Bayesian Approaches to Inverse Problems in Astrophysics and Cosmology

lecture 1

Dr. Prashin Jethwa
Institut für Astrophysik

What is a Bayesian Inverse Problem?

- What is an inverse problem?
    - Forward model: how to go from unknown model parameters to observed data
    - Inverse problem: how to go from observed data to unknown model parameters

- Bayesian approach:
    - Encode the model as a probability distribution over the unknown model parameters and observed data
    - Inverse problem becomes a problem of probabilistic inference:
        - i.e. what is the probability distribution of unknown parameters given the observed data?

Probabilistic Programming Languages (PPLs)

Wikipedia:

“Probabilistic programming is a programming paradigm in which probabilistic models are specified and inference for these models is performed automatically"

- sometimes standalone languages, sometimes packages within other languages e.g. in Python
- take advantage of modern hardware (e.g. GPUs) and software (e.g. automatic differentiation)
- *allow you to perform statistical inference on larger and more complex models than was possible previously*

Goals of the next three lectures

- Understand PPLs
- Be able to apply these to scientific modelling problems

For this course, we will use ***numpyro***

Lecture outline

- Recap of Bayesian statistics
- Generative Models
- Bayesian Networks / Directed Acyclic Graphs
- Example: hierarchical linear regression in *numpyro*

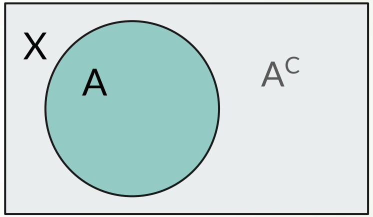
By the end of this lecture you should be able to:

- Read and write probabilistic notation
- Define the concept of a generative model
- Communicate generative models via Bayesian Networks
- Describe the concept of partial pooling in hierarchical models

Recap on Bayesian statistics

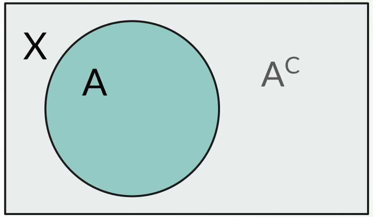
Probability

- Probability theory, we assign probabilities to events in sets
- There are some familiar axioms, e.g.
  - $\mathbb{P}(A^C) = 1 - \mathbb{P}(A)$



Probability

- Probability theory, we assign probabilities to events in sets
- There are some familiar axioms, e.g.
    - $\mathbb{P}(A^C) = 1 - \mathbb{P}(A)$



- Practically, for modelling, we never start with abstract set
- It’s more convenient to start with a probability distribution function

Probability distribution functions

- Think of these as *ready-made, useful* assignments of probability over familiar, useful sets

- $p(x)$ is a function over elements x in a domain X such that:
  - $p(x) \geq 0$ for all $x \in X$
  - $\int_X p(x)\, dx = 1$

- the support is the subset of the domain where $p(x) > 0$

- If the domain X is:
  - Continuous
    - $p(x)$ called a probability density function
    - evaluate probabilities by integrating i.e. $\mathbb{P}(a<x<b) = \int_a^b p(x)\, dx$
  - Discrete
    - $p(x)$ called a probability mass function
    - evaluate probabilities directly i.e. $\mathbb{P}(x=a) = p(a)$

Probability distribution functions: notation

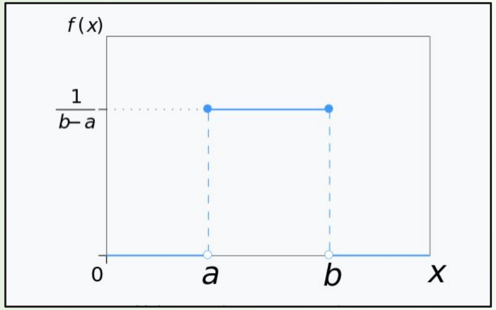
- Parameters
  - often pdfs depend on parameter(s) θ
  - we use the following notational convention:
    - ***p(x ; θ)*** if the parameter θ known/fixed, put it after a semicolon ;
    - ***p(x , θ)*** no semicolon if the parameter is unknown i.e. this is the joint distribution on ***x*** and ***θ***

- Sampling
  - *x ~ p(x)*
  - x is sampled from *p(x)*

- some common distributions have their own symbols/abbreviations e.g.
  - U Uniform
  - N Normal
  - Binom Binomial
  - Poiss Poisson

Probability distribution functions: example

Uniform distribution



- Parameters:
  - a, b : the start and end
- Domain:
  - real numbers
- Support:
  - a < real numbers < b
- Notation:
  - U(a, b) ← represents the distribution
  - U(x ; a, b) ← represents the distribution function
  - x ~ U(a,b) ← sampling

Exercise: notation

Write an expression for

“the probability distribution of x conditional on y and N with a fixed parameters a and b”

Exercise: notation

Write an expression for

"the probability distribution of x conditional on y and N with a fixed parameters a and b"

Solution:

$p(x \mid y, N ; a, b)$

Multivariate distributions

- Given two variables $x \in X$ and $y \in Y$ we can define a multivariate distribution function over both variables:
  - $p(x, y)$ = the joint distribution over x and y

- Given a joint distribution, if we are only interested in one of the variables, we can marginalise over the others:
  - $p(x) = \int_Y p(x, y)\, dy$ = the marginal distribution of x

- If we know the value of one variable y then the distribution of x conditional on y is
  - $p(x \mid y)$ = *… or …* the distribution of x given y
  - $= p(x, y) / p(y)$ = the conditional is the joint divided by the marginal

Conditional Probability

- *p(x | y) = p(x, y) / p(y)*
    - where does this come from?

- Easier to see with a concrete example:
    - we roll two dice, numbered 1 to 6
    - what is the probability that the first roll equals 1 given that the total of both rolls equals 7?

roll 1

roll 2	**1**	**2**	**3**	**4**	**5**	**6**
1	2	3	4	5	6	7
2	3	4	5	6	7	8
3	4	5	6	7	8	9
4	5	6	7	8	9	10
5	6	7	8	9	10	11
6	7	8	9	10	11	12

Conditional Probability

- $p(x \mid y) = p(x, y) \,/\, p(y)$
  - where does this come from?

- Easier to see with a concrete example:
  - we roll two dice, numbered 1 to 6
  - what is the probability that the first roll equals 1 given that the total of both rolls equals 7?

- Solution:
  - conditional = joint / marginal
  - $\mathbb{P}(\text{first roll} = 1 \mid \text{total} = 7)$

$$= \mathbb{P}(\text{first roll} = 1 \textbf{ and } \text{total} = 7) \,/\, \mathbb{P}(\text{total} = 7)$$
$$= (1/36) \,/\, (6/36)$$
$$= 1/6$$

roll 1

roll 2

	1	**2**	**3**	**4**	**5**	**6**
1	2	3	4	5	6	7
2	3	4	5	6	7	8
3	4	5	6	7	8	9
4	5	6	7	8	9	10
5	6	7	8	9	10	11
6	$\boxed{7}$	8	9	10	11	12

Conditional Independence

- A variable x is independent of another variable y if knowing the value of y gives us no extra information about x

- In other words:
  - $p(x \mid y) = p(x)$ the conditional distribution is equal to the marginal distribution

- ***Question:*** if x is independent of y, is y independent of x ...?

Conditional Independence

- A variable x is independent of another variable y if knowing the value of y gives us no extra information about x

- In other words:
  - $p(x \mid y) = p(x)$ the conditional distribution is equal to the marginal distribution

- ***Question:*** if x is independent of y , is y independent of x …?
  - Yes!
  - So we can say x and y are independent
  - The proof comes from…

Bayes' Theorem

- Two ways to express the joint pdf:
  - p(x, y) = p(x | y) p(y)
  - p(x, y) = p(y | x) p(x)

- Equate the two:
  - p(x | y) p(y) = p(y | x) p(x)

- and rearrange:
  - p(x | y) = p(y | x) p(x) / p(y) ← Bayes' theorem

- if x is independent of y
  - p(x | y) = p(x)
  - → p(y | x) = p(y)
  - → y is independent of x
  - i.e. being independent is symmetric

Thomas Bayes

Portrait purportedly of Bayes used in a 1936 book,[1] but it is doubtful whether the portrait is actually of him.[2] No earlier portrait or claimed portrait survives.

Interpretation of Bayes' theorem

For inference problems we interpret Bayes' theorem as follows:

- θ = model parameters
- y = observed data

$$p(\theta \mid y) \quad = \quad p(y \mid \theta) \quad p(\theta) \;\; / \;\; p(y)$$

Posterior
probability of parameters given some observed data i.e. what we are interested in

Likelihood
probability of the data given some parameters

Prior
our belief - encoded in a probability distribution - about the parameters *before* observing any data

Marginal Likelihood / Model Evidence
the probability of the data? Easier to interpret if we write the un-marginalised version:

$$p(y) = \int_Y p(y \mid \theta)\, p(\theta)\, d\theta$$

Think of it as a normalising factor which that the posterior integrates to 1.
Often possible to ignore it

Independent and identically distributed (iid) data

- Say we have N data points

$$\boldsymbol{y} = (y_1, y_2, \dots, y_N)$$

- If they are independent then the likelihood can be factorised as,

$$p(\boldsymbol{y} \mid \theta) \quad = \quad p_1(y_1 \mid \theta)\, p_2(y_2 \mid \theta) \dots p_N(y_N \mid \theta)$$

- If they are also identically distributed, then all of the factors are identical,

$$p(\boldsymbol{y} \mid \theta) \quad = \prod_{i=1, \dots, N} p(y_i \mid \theta)$$

Generative Models

What is a model?

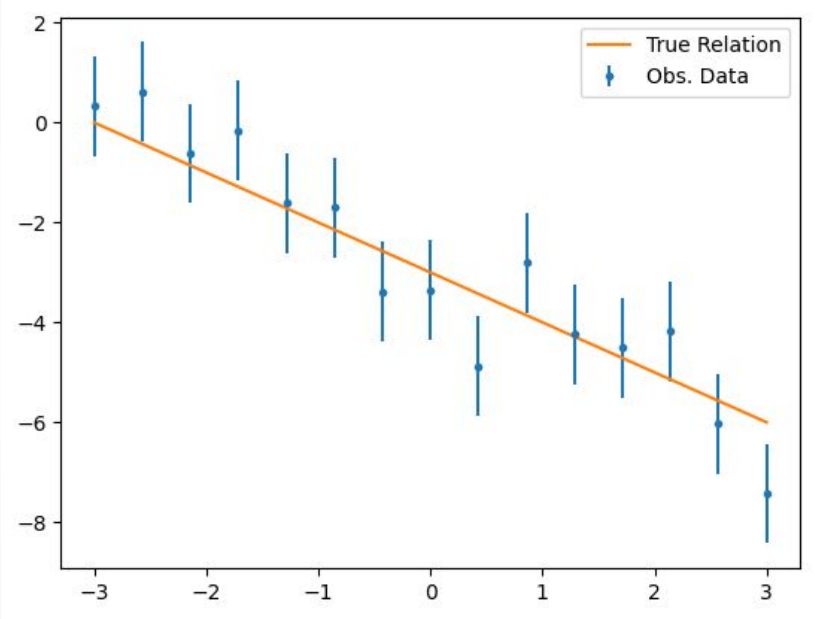
Wikipedia:

A statistical model is a mathematical model that embodies a set of statistical assumptions concerning the generation of sample data (and similar data from a larger population). A statistical model represents, often in considerably idealized form, the data-generating process.

Example: Linear Regression

- Say some observations give us datapoints:
  $$(x_i, y_i) \text{ for } i = 1, \ldots, N$$
  with a known observational error σ

- How can we infer the true linear relation?

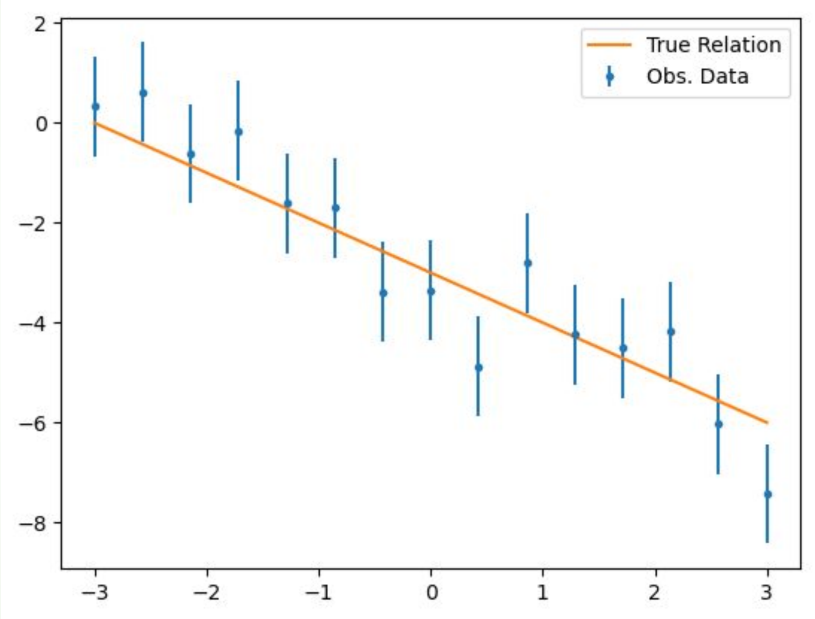


Linear Regression - the standard description

- Find gradient *m* and intercept *c* which that minimize χ^2 difference between data and line i.e. find

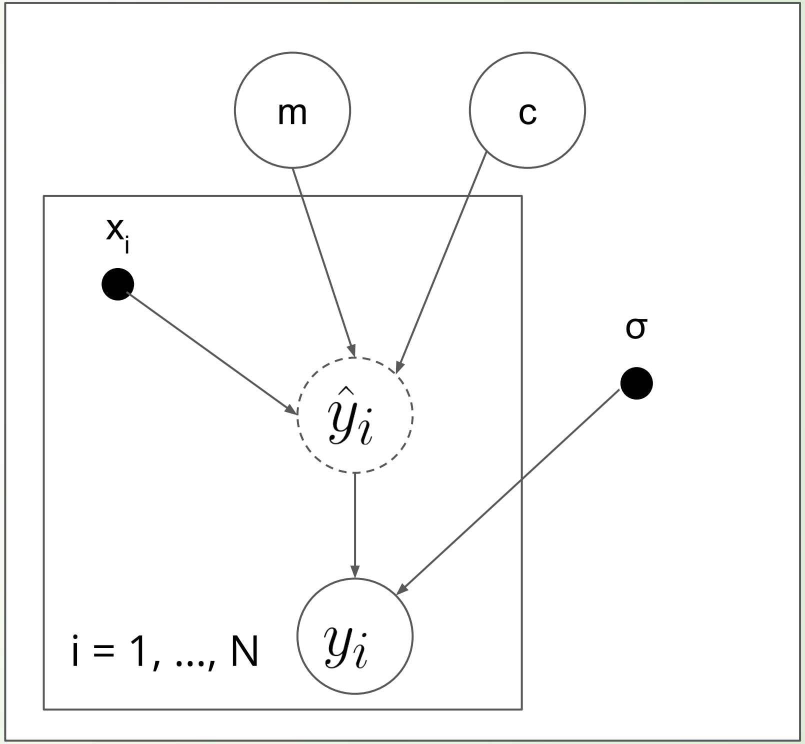
$$\underset{(m,c)}{\operatorname{argmin}} \left(\frac{(mx_i + c) - y_i}{\sigma} \right)^2$$

- Can be solved using standard techniques: linear least squares regression
- What is the associated generative model?



Linear Regression - the generative model

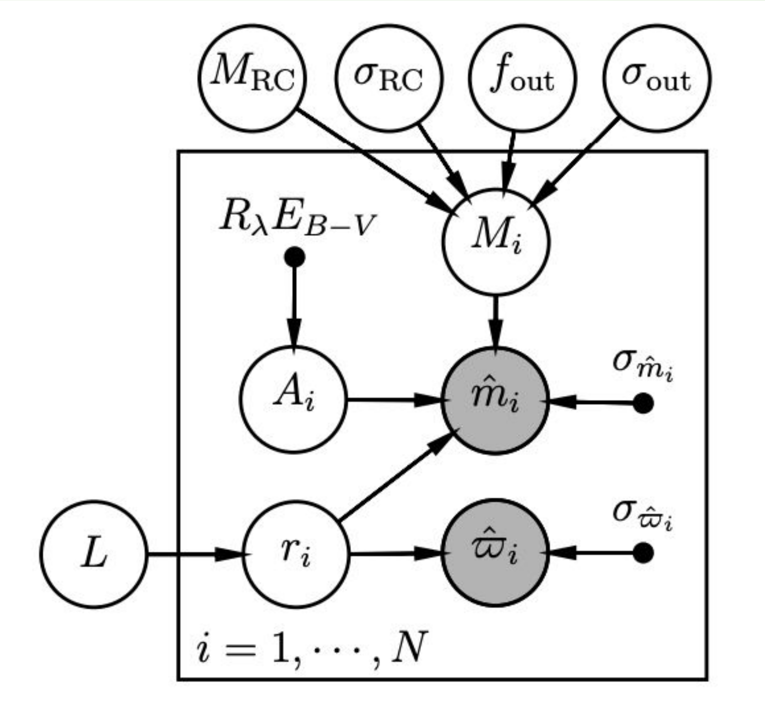
- What is the data generating process for linear regression?
- There is a gradient *m*, intercept *c* and set of fixed x positions x_i
- Noise free y-values are $\hat{y}_i = mx_i + c$
- Assuming a normal distribution for the noise model, the observed y-values are samples $y_i \sim \mathcal{N}(\hat{y}_i, \sigma)$



Example: linear regression in numpyro

Generative model

- A set of instructions for how to generate observed data according to a probabilistic model
- The instructions tell us how to combine unknown parameters to the generate observed data
- Often represented graphically using Bayesian Network
- Useful framework for building and communicating complex models

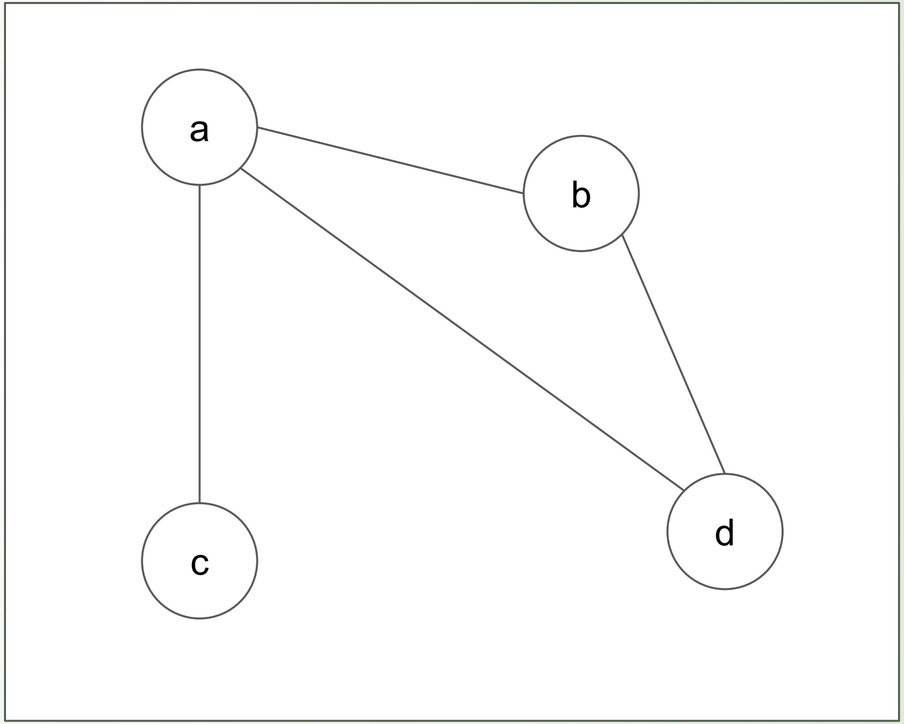


A Bayesian network representing a generative model from Hawkins et al 2017 paper: *Red clump stars and Gaia: Calibration of the standard candle using a hierarchical probabilistic model*

Bayesian Networks / Directed Acyclic Graphs (DAG)

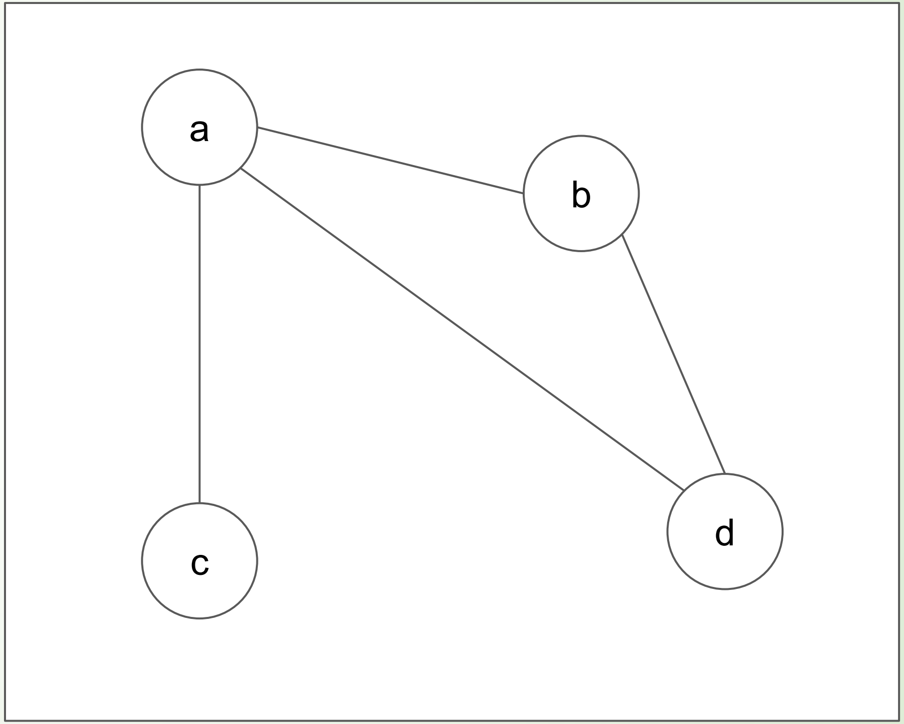
Bayesian Network / DAG : definitions

- A Bayesian network is a *graphical* representation of a generative model
- A *graph* is a collection of nodes and edges
  - nodes represent variables
  - edges represent dependencies between variables



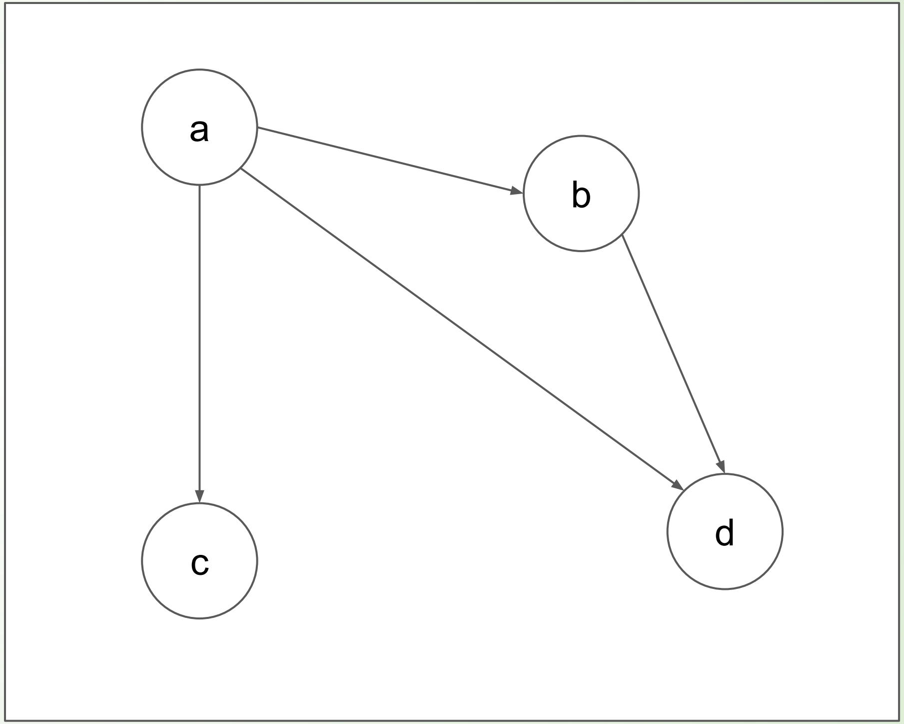
Bayesian Network / DAG : definitions

- A Bayesian network is a *graphical* representation of a generative model
- A *graph* is a collection of nodes and edges
    - nodes represent variables
    - edges represent dependencies between variables
- Bayesian networks are a specific type of graph: directed acyclic graphs (DAGs)



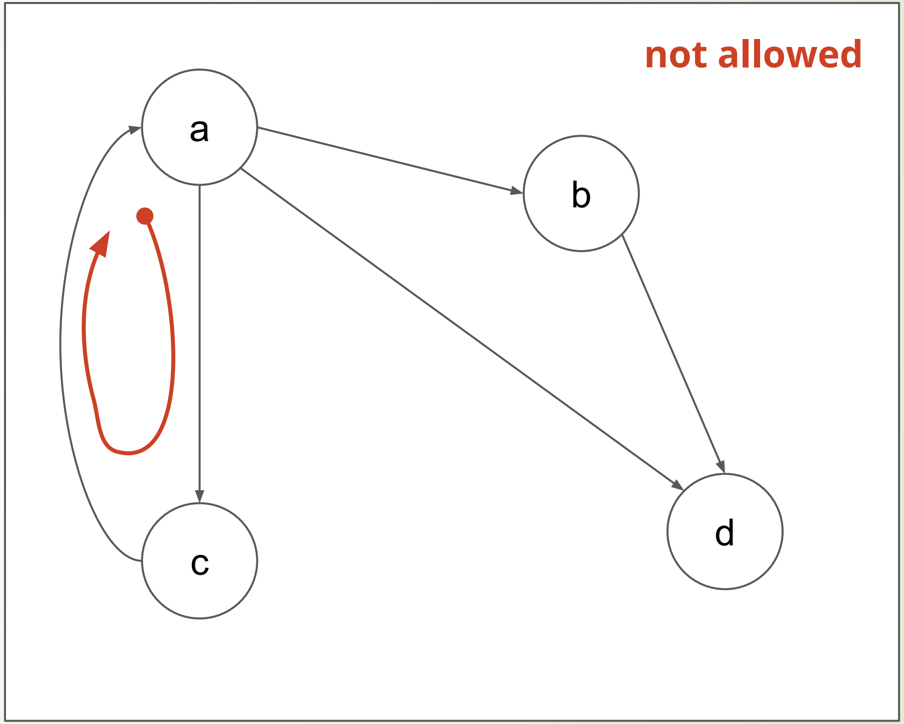
Bayesian Network / DAG : definitions

- A Bayesian network is a *graphical* representation of a generative model
- A *graph* is a collection of nodes and edges
  - nodes represent variables
  - edges represent dependencies between variables
- Bayesian networks are a specific type of graph: directed acyclic graphs (DAGs)
  - Directed
    - edges have direction
    - *parent* node points to *child* node
    - for generative models a→b often means “a *causes* b” or “a depends on b”



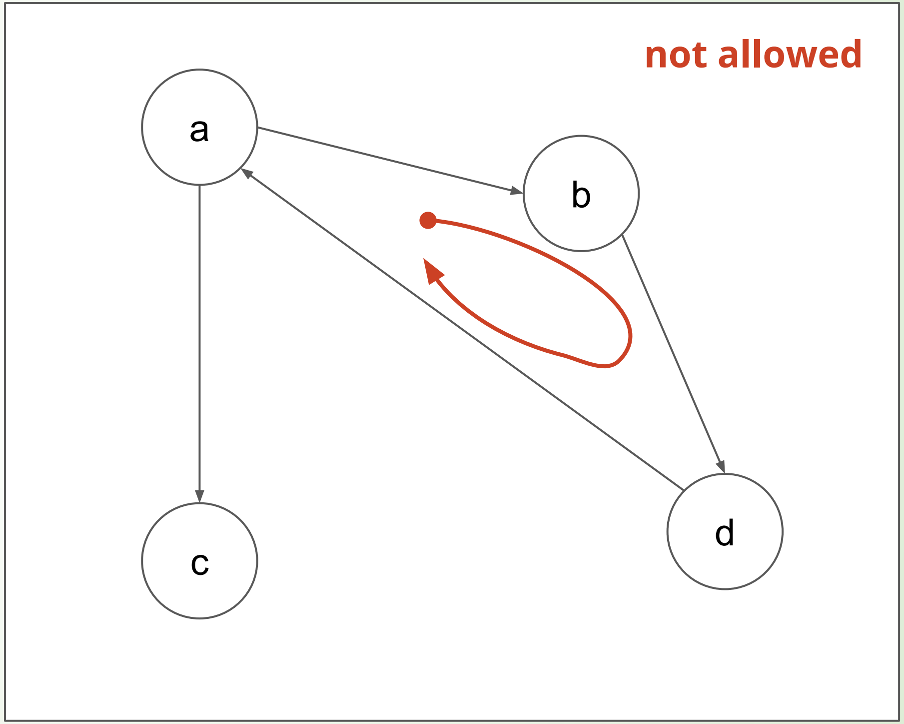
Bayesian Network / DAG : definitions

- A Bayesian network is a *graphical* representation of a generative model
- A *graph* is a collection of nodes and edges
    - nodes represent variables
    - edges represent dependencies between variables
- Bayesian networks are a specific type of graph: directed acyclic graphs (DAGs)
    - Directed
        - edges have direction
        - *parent* node points to *child* node
        - for generative models a→b often means "a *causes* b" or "a depends on b"
    - Acyclic
        - no cycles i.e. closed loops



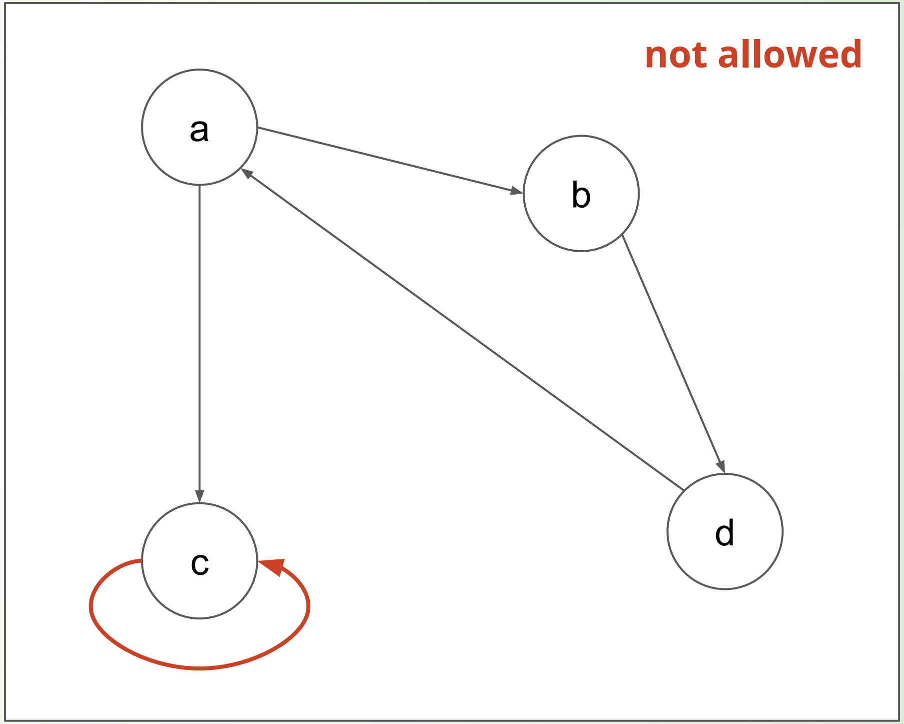
Bayesian Network / DAG : definitions

- A Bayesian network is a *graphical* representation of a generative model
- A *graph* is a collection of nodes and edges
  - nodes represent variables
  - edges represent dependencies between variables
- Bayesian networks are a specific type of graph: directed acyclic graphs (DAGs)
  - Directed
    - edges have direction
    - *parent* node points to *child* node
    - for generative models a→b often means “a *causes* b” or “a depends on b”
  - Acyclic
    - no cycles i.e. closed loops



Bayesian Network / DAG : definitions

- A Bayesian network is a *graphical* representation of a generative model
- A *graph* is a collection of nodes and edges
  - nodes represent variables
  - edges represent dependencies between variables
- Bayesian networks are a specific type of graph: directed acyclic graphs (DAGs)
  - Directed
    - edges have direction
    - *parent* node points to *child* node
    - for generative models a→b often means “a *causes* b” or “a depends on b”
  - Acyclic
    - no cycles i.e. closed loops

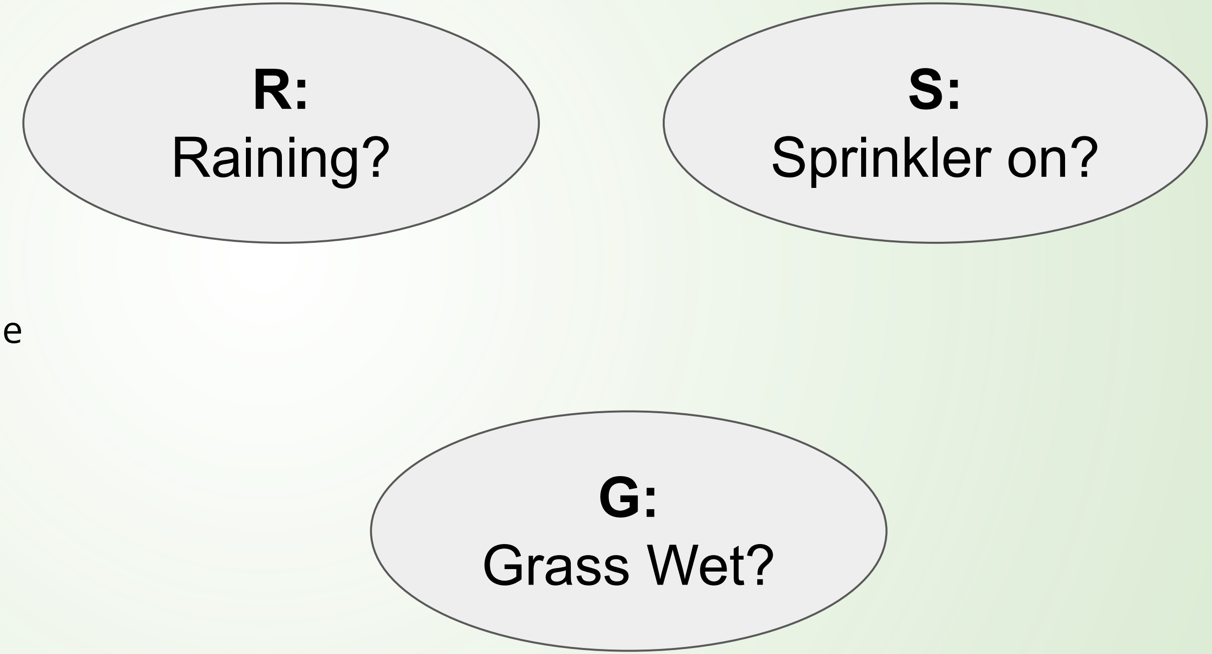


Bayesian Network: wet grass example

- Three True/False variables:

  R - is it raining?
  S - is the sprinkler on?
  G - is the grass wet?

- Draw the edges: which variable influences which others?



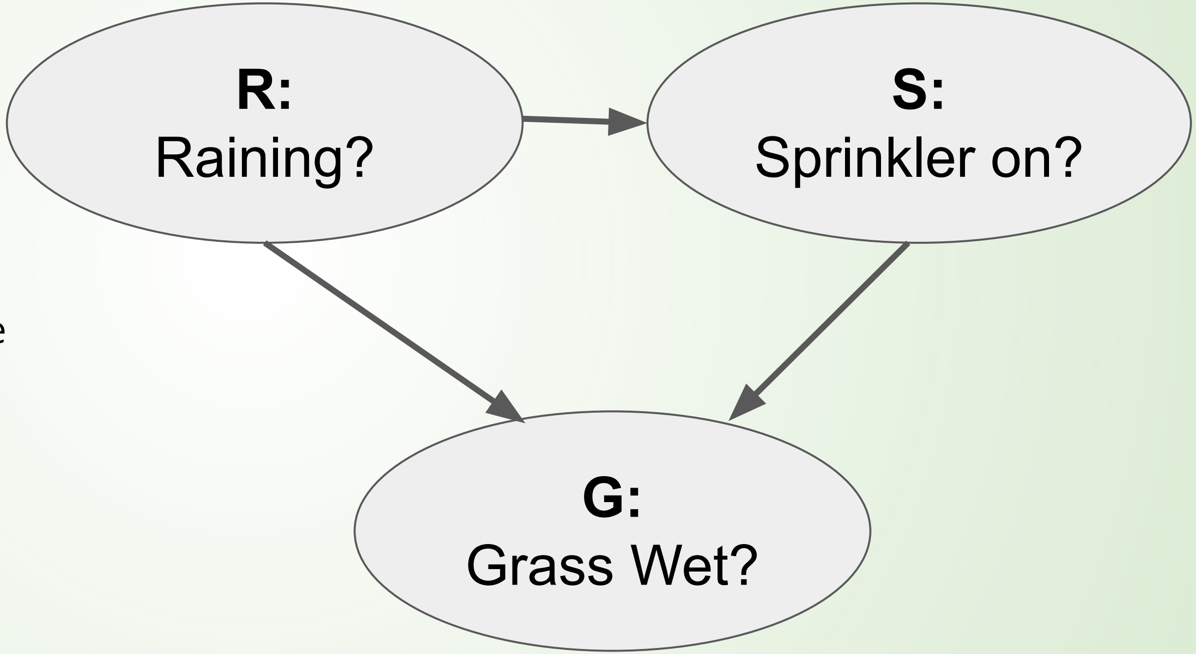
Bayesian Network: wet grass example

- Three True/False variables:

  R - is it raining?
  S - is the sprinkler on?
  G - is the grass wet?

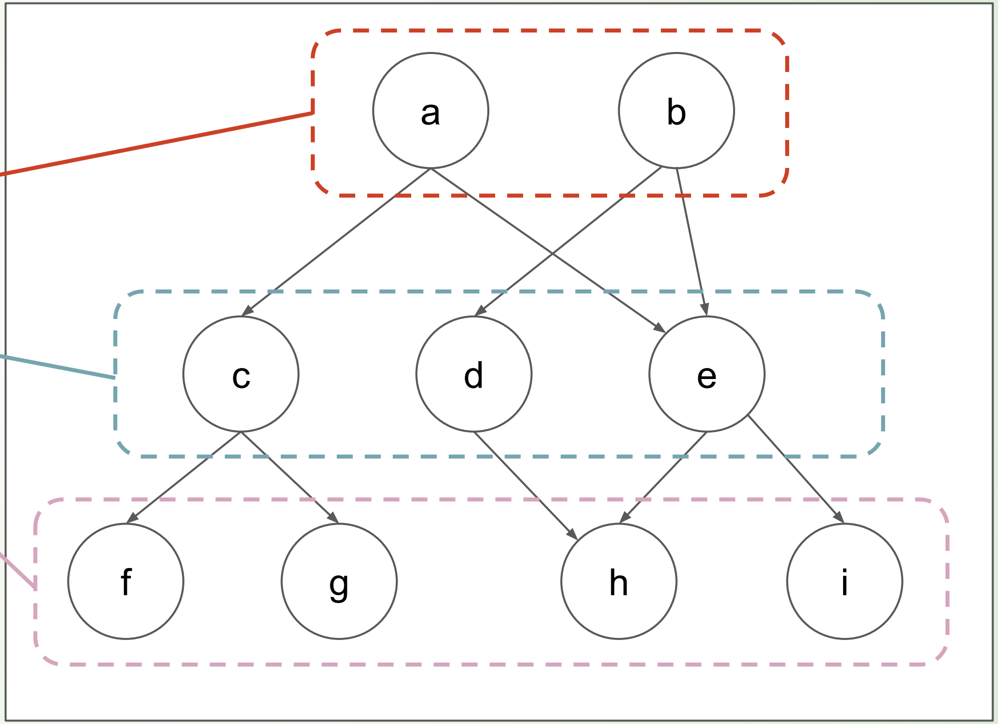
- Draw the edges: which variable influences which others?

- See Wikipedia for more details about this example

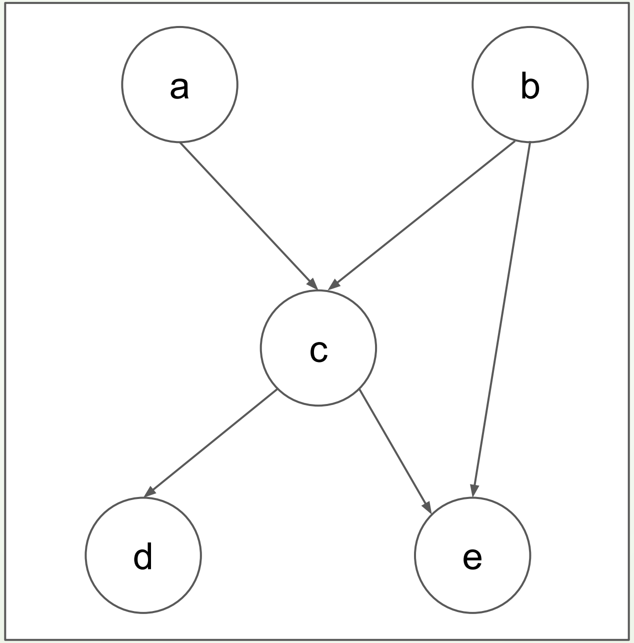


Bayesian Networks: hierarchical models

- ***Hierarchical models*** have multiple layers
- *Root nodes* have no parent
- Intermediate nodes
- *Leaf nodes* have no children



Bayesian Networks factorise the joint probability density into conditionally dependent terms

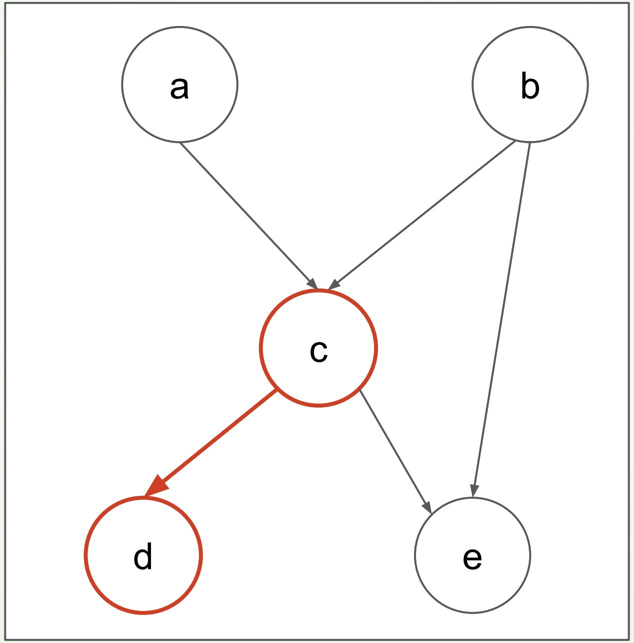


p(a, b, c, d, e) = ... ?

Bayesian Networks factorise the joint probability density into conditionally dependent terms

start at the leaf nodes

add a factor for every dependency

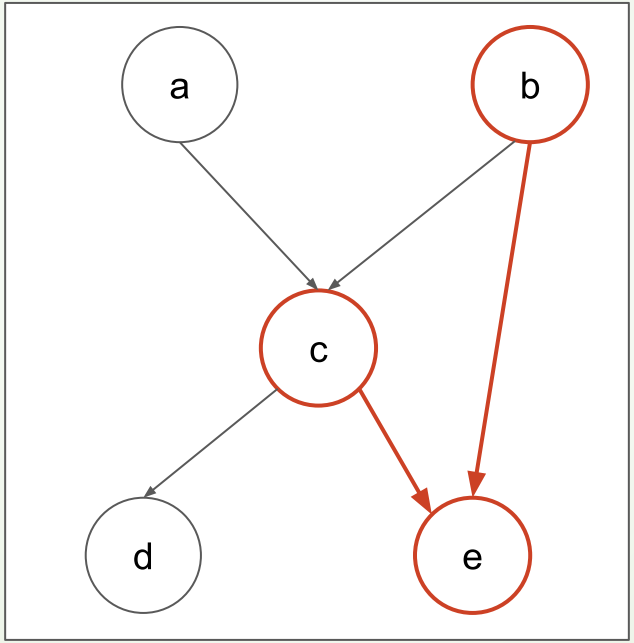


$$p(a, b, c, d, e) = p(d \mid c) \ldots$$

Bayesian Networks factorise the joint probability density into conditionally dependent terms

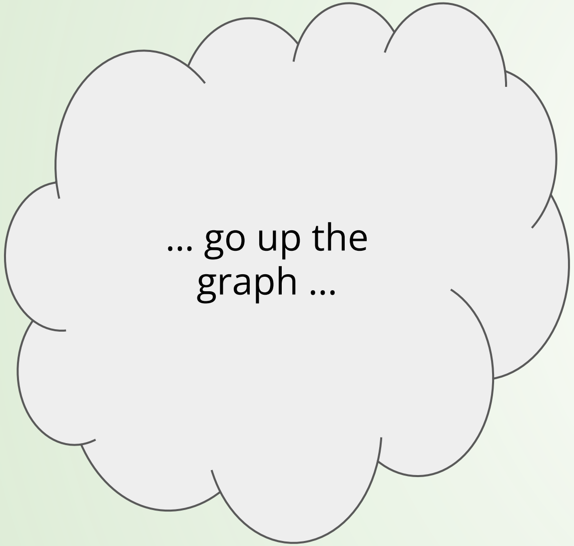
start at the leaf nodes

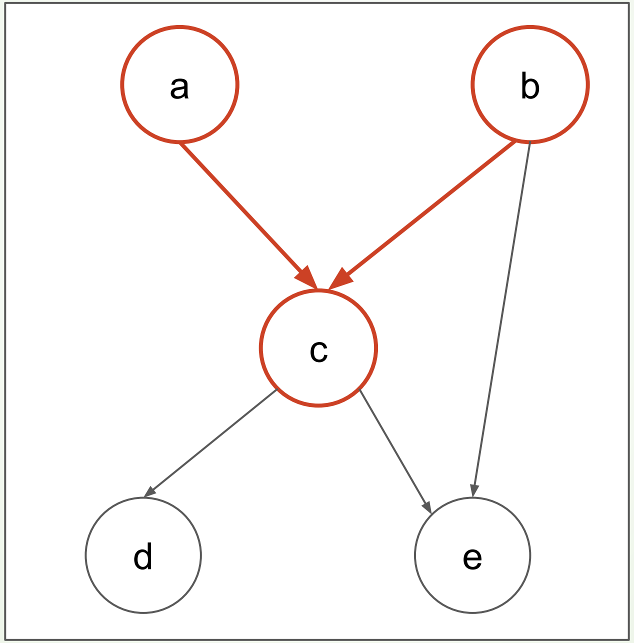
add a factor for every dependency



p(a, b, c, d, e) = p(d|c) p(e|c, b) ...

Bayesian Networks factorise the joint probability density into conditionally dependent terms



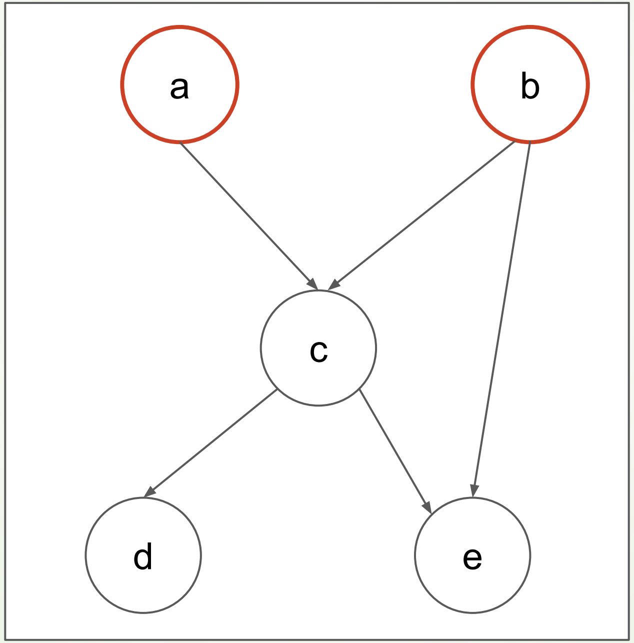


p(a, b, c, d, e) = p(d|c) p(e|c, b) p(c|a, b) …

Bayesian Networks factorise the joint probability density into conditionally dependent terms

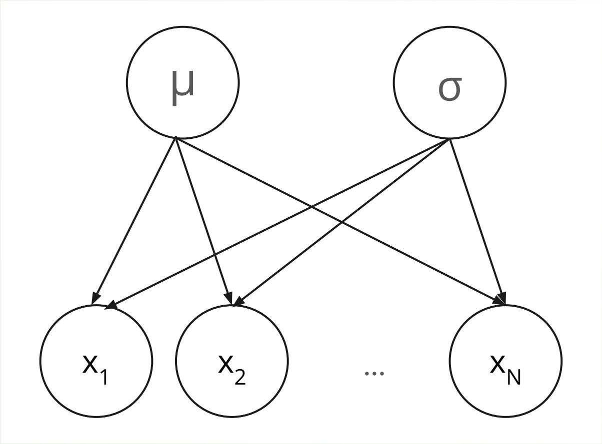
... till you reach the root nodes.

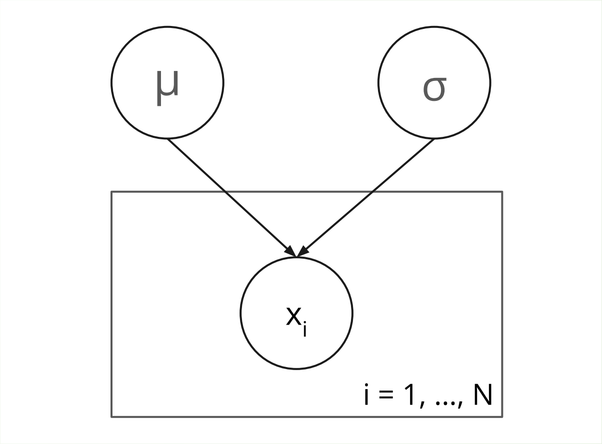
Add a marginal distribution for each root node.



$$p(a, b, c, d, e) = p(d|c)\, p(e|c, b)\, p(c|a, b)\, p(a)\, p(b)$$

Plate notation is used for iid variables

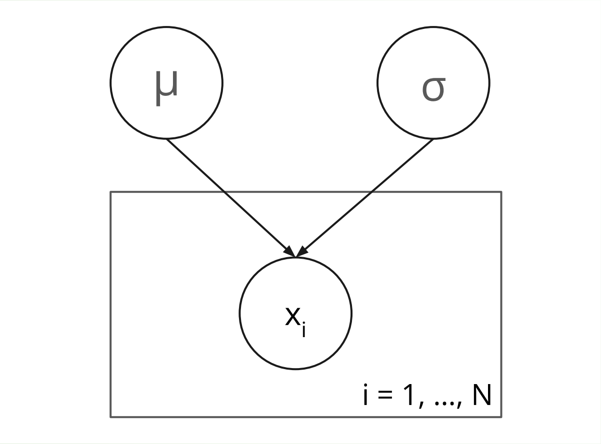


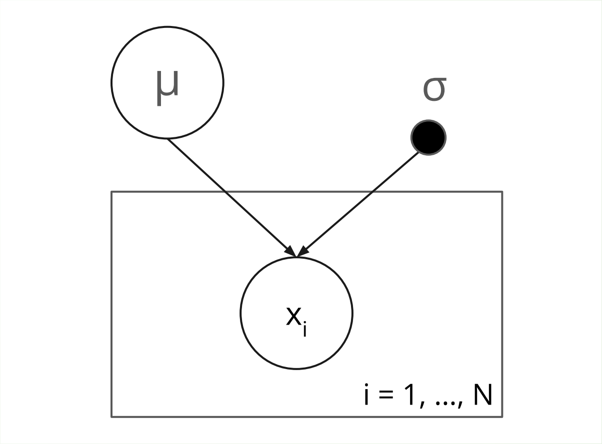


the box is called a *plate*

Fixed variables are represented by dots

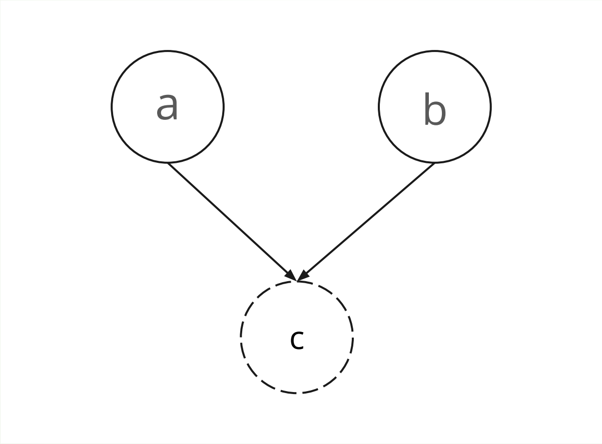
e.g. if we are treating σ as a fixed rather than an unknown parameter, then ...





Deterministic nodes are dashed

if a variable is related to others deterministically rather than probabilistically, it is given a dashed line



e.g.

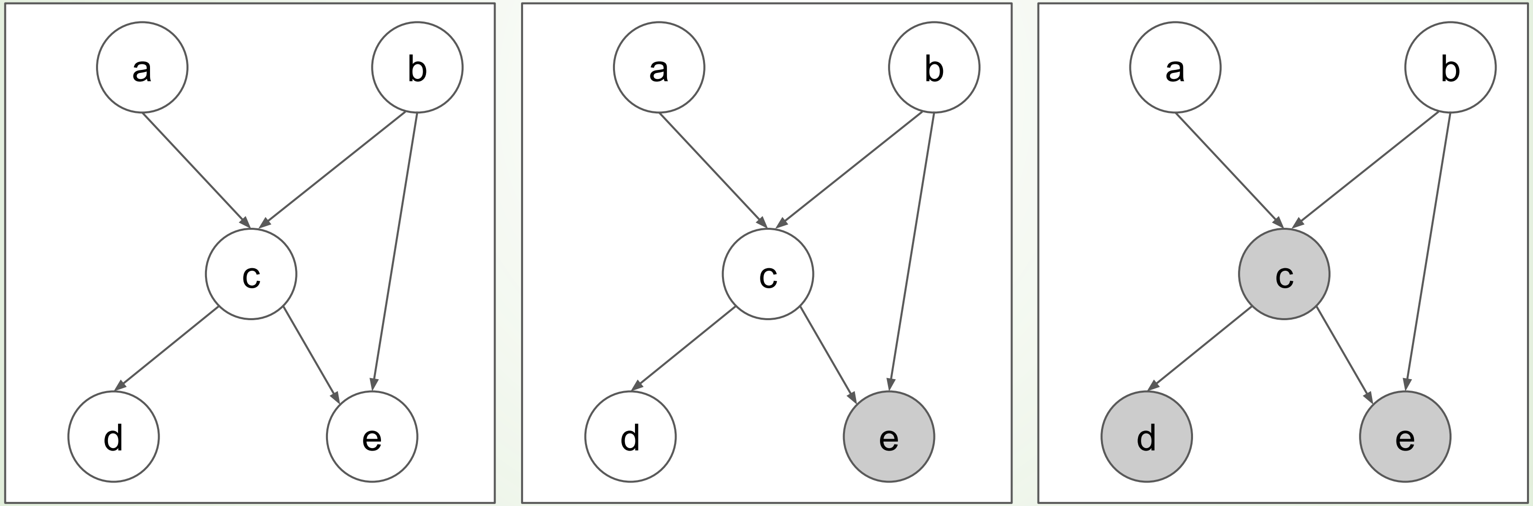
$a \sim p(a)$

$b \sim p(b)$

$c = a + b$

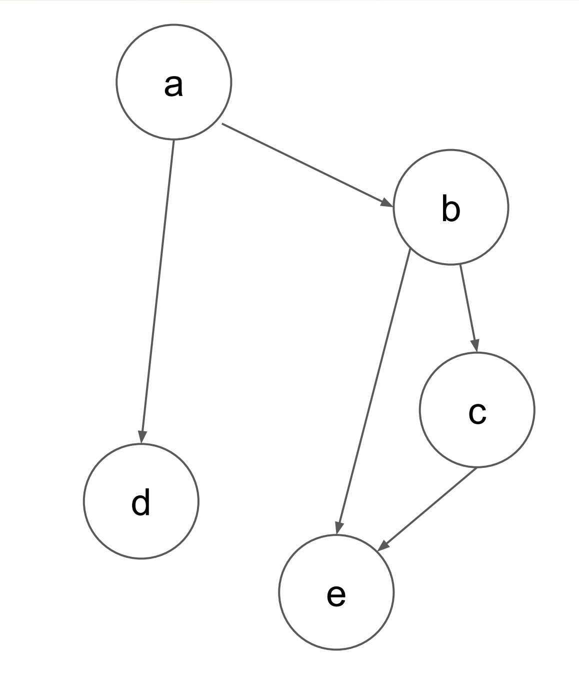
Observed values

- Variables which are observed (i.e. data) are shaded in
- The graph no longer represents the joint distribution, but the conditional distribution given the observed values

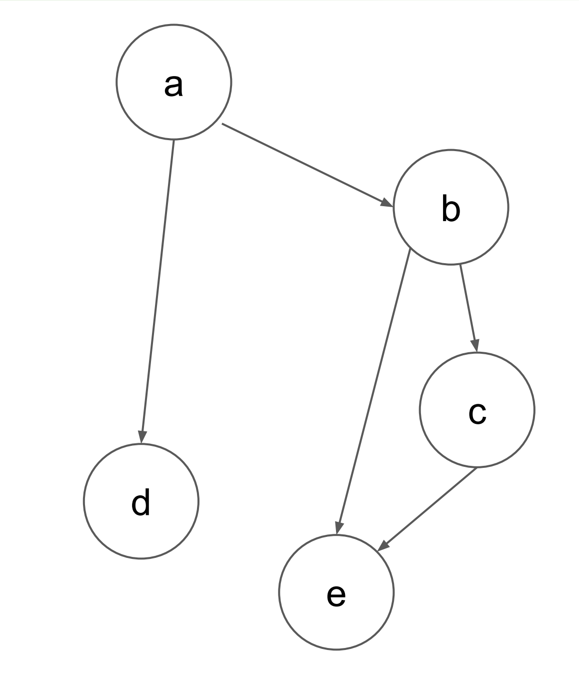


p(a, b, c, d, e) p(a, b, c, d | e) p(a, b| c , d, e)

Exercise 1: write the factorised joint distribution corresponding to this Bayesian network



Exercise 1: write the factorised joint distribution corresponding to this Bayesian network



$p(a,b,c,d,e) =$

$$p(d|a)\ p(e|c,b)\ p(c|b)\ p(b|a)\ p(a)$$

Defining a generative models

1. Identify the variables of interest needed to generate the observed data

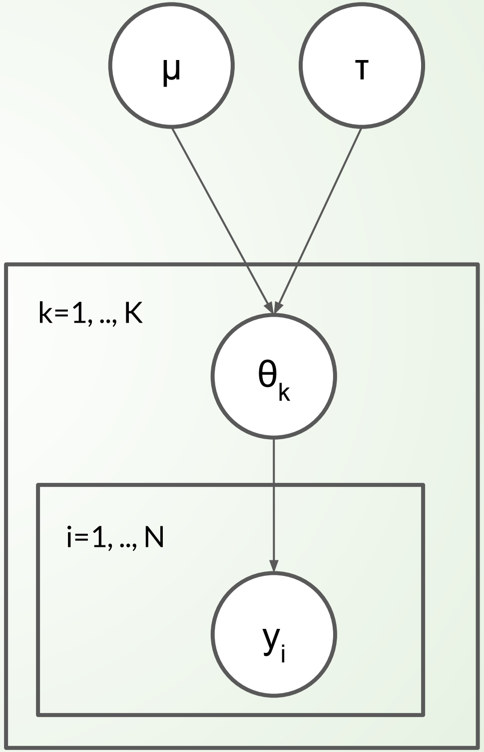
2. Draw the Bayesian showing dependencies between variables and the observed data

3. For each factor in the network, specify a probability distribution or deterministic function

Example notebook:
hierarchical linear regression in numpyro

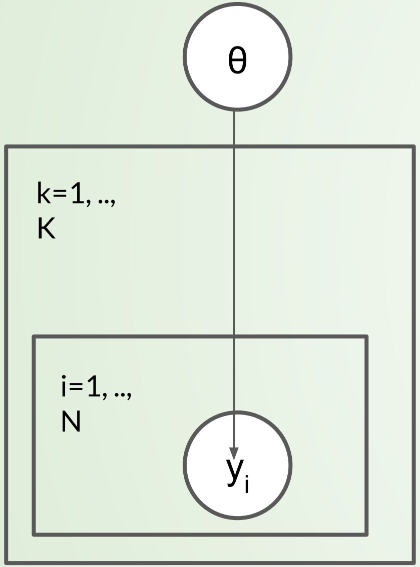
Hierarchical Models

- Useful for modelling *heterogeneity* in your data
- Say data *y* was collected in *K* different *contexts*
- We may expect data in different contexts to have different parameters θ_k
- *Population parameters* - mean μ and scale τ - control the distribution of per-context parameters θ_k
- Limits:
  - $\tau \to 0$: no variation allowed between contexts
  - $\tau \to$ infty : large variations allowed

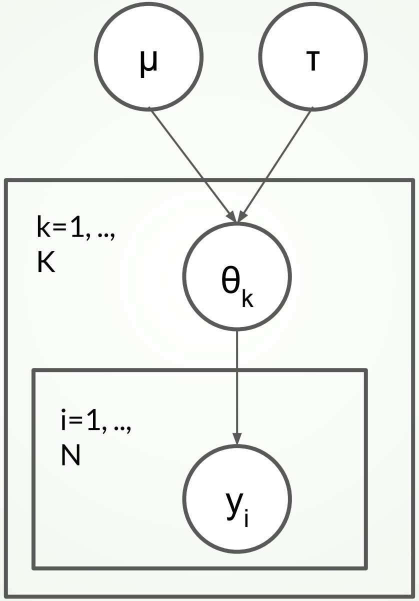


$$\theta_k \sim \text{normal}(\mu, \tau).$$

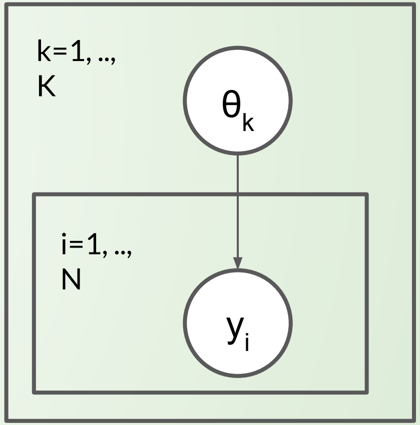
Hierarchical Models Allow *Partial Pooling* of Information between Contexts



Complete Pooling:
all contexts share parameters
Ignores heterogeneity between contexts



Partial Pooling:
per-context parameters related via population parameters - information shared between contexts allowing for heterogeneity



No Pooling:
each context treated independently
ignores similarity between contexts

References

- Michael Betancourt's (STAN developer) blog:
    - https://betanalpha.github.io/writing/
    - For today:
        - Foundations of Probability Theory and Conditional Probability Theory
        - Product Placement, especially Section 4

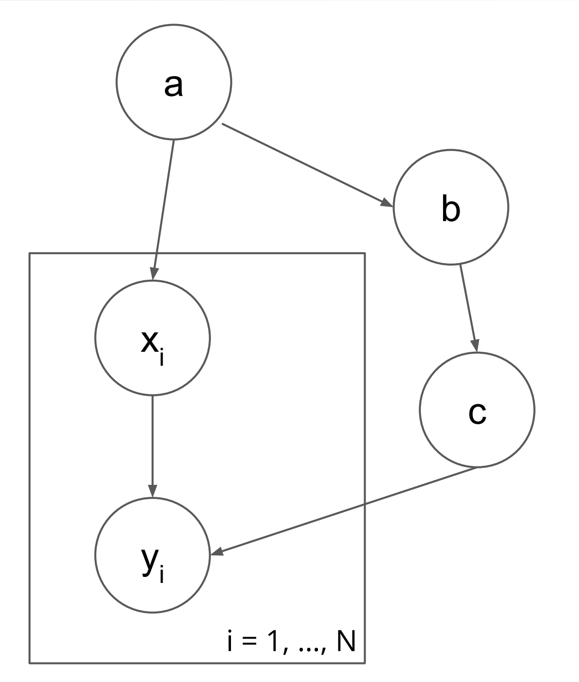
- Bishop, C. M. (2006). Pattern recognition and machine learning
    - For Bayesian networks

- Astronomy papers with Bayesian networks:
    - Red clump stars and Gaia: Calibration of the standard candle using a hierarchical probabilistic model - Hawkins+17
    - Hierarchical Bayesian inference of galaxy redshift distributions from photometric surveys - Leistedt + 16
    - Approximate inference for constructing astronomical catalogs from images - Regier + 19
    - Improved constraints on cosmological parameters from Type Ia supernova data - March +11

Exercises

- Run the notebooks for linear regression and hierarchical linear regression
  - install numpyro
- A few more examples with Bayesian Networks…

Exercise 2: write the factorised joint distribution corresponding to this Bayesian network



Exercise 3: draw a Bayesian network corresponding to this factorisation of a joint probability function

$p(a, b, c, d \,;\, \theta) = p(a \mid b, c)\, p(b \mid c \,;\, \theta)\, p(d \mid c)\, p(c)$