Kafka, Spark, etc





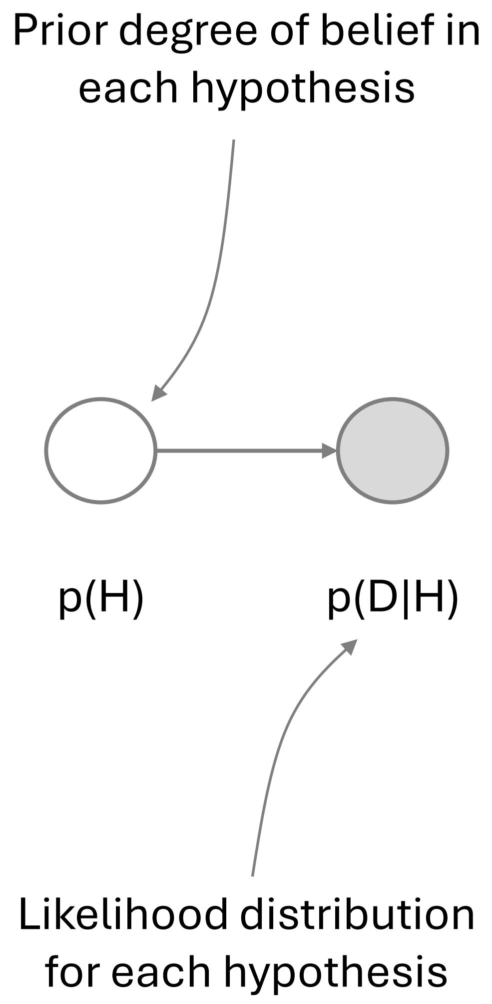
INFERENCE THROUGH TIME

TWO NODE MODEL

H Latent variable; degrees of belief in hypotheses

D Observed data

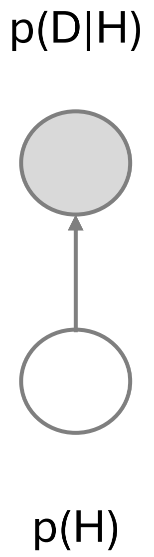
Examples
- Mixture model
- Factor analysis
- Bayesian regression
- Compound distributions



TWO NODE MODEL

H Latent variable; degrees of belief in hypotheses
D Observed data

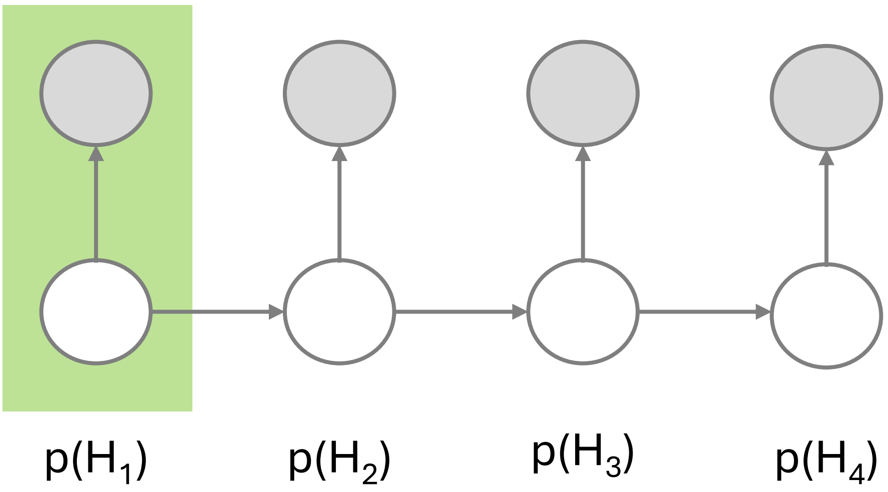
Examples
- Mixture model
- Factor analysis
- Bayesian regression
- Compound distributions



TWO NODE MODEL THROUGH TIME

H Latent variable; degrees of belief in hypotheses
D Observed data

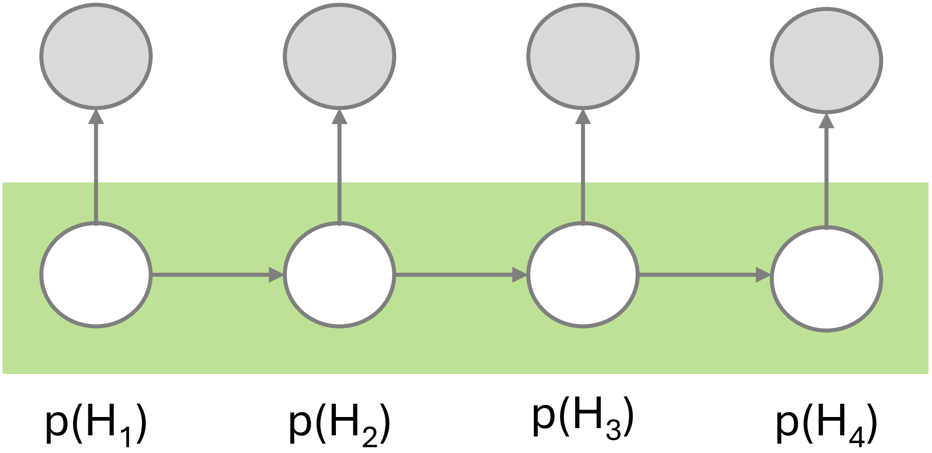
Examples
- Mixture model
- Factor analysis
- Bayesian regression
- Compound distributions



TWO NODE MODEL THROUGH TIME

H Latent variable; degrees of belief in hypotheses
D Observed data

Markov chain
- Latent variables H_n
- H discrete: Hidden Markov Model
- H continuous: filter (e.g. Kalman Filter)

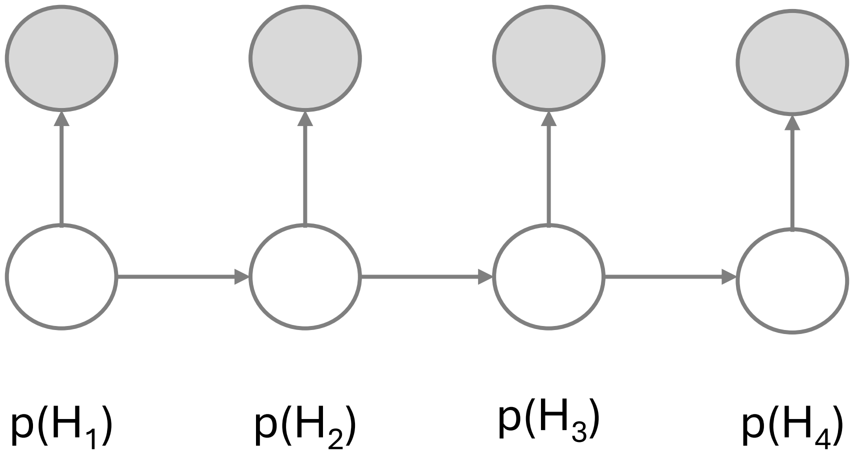


TWO NODE MODEL THROUGH TIME

H	Latent variable; degrees of belief in hypotheses
D	Observed data

Specify three densities:

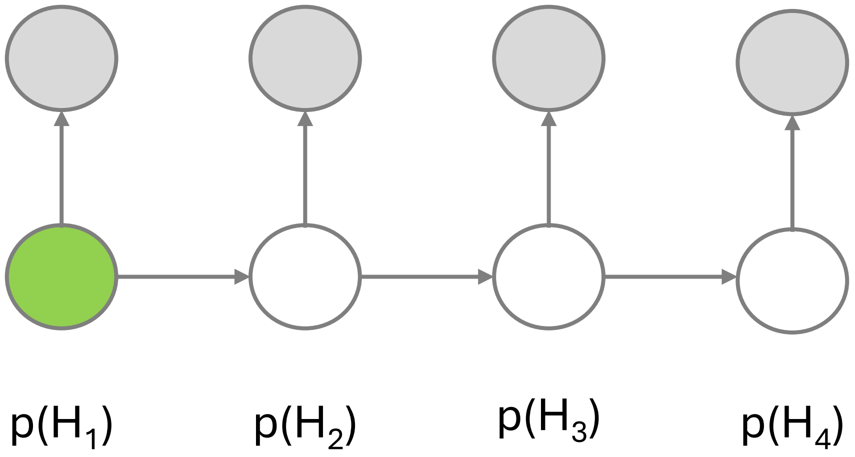
Prior	$p(H_1)$
Emission	$p(D_n \mid H_n)$
Transition	$p(H_{n+1} \mid H_n)$



FORWARD RECURSIONS

H Latent variable; degrees of belief in hypotheses
D Observed data

Initialise with prior $p(H_1)$



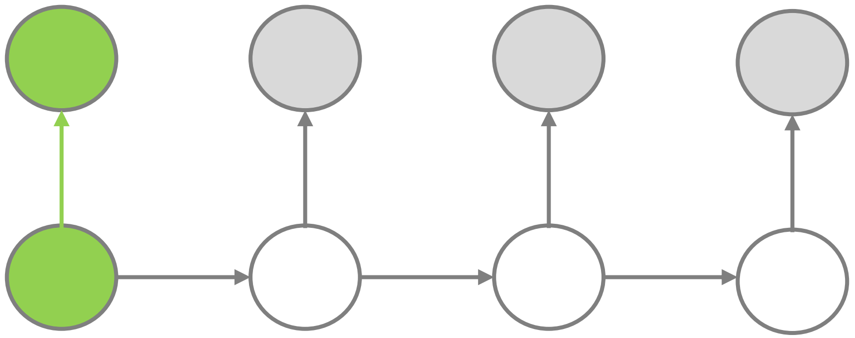
FORWARD RECURSIONS

H Latent variable; degrees of belief in hypotheses
D Observed data

Initialise with prior $p(H_1)$

Bayesian update from D_1, first data point

$$p(H_1|D_1) = p(D_1|H_1)p(H_1) / p(D_1)$$



FORWARD RECURSIONS

H	Latent variable; degrees of belief in hypotheses
D	Observed data

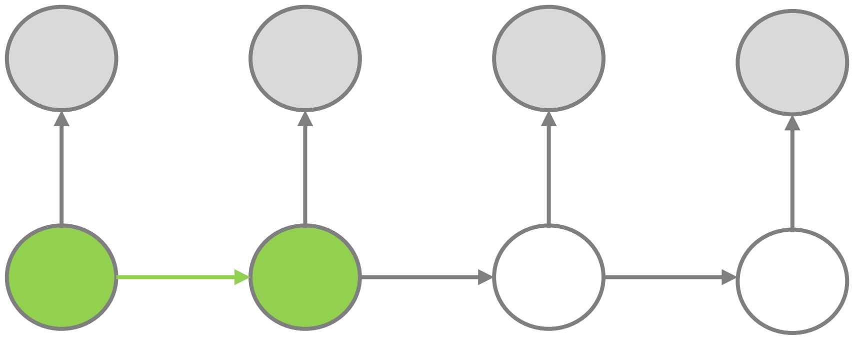
Initialise with prior $p(H_1)$

Bayesian update from D_1, first data point

$$p(H_1|D_1) = p(D_1|H_1)p(H_1) / p(D_1)$$

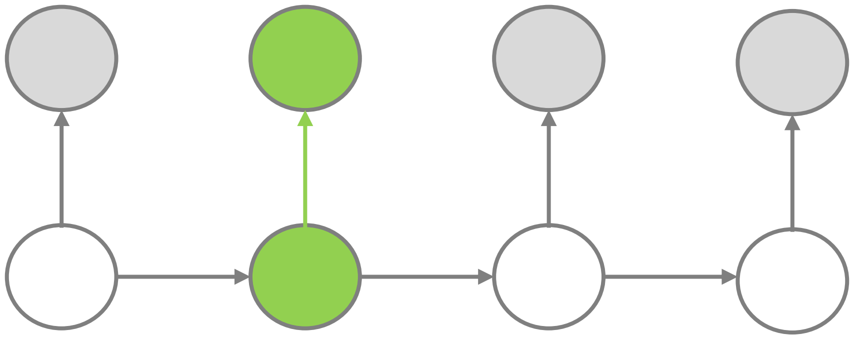
Marginalise and push forward to create prior $p(H_2|D_1)$

$$p(H_2|D_1) = \int p(H_2|H_1)p(H_1|D_1)\, dH_1$$



FORWARD RECURSIONS

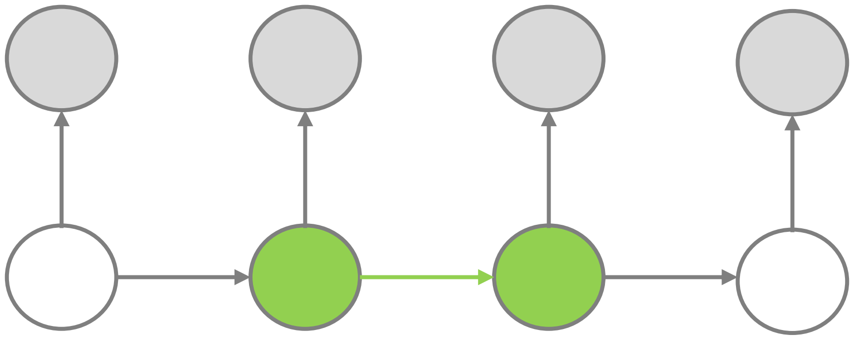
H	Latent variable; degrees of belief in hypotheses
D	Observed data



Bayesian update from data

FORWARD RECURSIONS

H Latent variable; degrees of belief in hypotheses
D Observed data

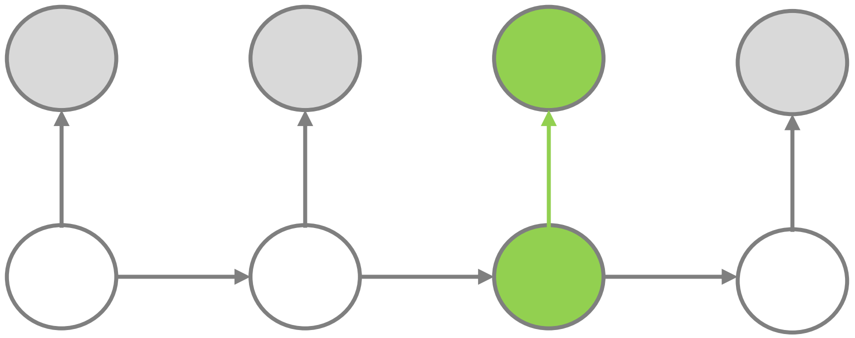


Marginalise and push forwards

FORWARD RECURSIONS

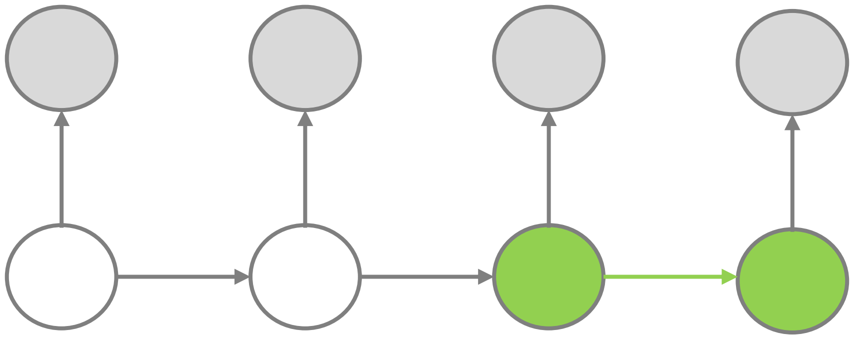
H Latent variable; degrees of belief in hypotheses
D Observed data

Bayesian update from data



FORWARD RECURSIONS

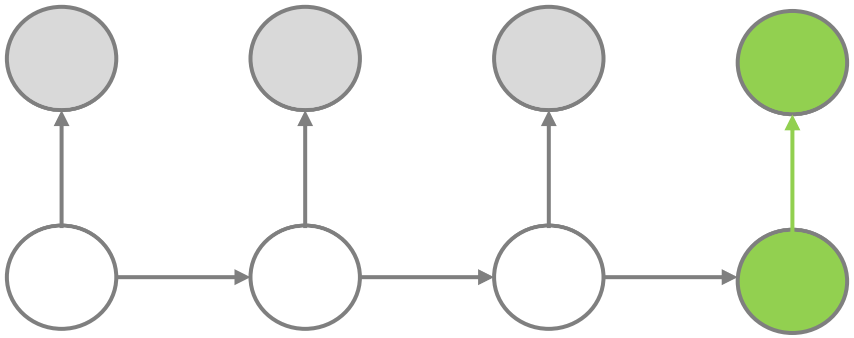
H Latent variable; degrees of belief in hypotheses
D Observed data



Marginalise and push forwards

FORWARD RECURSIONS

H Latent variable; degrees of belief in hypotheses
D Observed data



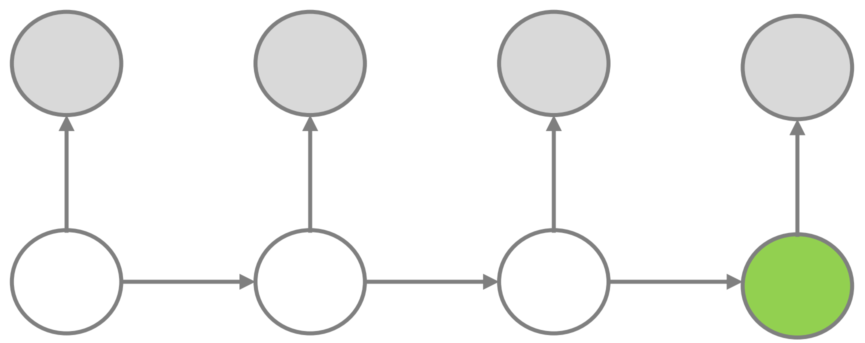
Bayesian update from data

FORWARD RECURSIONS

Have updated final latent variable with all data

The FILTERING problem
- Estimate current state

Many such models
- Aeronautics
- Algorithmic trading

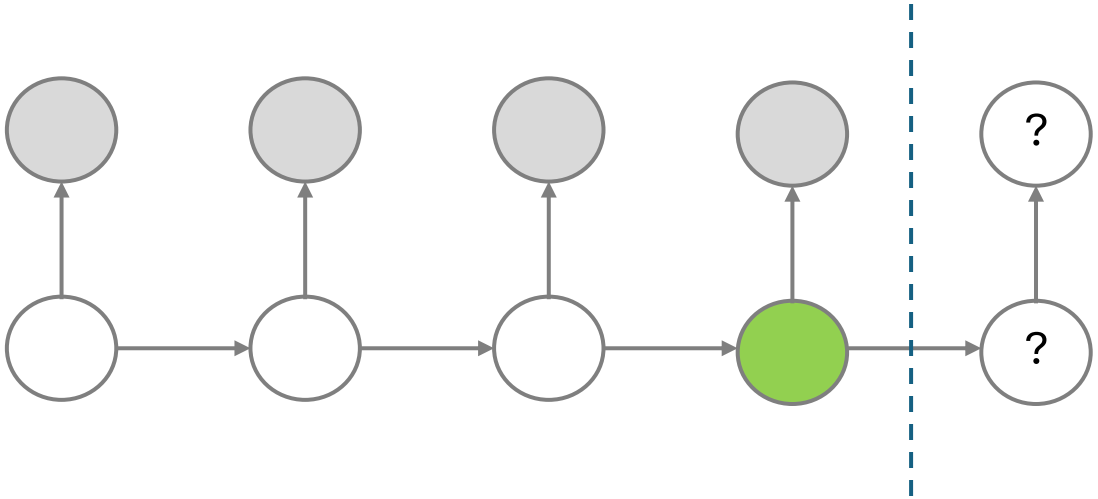


PREDICTION

Have updated final latent variable with all data

The FILTERING problem
- Estimate current state

The PREDICTION problem



PREDICTION

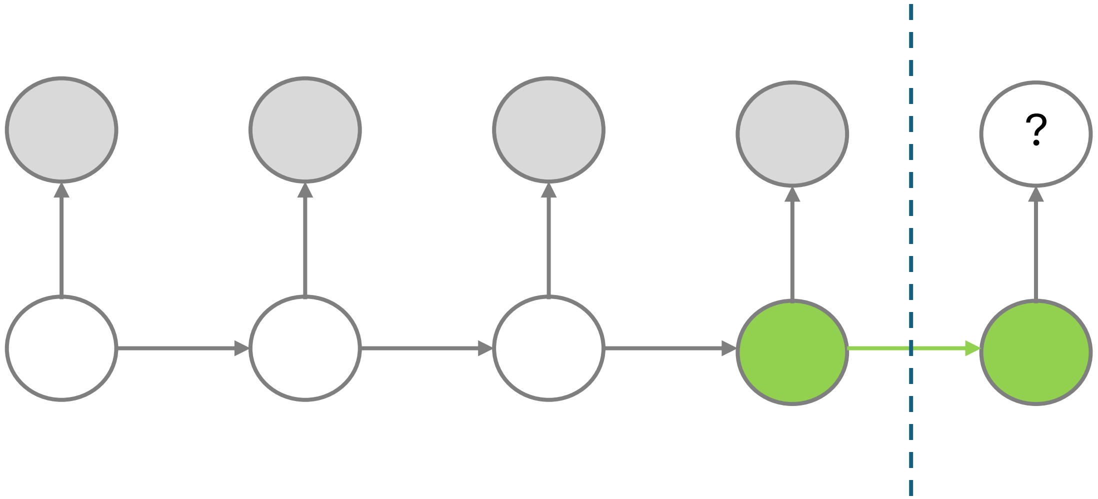
Have updated final latent variable with all data

The FILTERING problem

- Estimate current state

The PREDICTION problem

1. Marginalise and push forwards

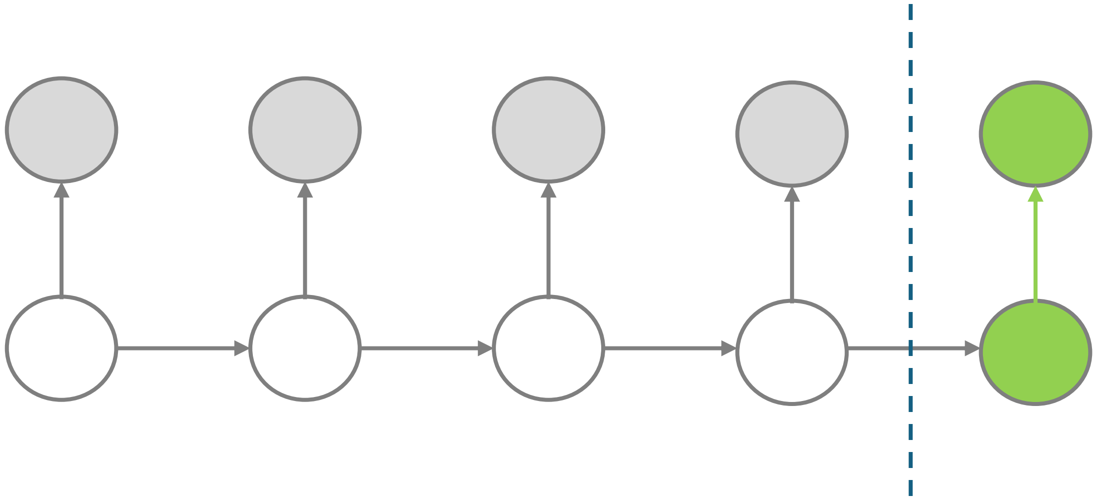


PREDICTION

Have updated final latent variable with all data

The FILTERING problem
- Estimate current state

The PREDICTION problem
1. Marginalise and push forwards
2. Marginalise again to obtain predictive density



$$p(D_{N+1}|D_{1:N}) = \int p(D_{N+1}|H_{N+1}, D_{1:N}) \int p(H_{N+1}|H_N, D_{1:N})p(H_N|D_{1:N})\, dH_N\, dH_{N+1}$$

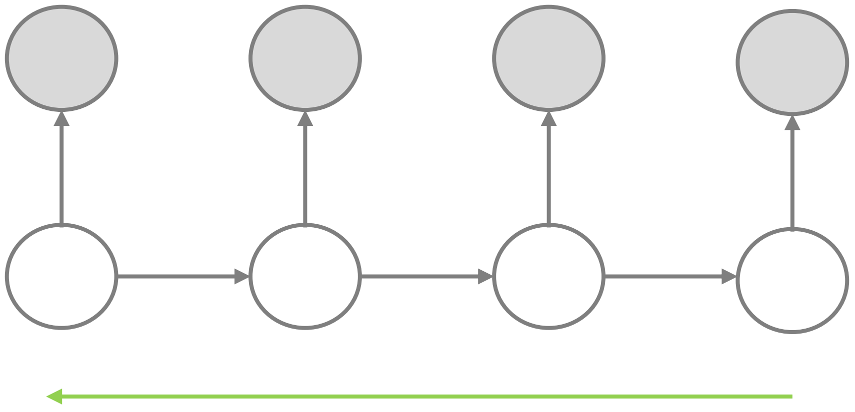
BACKWARD RECURSIONS

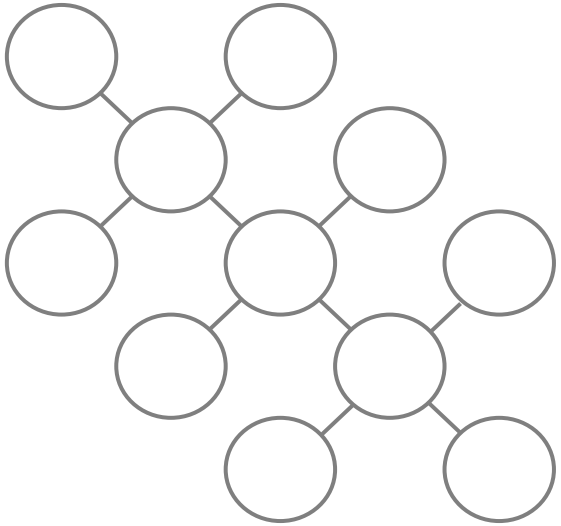
Have updated final latent variable with all data

The FILTERING problem
- Estimate current state

The PREDICTION problem
1. Marginalise and push forwards
2. Marginalise again to obtain predictive density

The SMOOTHING problem
1. What do we think now about what happened?
2. Backwards recursions: Rauch Tung Striebel
3. Each latent node incorporates every data observation





THE EXPECTATION MAXIMISATION ALGORITHM

FITTING THE MODEL

Frequentist models:
- Maximise likelihood

$$\mathcal{L}(\mathbf{y};\theta) = p(\mathbf{y}|\theta)$$

Find the parameter values that maximise likelihood

Bayesian models:
- Have parameters
- And latent variables
- Marginalise out the latent variables
- Maximise marginal likelihood

$$\begin{aligned}\mathcal{L}(\mathbf{y};\theta) &= p(\mathbf{y}|\theta) \\ &= \sum_{\mathbf{z}} p(\mathbf{y},\mathbf{z}|\theta)\end{aligned}$$

Find the parameter values and latent variable values that maximise marginal likelihood

- ✓ Same principle
- ➢ Hard to do

DECOMPOSITION

Proposition 1 – note that Y does not depend on Z

$$\log p(Y|\Theta) = \sum_Z q(Z) \log p(Y|\Theta)$$

Proposition 2 – product rule, take logs, rearrange

$$\begin{aligned}
p(Y, Z|\Theta) &= p(Z|Y, \Theta)p(Y|\Theta) \\
\log p(Y, Z|\Theta) &= \log p(Z|Y, \Theta) + \log p(Y|\Theta) \\
\log p(Y|\Theta) &= \log p(Y, Z|\Theta) - \log p(Z|Y, \Theta)
\end{aligned}$$

Let's go!

$$\begin{aligned}
\log p(Y\Theta) &= \sum_Z q(Z)\left[\log p(Y, Z|\Theta) - \log p(Z|Y, \Theta)\right] \\
&= \sum_Z q(Z)\left[\log p(Y, Z|\Theta) - \log p(Z|Y, \Theta) + \log q(Z) - \log q(Z)\right] \\
&= \sum_Z q(Z)\left[\log\left(\frac{p(Y, Z|\Theta)}{q(Z)}\right) - \log\left(\frac{p(Z|Y, \Theta)}{q(Z)}\right)\right] \\
&= \underline{\sum_Z q(Z)\log\left(\frac{p(Y, Z|\Theta)}{q(Z)}\right)} - \underline{\sum_Z q(Z)\log\left(\frac{p(Z|Y, \Theta)}{q(Z)}\right)} \\
&= \mathcal{L}(q, \Theta) + KL(q||p)
\end{aligned}$$

The strategy for decomposing the log marginal likelihood is interesting (to me, at least) because it involves some manoeuvres whose purpose is not initially obvious. The version on this slide is based on the presentation in Bishop 2006.

Substitute Proposition 2 into Proposition 1

Add *and* subtract log q(Z)

Rearrange the log q(Z) terms

We obtain these two terms

Which we will now use

ITERATIVE ALGORITHM

$$
\begin{aligned}
&= \sum_Z q(Z) \log\left(\frac{p(Y,Z|\Theta)}{q(Z)}\right) - \sum_Z q(Z) \log\left(\frac{p(Z|Y,\Theta)}{q(Z)}\right) \\
&= \mathcal{L}(q,\Theta) + KL(q||p)
\end{aligned}
$$

M Step: LEARNING

- Hold proposal distribution q fixed
- Obtain parameter values that maximise L(q, Theta)

Re-calibration of the model
Dreams?

E Step: INFERENCE

- Hold parameters fixed
- Obtain distribution q that minimises the KL divergence to the true posterior distribution, p
- This is exact for some models (Mixture of Gaussians, Linear Gaussian)
- If not exact, we can choose a sensible proposal distribution q

Corresponds to rational thought
Having seen the data, what do I think now?