Inference with Bayesian Networks

Greg Britten

Darwin Ocean Modelling Lab
EAPS, MIT

gbritten@mit.edu

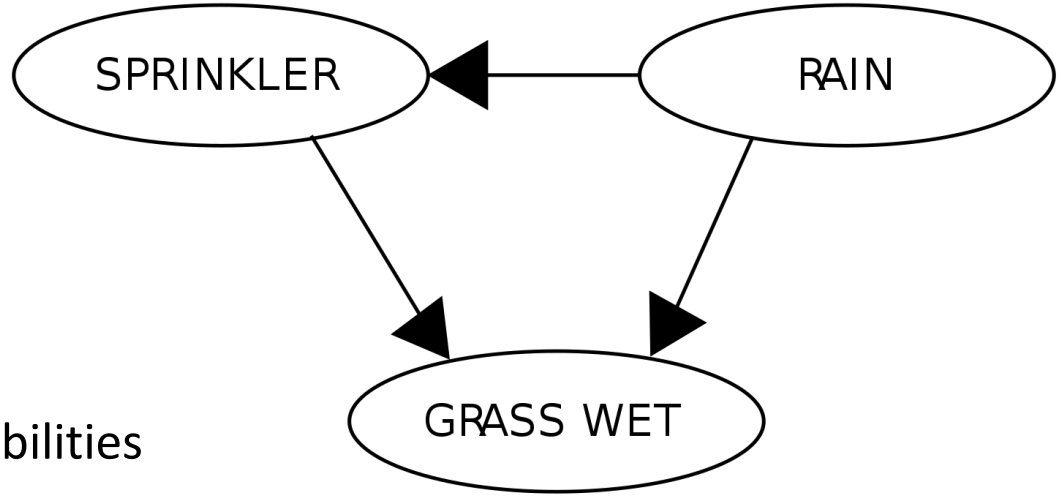
TUTORIAL GitHub: *https://github.com/gregbritten/git_ml_essg*

Outline

- What is a Bayesian network and what does ‘inference with a Bayesian network’ mean?
- CONCEPTS: Directed Markov graph, hierarchical network model
- Perform sampling via probabilistic programming languages (example with Stan, others available)
- Worked example via linear regression on a hierarchical network

A simple Bayesian network

RAIN	SPRINKLER T	SPRINKLER F
F	0.4	0.6
T	0.01	0.99



RAIN T	RAIN F
0.2	0.8

State transition matrices

Directed and *acyclic*: lower level probabilities are updated conditional on particular choices (paths) at upper levels

SPRINKLER	RAIN	GRASS WET T	GRASS WET F
F	F	0.0	1.0
F	T	0.8	0.2
T	F	0.9	0.1
T	T	0.99	0.01

Two ways people talk about ‘inference’ on a Bayesian network

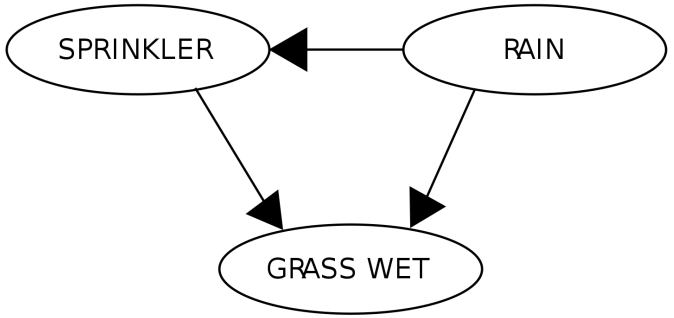
1. Given the conditional probabilities, we can update the state of network when certain variables are observed
2. Given observations (R,S,W) we can build a model for the probabilities

First case of ‘inference’

Given the probabilities, calculations are with rules probability straightforward albeit cumbersome

Linear algebra can make calculations very compact

RAIN	SPRINKLER T	SPRINKLER F
F	0.4	0.6
T	0.01	0.99



RAIN T	RAIN F
0.2	0.8

Example from https://en.wikipedia.org/wiki/Bayesian_network

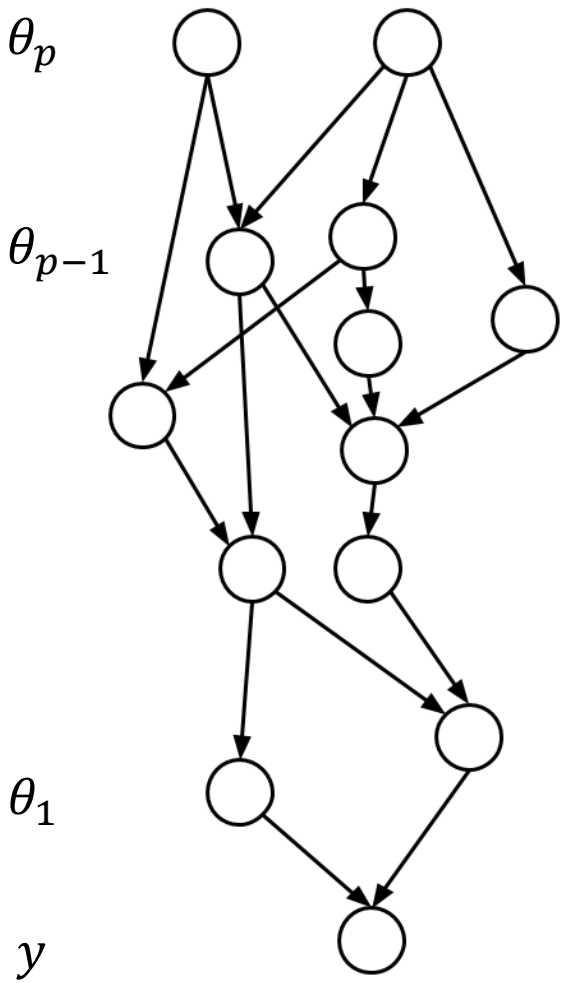
		GRASS WET T	GRASS WET F
SPRINKLER F	RAIN F	0.0	1.0
SPRINKLER F	RAIN T	0.8	0.2
SPRINKLER T	RAIN F	0.9	0.1
SPRINKLER T	RAIN T	0.99	0.01

$$\Pr(R = T \mid G = T) = \frac{\Pr(G = T, R = T)}{\Pr(G = T)} = \frac{\sum_{S \in \{T,F\}} \Pr(G = T, S, R = T)}{\sum_{S,R \in \{T,F\}} \Pr(G = T, S, R)}$$

$$\begin{aligned}\Pr(G = T, S = T, R = T) &= \Pr(G = T \mid S = T, R = T) \Pr(S = T \mid R = T) \Pr(R = T) \\ &= 0.99 \times 0.01 \times 0.2 \\ &= 0.00198.\end{aligned}$$

$$\Pr(R = T \mid G = T) = \frac{0.00198_{TTT} + 0.1584_{TFT}}{0.00198_{TTT} + 0.288_{TTF} + 0.1584_{TFT} + 0.0_{TFF}} \approx 35.77\%.$$

Inference on model parameters defining probability distributions



Think of data as the output of the graph – i.e. the distribution of variables you would expect to see under particular underlying probability distributions and relationships among those probability distributions

$$p(\theta|y) = \frac{p(y|\theta)p(\theta)}{p(y)} \propto p(y|\theta)p(\theta)$$

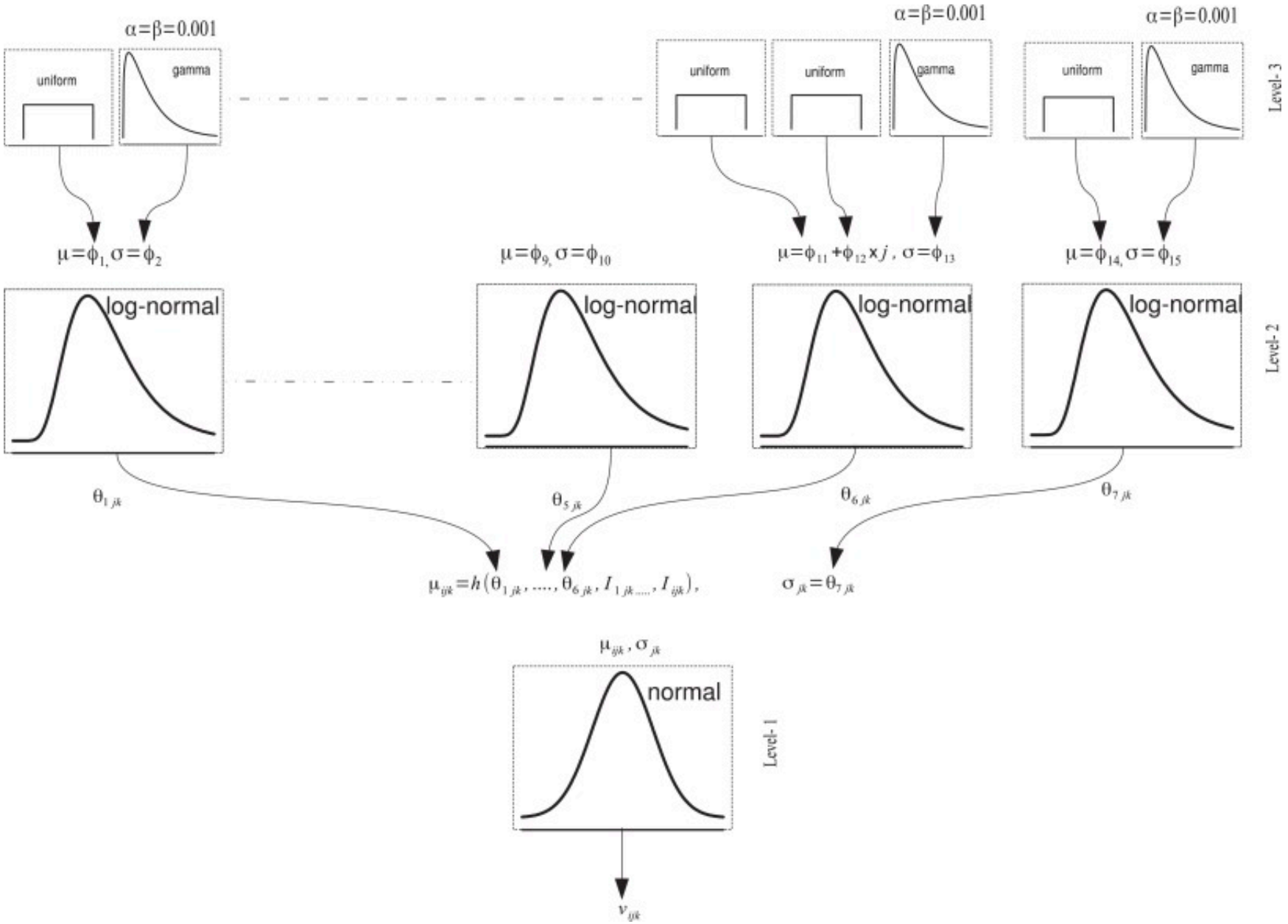
If the parameters have hierarchical network structure, we can expand

$$p(\theta_1, \theta_2, \dots, \theta_p|y) \propto p(y|\theta_1, \theta_2, \dots)p(\theta_1|\theta_2, \dots, \theta_p) \dots p(\theta_p)$$

Markov property: Each node has an independent distribution, conditional on the nodes above

= Bayesian hierarchical model

A nice picture from Google



Remarks

We want the posterior for the model parameters, conditional on observed data

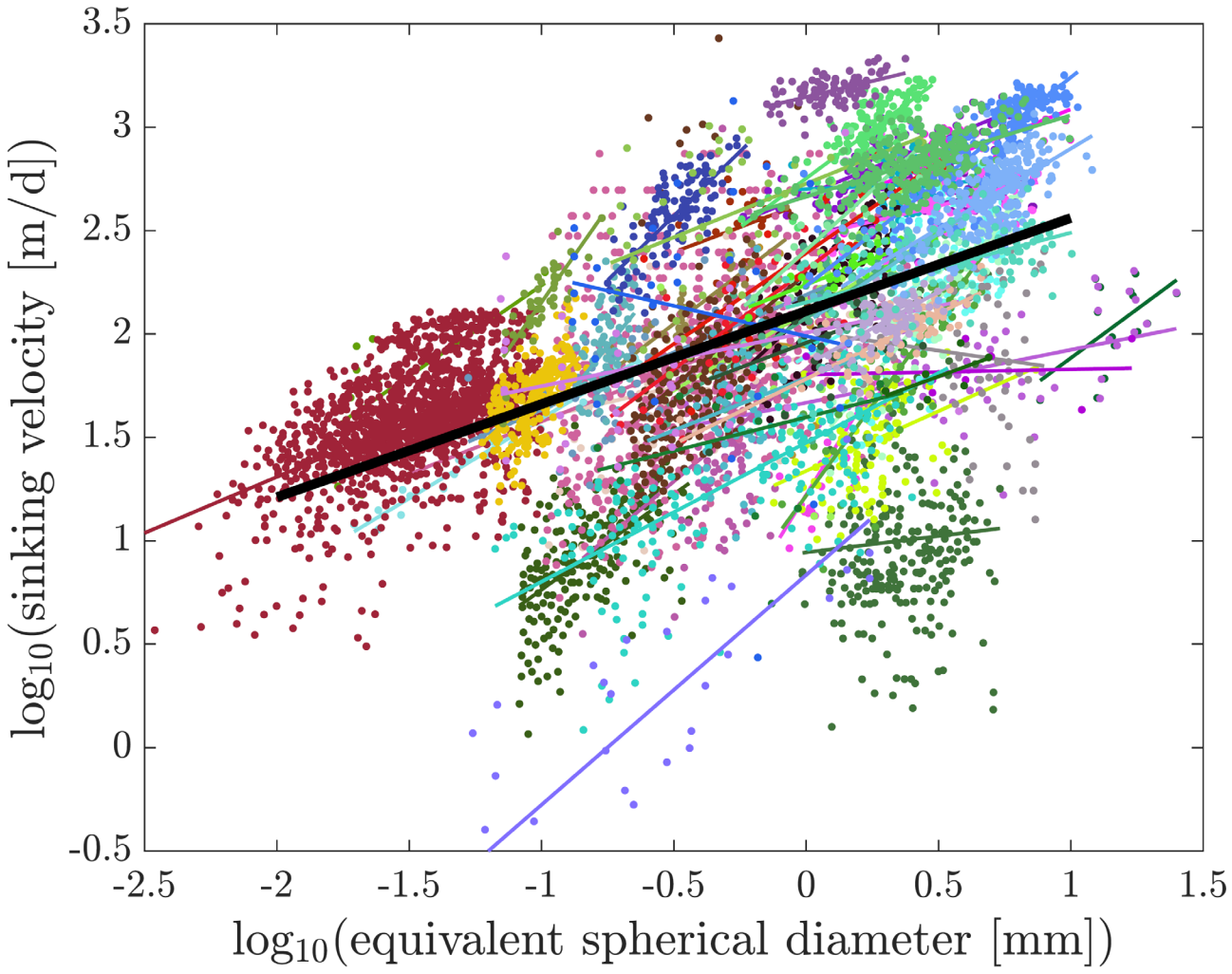
Difficulties:

- No analytical form for vast majority of posteriors
- We need to run samples on the graph to see what combination of transition probabilities is most consistent with the data
- Potentially very high dimensional space – hard to sample; vast majority of independent samples from prior will have negligible posterior probability
- Run correlated Markov Chains that move to and stay in regions of high probability (e.g. random walk Metropolis-Hastings) – STILL HARD!

Solution:

- Use sophisticated algorithms (e.g. Hamiltonian Monte Carlo) that perform high efficiency sampling across general classes of models -> Probabilistic programming languages

Real world example: Hierarchical linear regression on a Bayesian network



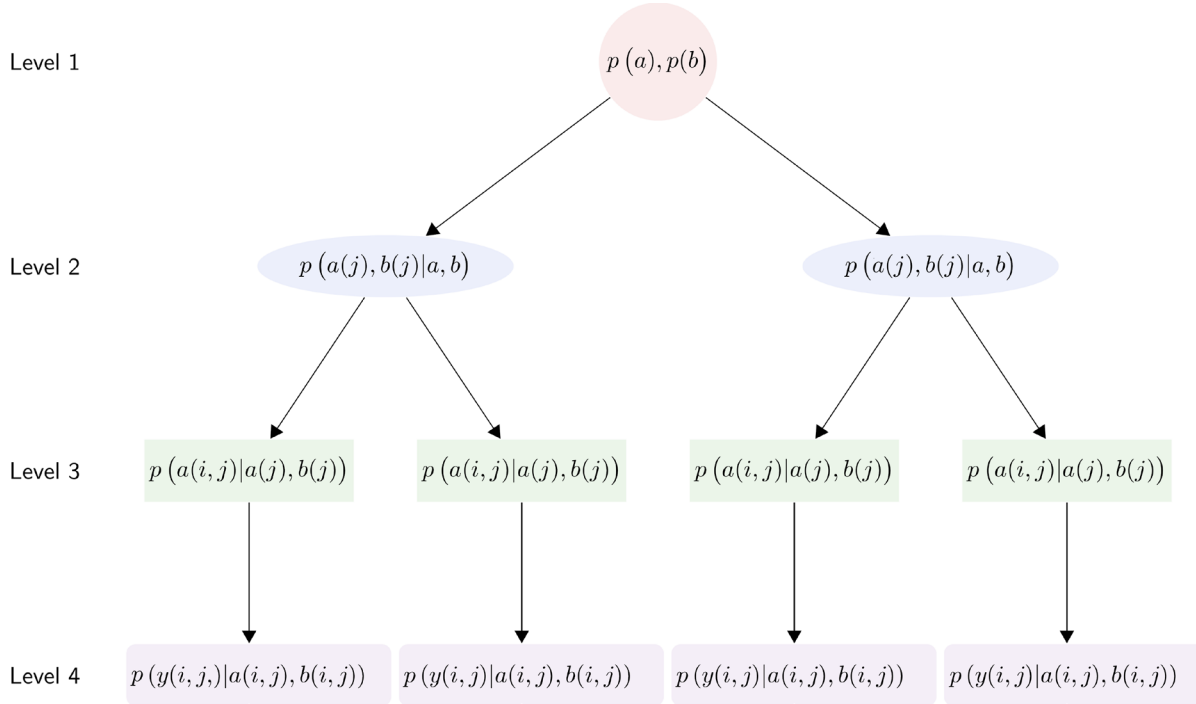
$$w = \left(\frac{4g}{3\rho_f}\frac{\Delta\rho}{C_D}d\right)^{\kappa} \qquad w = \alpha d^{\beta}$$

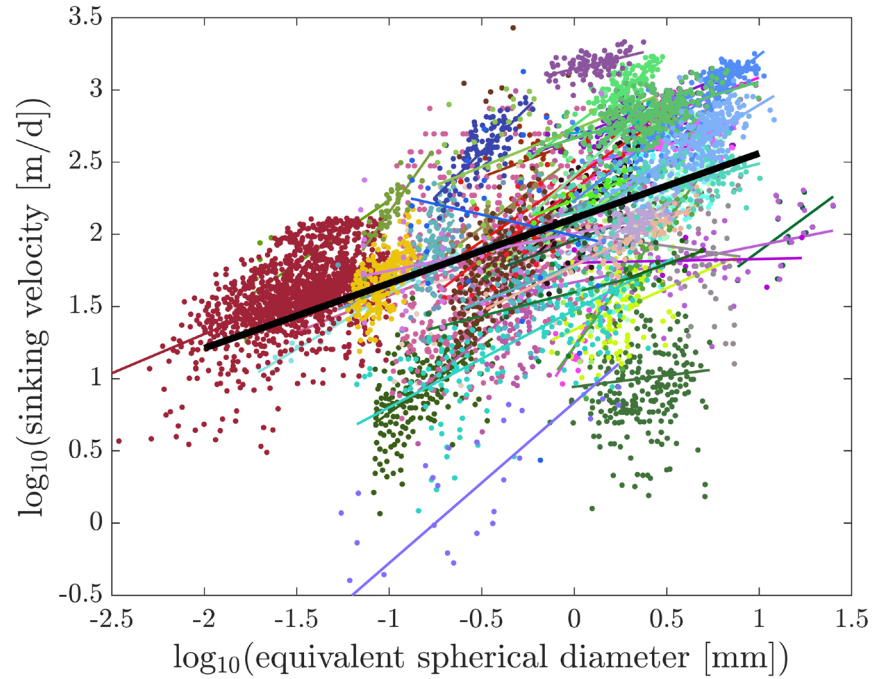
$$\log w = \log\alpha + \beta \log d$$

$$y = a + bx$$

$$y_{i,j} = a + a_j + a_{i|j} + (b + b_j + b_{i|j})x_{i,j}$$

Why not just fit a bunch of independent regressions?

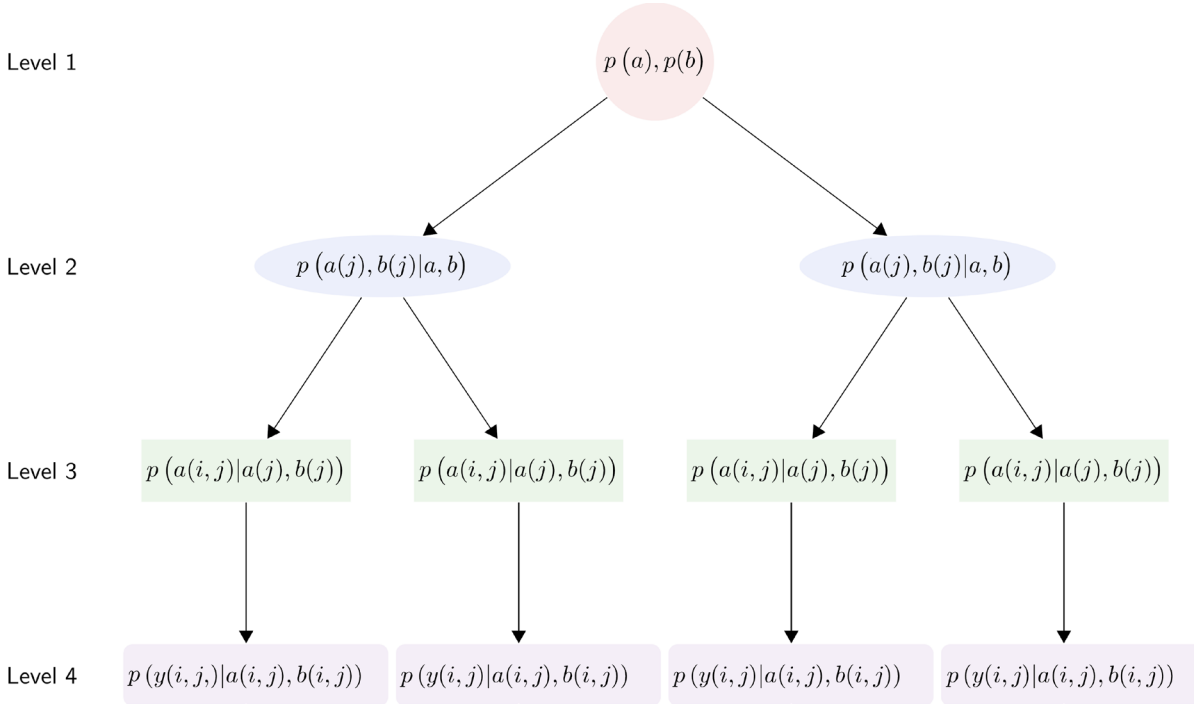


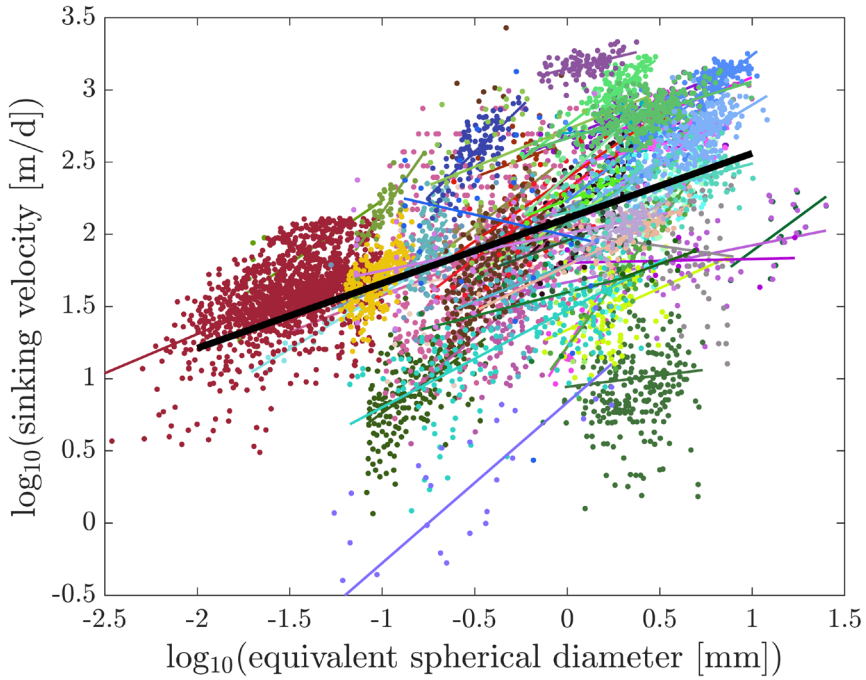


$$p(a_j|a) \sim N(a, \sigma^2_{a,j}) \qquad p(b_j|b) \sim N(b, \sigma^2_{b,j})$$

$$p(a_{i,j}|a_j) \sim N(a_j, \sigma^2_{a,i,j}) \qquad p(b_{i,j}|b_j) \sim N(b_j, \sigma^2_{b,i,j})$$

$$p(y_{i,j}|a_{i,j}, b_{i,j}) \sim N(a_{i,j} + b_{i,j}x_{i,j}, \sigma^2_y)$$





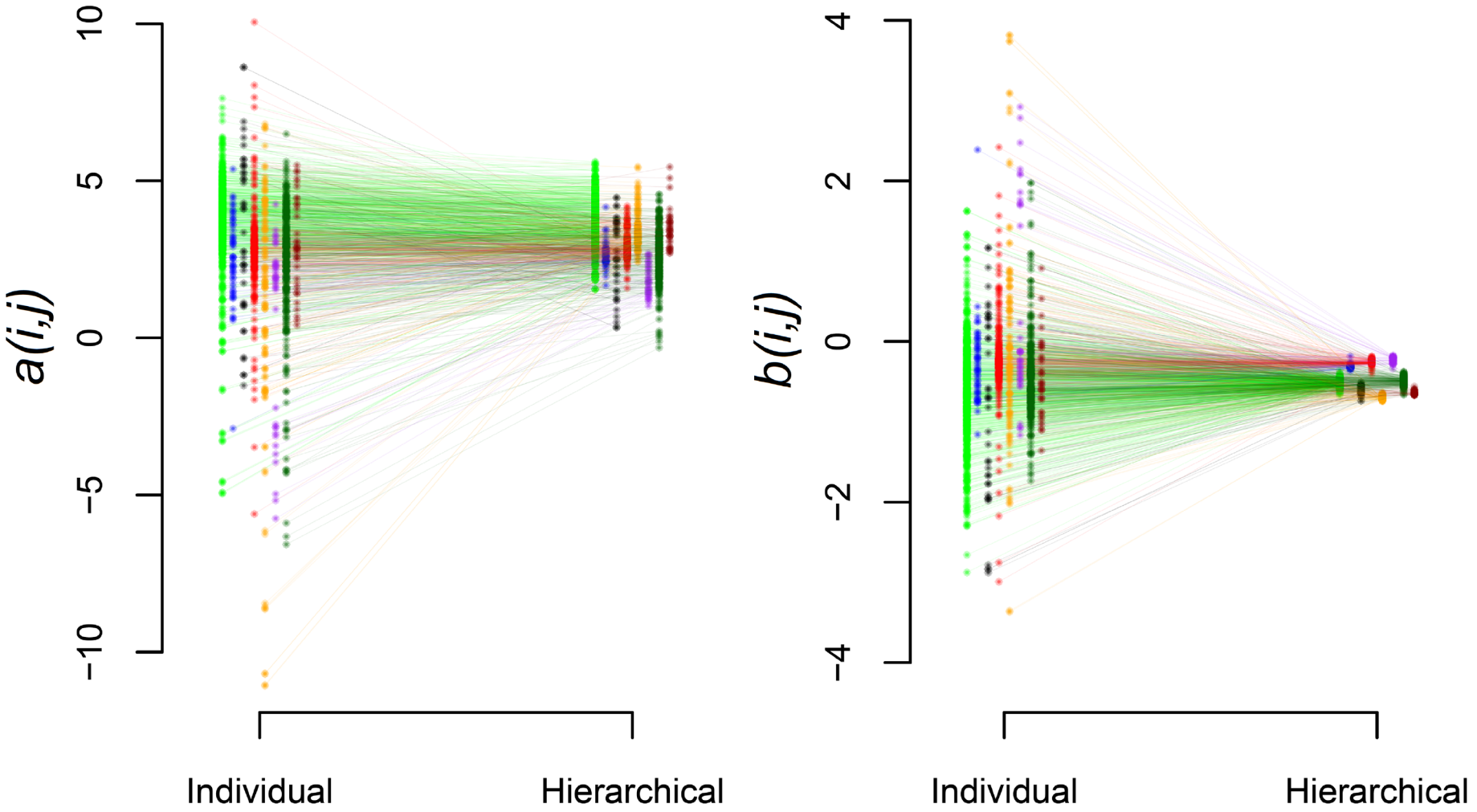
$$p(a_j|a) \sim N(a, \sigma^2_{a,j}) \qquad p(b_j|b) \sim N(b, \sigma^2_{b,j})$$

$$p(a_{i,j}|a_j) \sim N(a_j, \sigma^2_{a,i,j}) \qquad p(b_{i,j}|b_j) \sim N(b_j, \sigma^2_{b,i,j})$$

$$p(y_{i,j}|a_{i,j}, b_{i,j}) \sim N(a_{i,j} + b_{i,j}x_{i,j}, \sigma^2_y)$$

NOTE: Can fit this model without MCMC due to Gaussian assumptions, but general MCMC approach will fit more general models and distributions

Individual fits vs. hierarchical Bayesian network



Probabilistic programming

Probabilistic programming

From Wikipedia, the free encyclopedia

Probabilistic programming (**PP**) is a programming paradigm in which probabilistic models are specified and inference for these models is performed automatically.[1] It represents an attempt to unify probabilistic modeling and traditional general purpose programming in order to make the former easier and more widely applicable.[2][3] It can be used to create systems that help make decisions in the face of uncertainty.

Programming languages used for probabilistic programming are referred to as "Probabilistic programming languages" (PPLs).

"Automatic": You specify the model and functional forms and supply data and the software will compute the posteriors

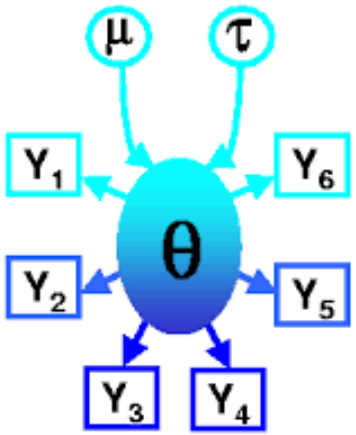
Since ~10 years, most posteriors encountered in practice were too hard to compute (no closed forms without extremely strong approximation, MCMC sampling would take forever)

Led to a discounting of fully Bayesian analysis

Algorithmic developments, in particular the theory and application of *Hamiltonian Monte Carlo* has made Bayesian *extremely feasible for a wide class of research problems*

Led to the development of abstracted 'automatic systems' for inference

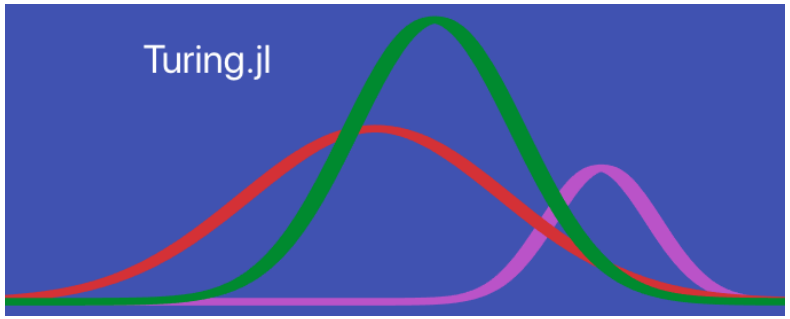
Probabilistic programming



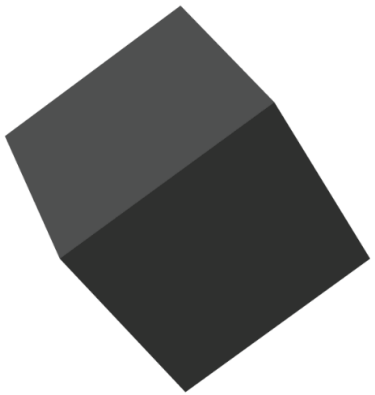
BUGS

Stan

Gen.jl



Edward



NOTEBOOK

Summary

Graphical models are a convenient way to characterize conditional/causal relationships

Bayesian graphs have a particular structure where conditional probability flows directionally from prior distributions

Many datasets have natural hierarchical structure that is well modeled by Bayesian graphs (i.e. ‘Bayesian hierarchical model’)

Probabilistic programming languages now available to make general Bayesian graphical inference orders of magnitude easier than just five years ago…