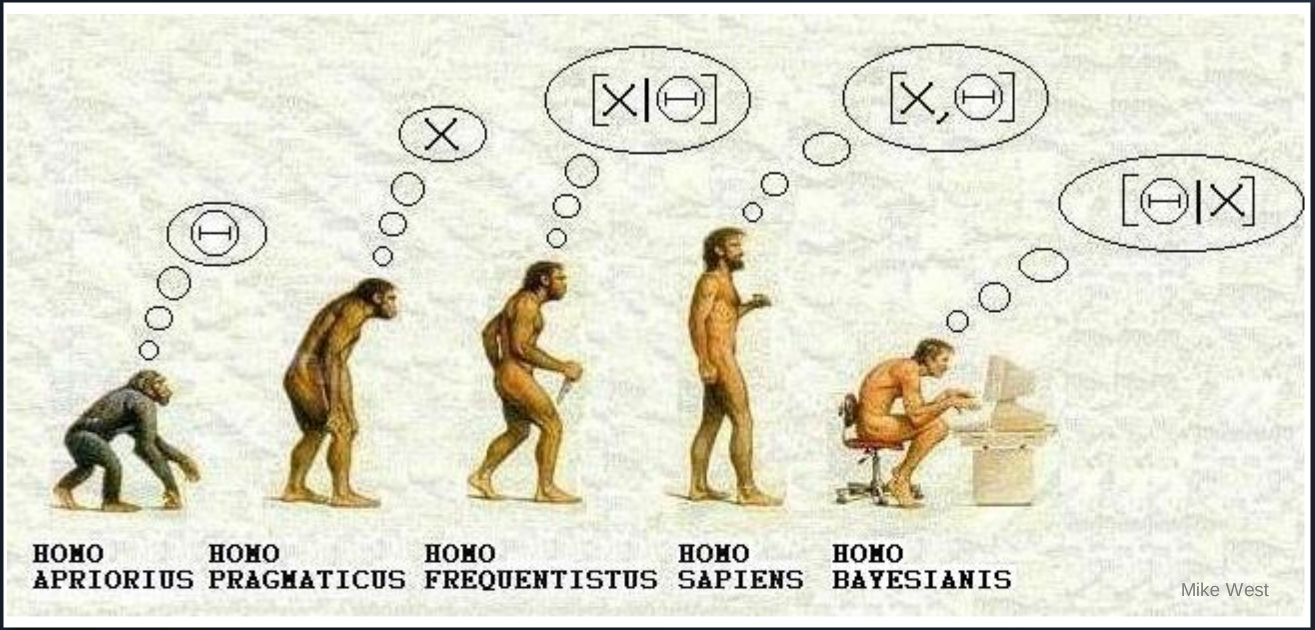
Bayesian inversion of deterministic dynamic causal models

Kay H. Brodersen[1,2] & Ajita Gupta[2]

[1] Department of Economics, University of Zurich, Switzerland
[2] Department of Computer Science, ETH Zurich, Switzerland



Mike West

Overview

1. **Problem setting**

   Model; likelihood; prior; posterior.

2. **Variational Laplace**

   Factorization of posterior; why the free-energy; energies; gradient ascent; adaptive step size; example.

3. **Sampling**

   Transformation method; rejection method; Gibbs sampling; MCMC; Metropolis-Hastings; example.

4. **Model comparison**

   Model evidence; Bayes factors; free-energy; prior arithmetic mean; posterior harmonic mean; Savage-Dickey; example.

With material from Will Penny, Klaas Enno Stephan, Chris Bishop, and Justin Chumbley.

1 Problem setting

Model = likelihood + prior

y	data
m	model
θ	model parameters

$p(y\|\theta, m)$	likelihood
$p(\theta\|m)$	prior

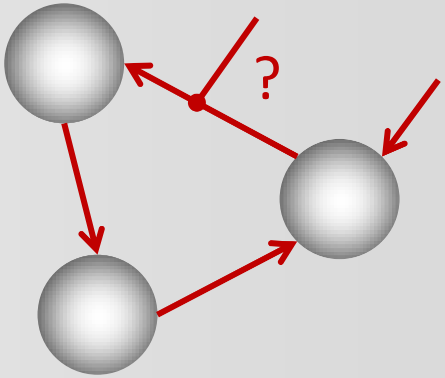
$p(\theta\|y, m)$	posterior
$p(y\|m)$	model evidence

Bayesian inference is conceptually straightforward

Question 1: what do the data tell us about the model parameters?

⇨ compute the posterior

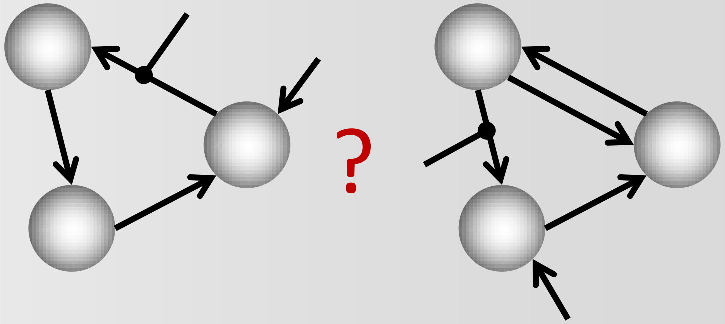
$$p(\theta|y,m) = \frac{p(y|\theta,m)p(\theta|m)}{p(y|m)}$$



Question 2: which model is best?

⇨ compute the model evidence

$$\begin{aligned} p(m|y) &\propto p(y|m)p(m) \\ &= \int p(y|\theta,m)p(\theta|m)d\theta \end{aligned}$$



2 Variational Laplace

Variational Laplace in a nutshell

❶ Neg. free-energy approx. to model evidence.

$$\ln p(y|m) = F + KL\left[q(\theta,\lambda), p(\theta,\lambda|y)\right]$$

$$F = \left\langle \ln p(y,\theta,\lambda)\right\rangle_q - KL\left[q(\theta,\lambda), p(\theta,\lambda|m)\right]$$

❷ Mean field approx.

$$p(\theta,\lambda|y) \approx q(\theta,\lambda) = q(\theta)q(\lambda)$$

❸ Maximise neg. free energy wrt. q = minimise divergence, by maximising variational energies

$$q(\theta) \propto \exp(I_\theta) = \exp\left[\left\langle \ln p(y,\theta,\lambda)\right\rangle_{q(\lambda)}\right]$$

$$q(\lambda) \propto \exp(I_\lambda) = \exp\left[\left\langle \ln p(y,\theta,\lambda)\right\rangle_{q(\theta)}\right]$$

❹ Iterative updating of sufficient statistics of approx. posteriors by gradient ascent.

Variational Laplace

Assumptions

$$q(\theta, \lambda | y, m) = q(\theta | y, m) q(\lambda | y, m)$$

mean-field approximation

$$\left.\begin{aligned} q(\theta | y, m) &= \mathrm{N}(\theta; m_\theta, S_\theta) \\ q(\lambda | y, m) &= \mathrm{N}(\lambda; m_\lambda, S_\lambda) \end{aligned}\right\}$$

Laplace approximation

Variational Laplace

Inversion strategy

Recall the relationship between the log model evidence and the negative free-energy F:

$$\ln p(y|m) = \underbrace{E_q[\ln p(y|\theta)] - KL[q(\theta)|p(\theta|m)]}_{=:\,F} + \underbrace{KL[q(\theta)||p(\theta|y,m)]}_{\geq 0}$$

Maximizing F implies two things:

(i) we obtain a good approximation to $\ln p(y|m)$

(ii) the KL divergence between $q(\theta)$ and $p(\theta|y,m)$ becomes minimal

Practically, we can maximize F by iteratively (EM) maximizing the variational energies:

$$\begin{aligned} I(\theta) &= \int L(\theta,\lambda) q(\lambda) \\ I(\lambda) &= \int L(\theta,\lambda) q(\theta) \end{aligned}$$

W. Penny

Variational Laplace

Implementation: gradient-ascent scheme (Newton's method)

Newton's Method for finding a root (1D)

$$x(new) = x(old) - \frac{f(x(old))}{f'(x(old))}$$

Compute gradient vector

$$j_\theta(i) = \frac{\partial I(\theta)}{\partial \theta(i)}$$

Compute curvature matrix

$$H_\theta(i,j) = \frac{\partial^2 I(\theta)}{\partial \theta(i) \partial \theta(j)}$$

Variational Laplace

Implementation: gradient-ascent scheme (Newton’s method)

Compute Newton update (change)

$$\Delta m_\theta = -H_\theta^{-1} j_\theta$$

New estimate

$$m_\theta(new) = m_\theta(old) + \Delta m_\theta$$

Big curvature -> small step

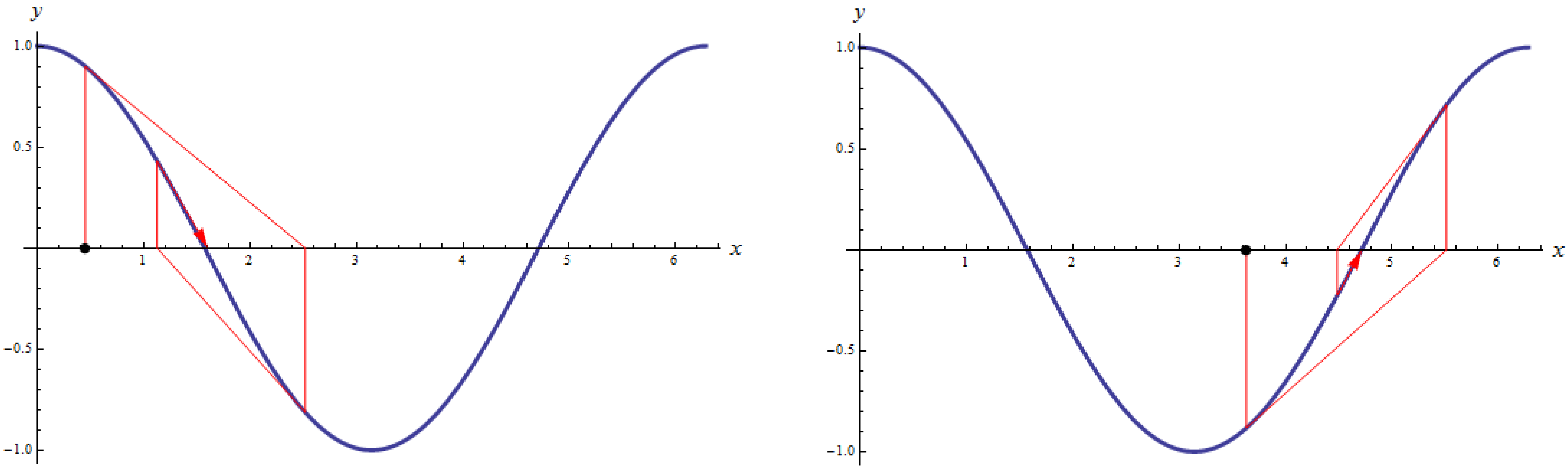
Small curvature -> big step

Newton’s Method for finding a root (1D)

$$x(new) = x(old) - \frac{f(x(old))}{f'(x(old))}$$

W. Penny

Newton’s method – demonstration



Newton’s method is very efficient. However, its solution is not insensitive to the starting point, as shown above.

http://demonstrations.wolfram.com/LearningNewtonsMethod/

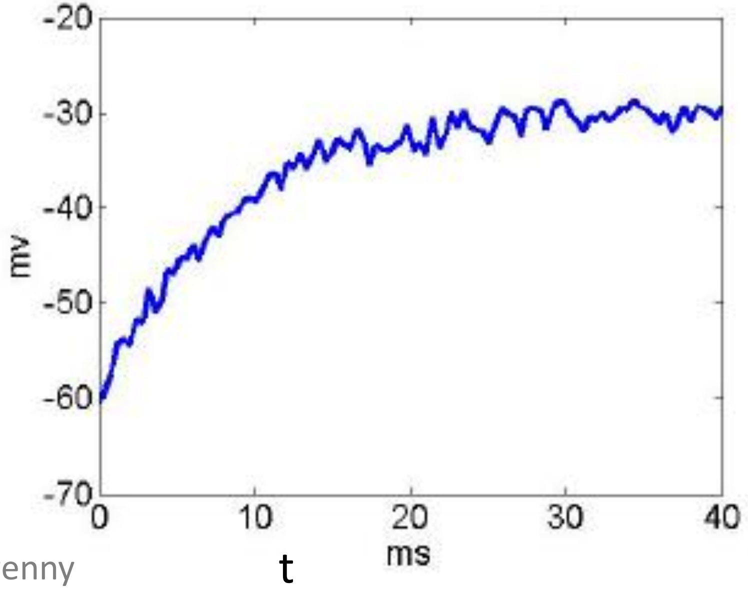
Variational Laplace

Nonlinear regression (example)

Model (likelihood):

$$y(t) = -60 + V_a[1 - \exp(-t/\tau)] + e(t)$$

Data:



Ground truth

(known parameter values that were used to generate the data on the left):

$$V_a = 30, \tau = 8, \exp(\lambda) = 1$$

where

$$p(y|\theta, \lambda, m) = \mathrm{N}(y; g(\theta, m), C_y)$$

$$C_y^{-1} = \sum_i \exp(\lambda_i) Q_i$$

W. Penny

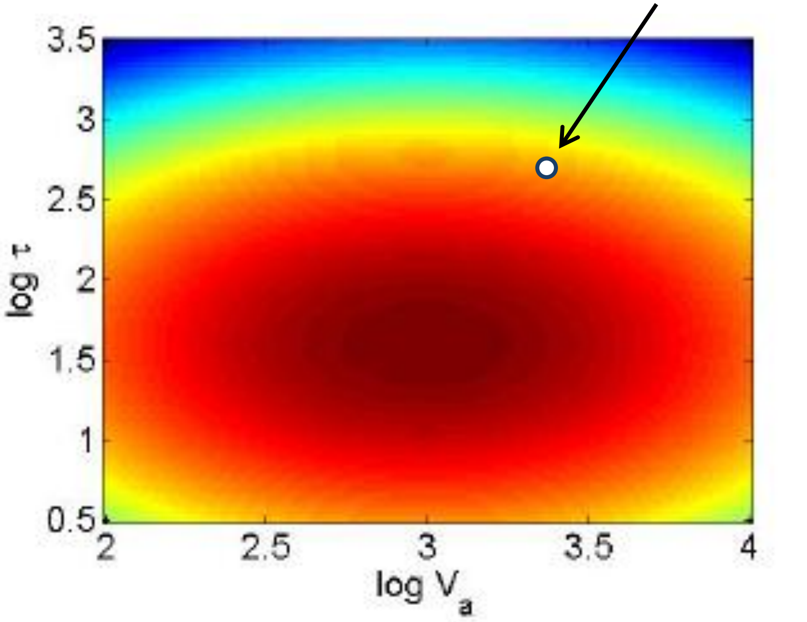
Nonlinear regression (example)

We begin by defining our prior:

$$\mu_\theta = [3, 1.6]^T, C_\theta = diag([1/16, 1/16]);$$

$$\mu_\lambda = 0, C_\lambda = 1/16$$

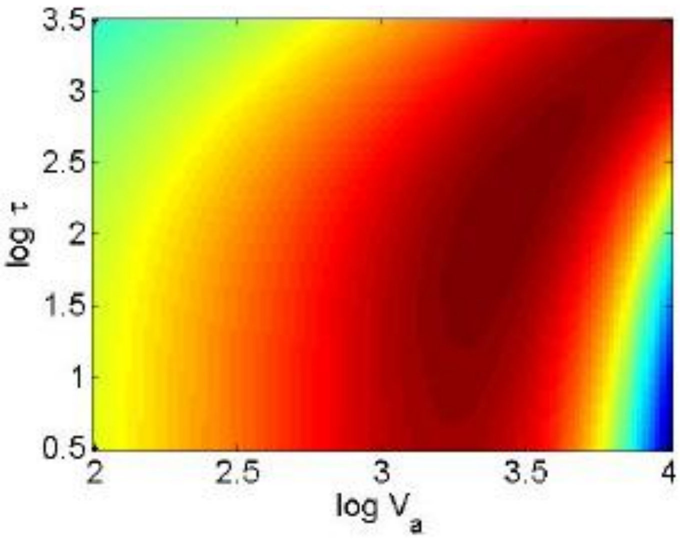
Prior density ($\log p(\theta)$)



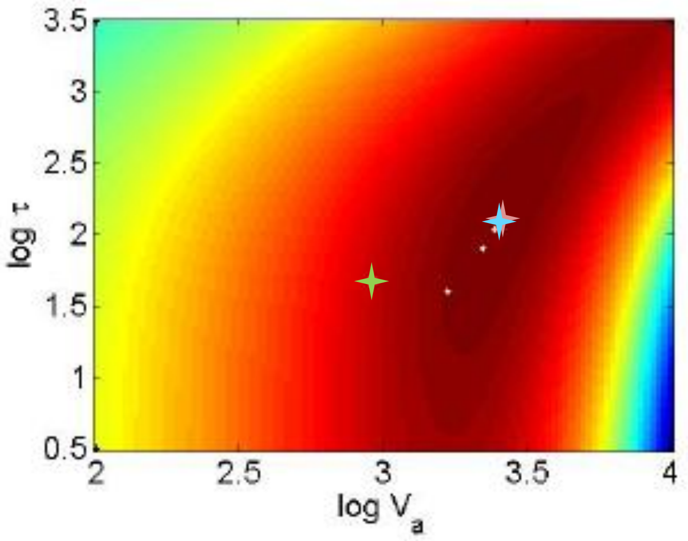
Variational Laplace

Nonlinear regression (example)

Posterior density ($\log[p(y|\theta)p(\theta)]$)



VL optimization (4 iterations)



✦ Starting point (2.9, 1.65)

✦ True value (3.4012, 2.0794)

✦ VL estimate (3.4, 2.1)

W. Penny

3 Sampling

Sampling

Deterministic approximations

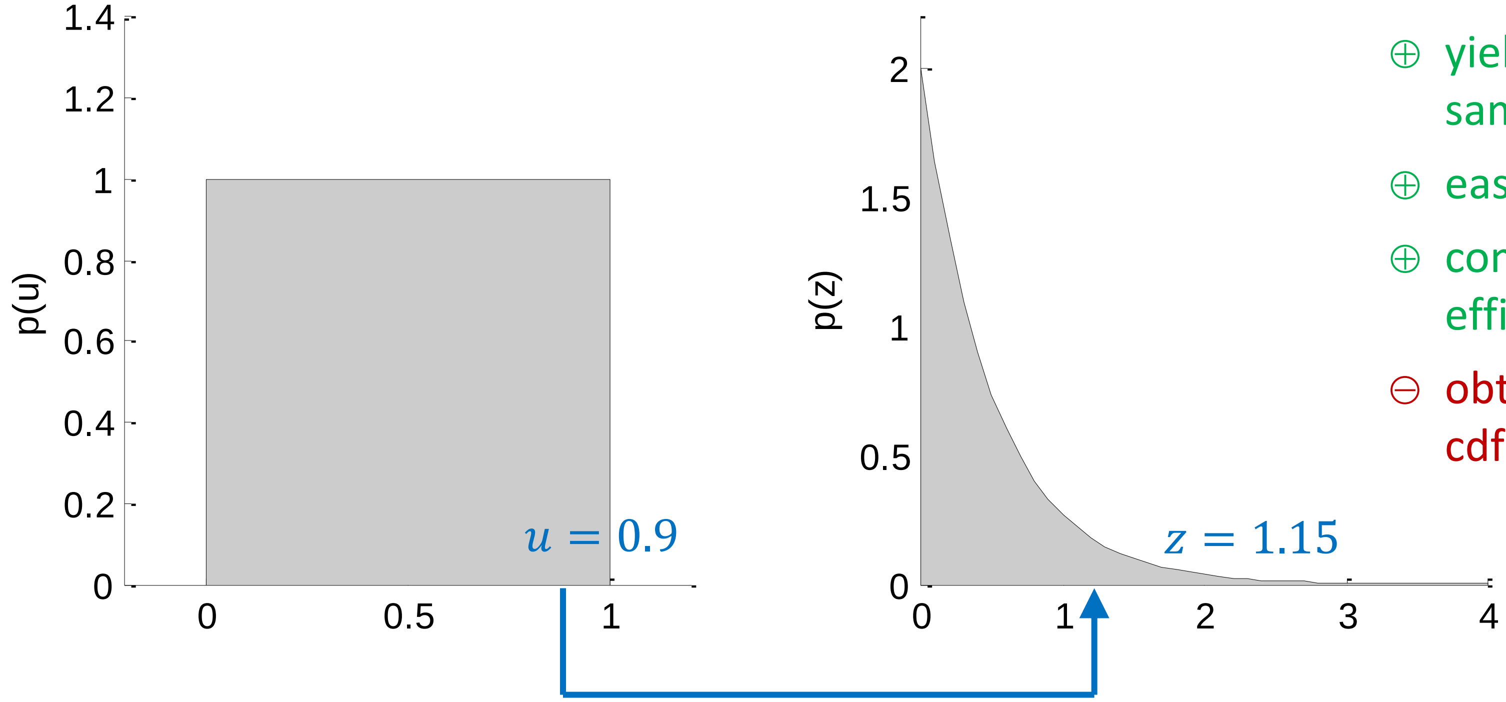
- ⊕ computationally efficient
- ⊕ efficient representation
- ⊕ learning rules may give additional insight
- ⊖ application initially involves hard work
- ⊖ systematic error

Stochastic approximations

- ⊕ asymptotically exact
- ⊕ easily applicable general-purpose algorithms
- ⊖ computationally expensive
- ⊖ storage intensive

Strategy 1 – Transformation method

We can obtain samples from some distribution $p(z)$ by first sampling from the uniform distribution and then *transforming* these samples.

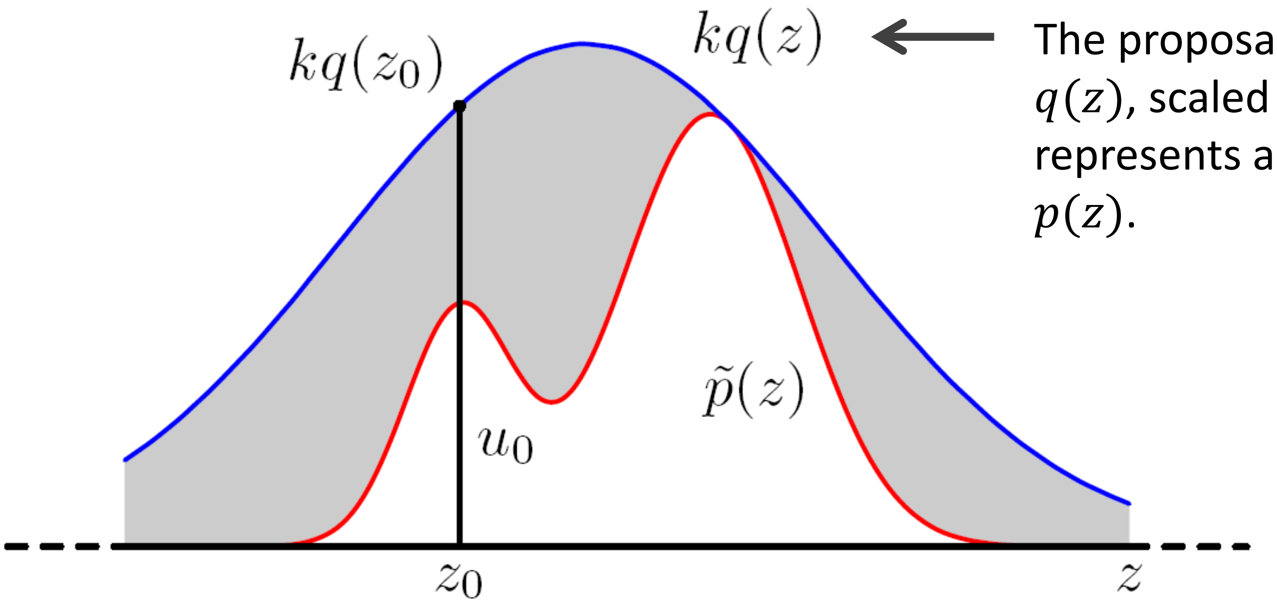


transformation: $z^{(\tau)} = F^{-1}\left(u^{(\tau)}\right)$

- ⊕ yields high-quality samples
- ⊕ easy to implement
- ⊕ computationally efficient
- ⊖ obtaining the inverse cdf can be difficult

Strategy 2 – Rejection method

When the transformation method cannot be applied, we can resort to a more general method called *rejection sampling*. Here, we draw random numbers from a simpler *proposal distribution* $q(z)$ and keep only some of these samples.



← The proposal distribution $q(z)$, scaled by a factor k, represents an envelope of $p(z)$.

⊕ yields high-quality samples

⊕ easy to implement

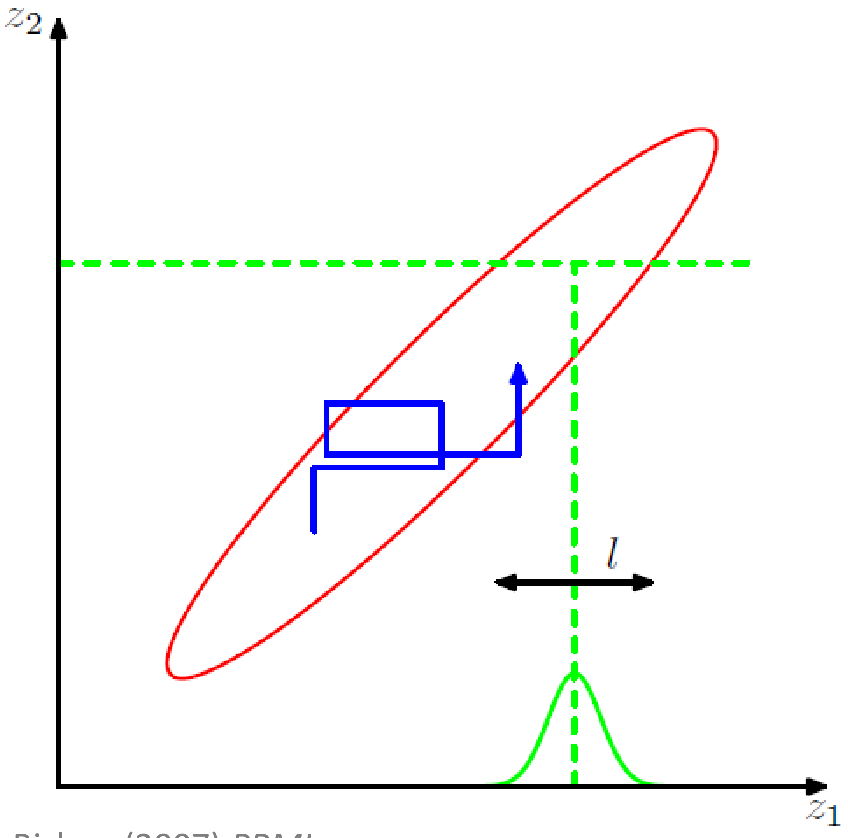
⊕ can be computationally efficient

⊖ computationally inefficient if proposal is a poor approximation

Bishop (2007) *PRML*

Strategy 3 – Gibbs sampling

Often the joint distribution of several random variables is unavailable, whereas the full-conditional distributions are available. In this case, we can cycle over full-conditionals to obtain samples from the joint distribution.



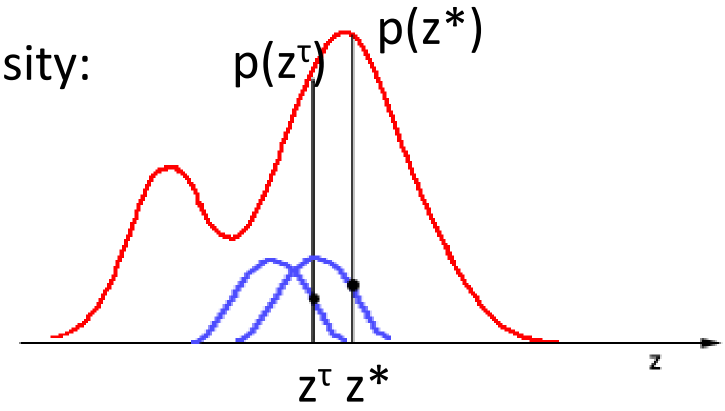
Bishop (2007) *PRML*

- ⊕ easy to implement
- ⊖ samples are serially correlated
- ⊖ the full-conditions may not be available

Strategy 4 – Markov Chain Monte Carlo (MCMC)

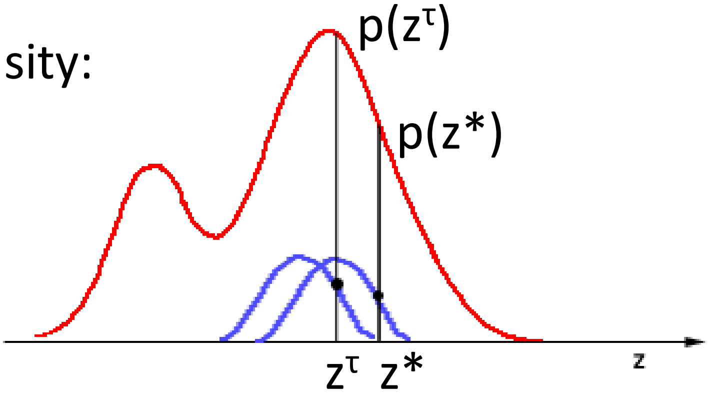
Idea: we can sample from a large class of distributions and overcome the problems that previous methods face in high dimensions using a framework called *Markov Chain Monte Carlo*.

Increase in density:



$$\alpha = 1$$

Decrease in density:

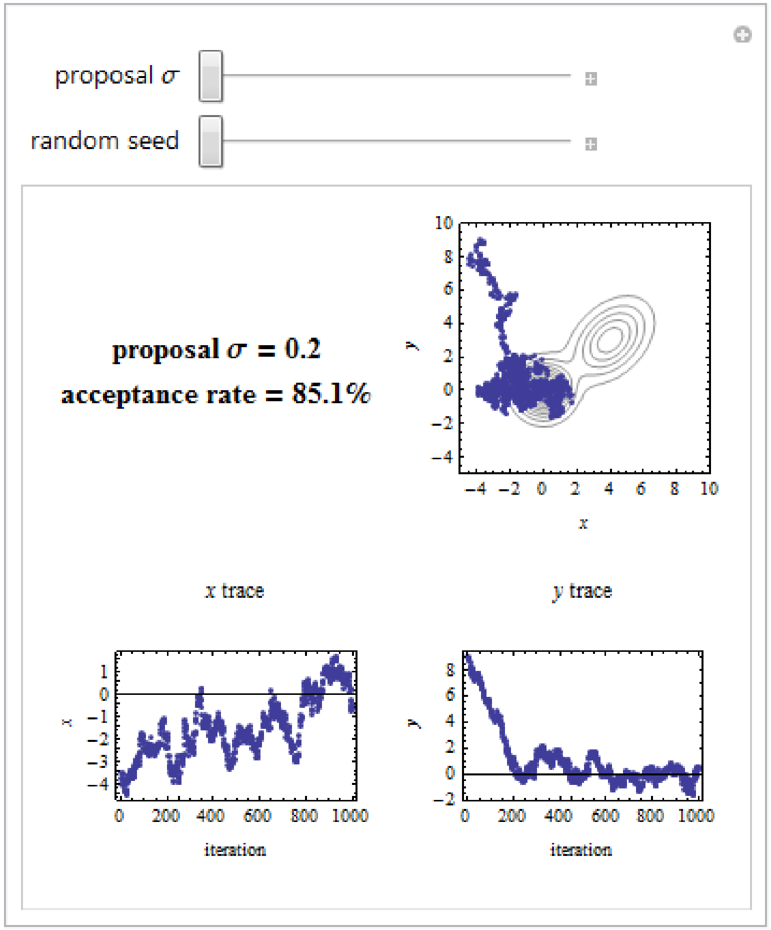


$$\alpha = \frac{\tilde{p}(z^*)}{\tilde{p}(z^\tau)}$$

M. Dümcke

MCMC demonstration: finding a good proposal density

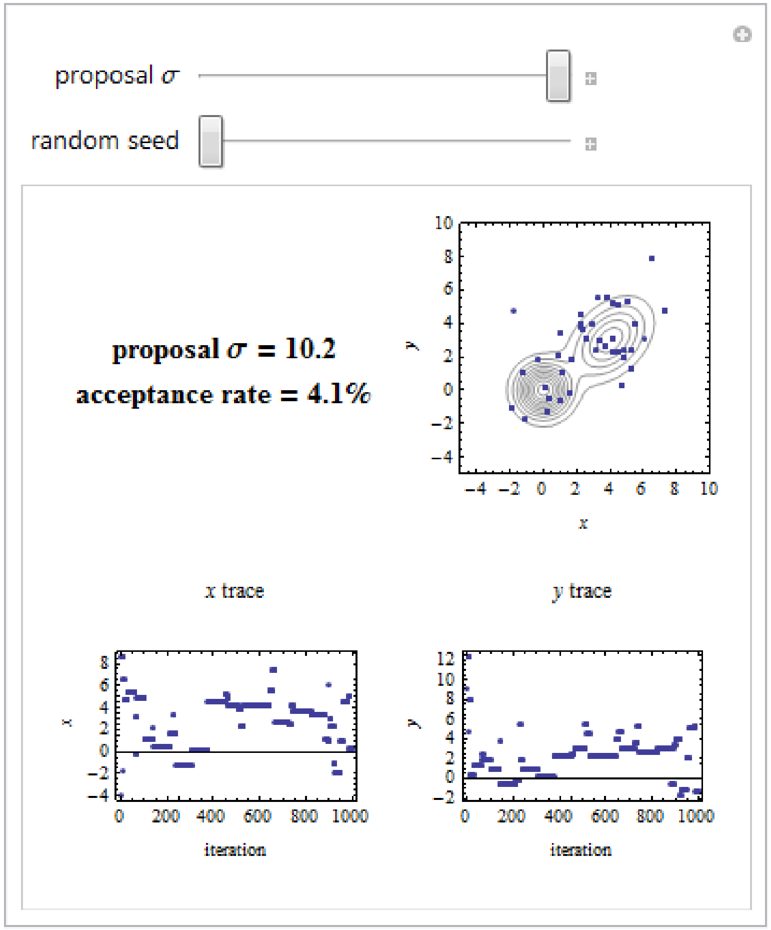
When the proposal distribution is too narrow, we might miss a mode.



proposal σ = 0.2

acceptance rate = 85.1%

When it is too wide, we obtain long constant stretches without an acceptance.



proposal σ = 10.2

acceptance rate = 4.1%

http://demonstrations.wolfram.com/MarkovChainMonteCarloSimulationUsingTheMetropolisAlgorithm/

MCMC for DCM

MH creates as series of random points $\theta^{(1)}, \theta^{(2)}, \ldots,$ whose distribution converges to the target distribution of interest. For us, this is the posterior density $p(\theta|y)$.

We could use the following proposal distribution:

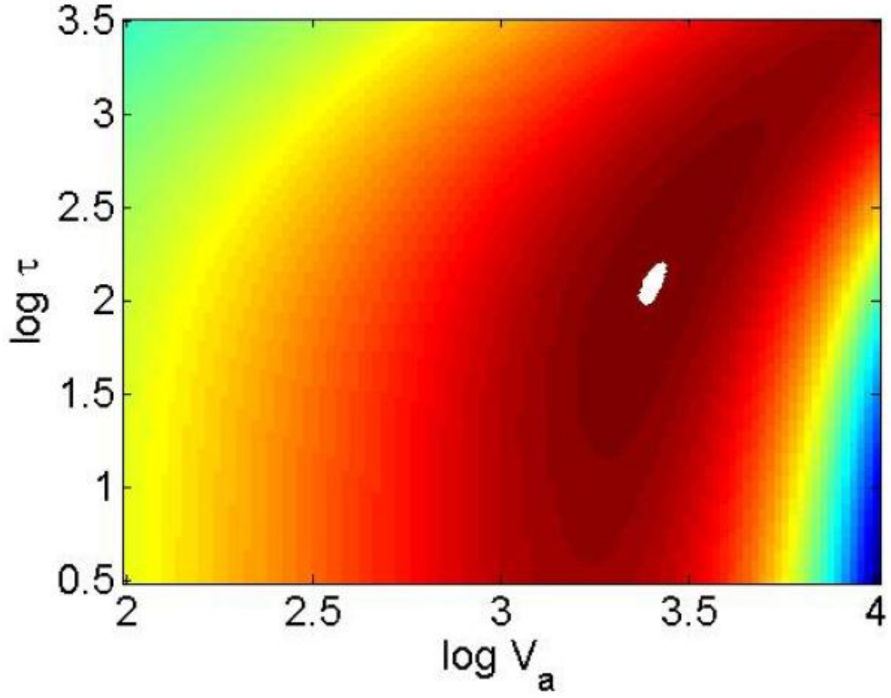
$$q\left(\theta^{(\tau)}\middle|\theta^{(\tau-1)}\right) = \mathcal{N}(0, \sigma C_{\theta})$$

prior covariance

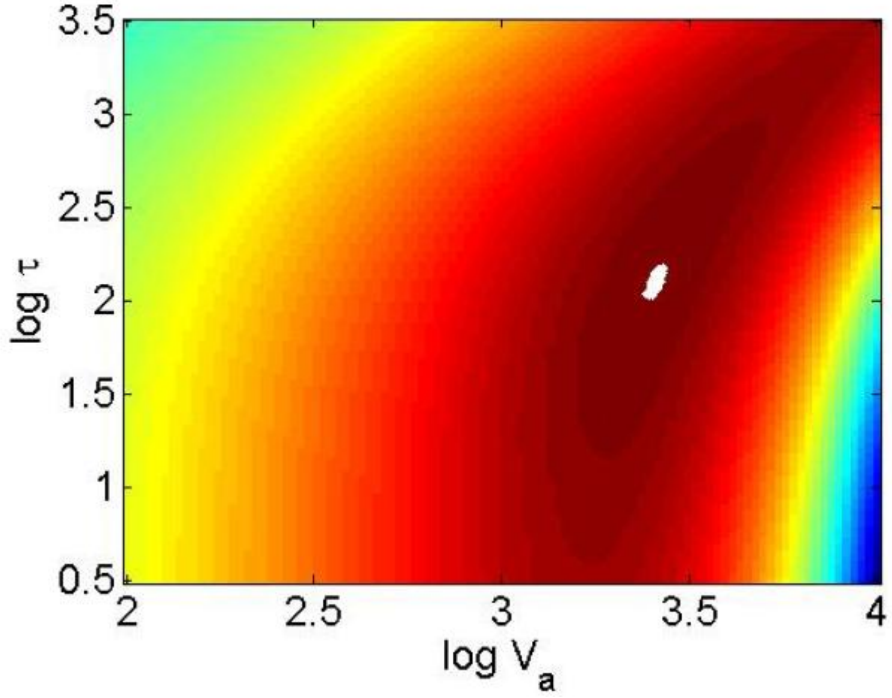
scaling factor
adapted such that acceptance rate is between 20% and 40%

MCMC for DCM

64,000 samples from MH posterior

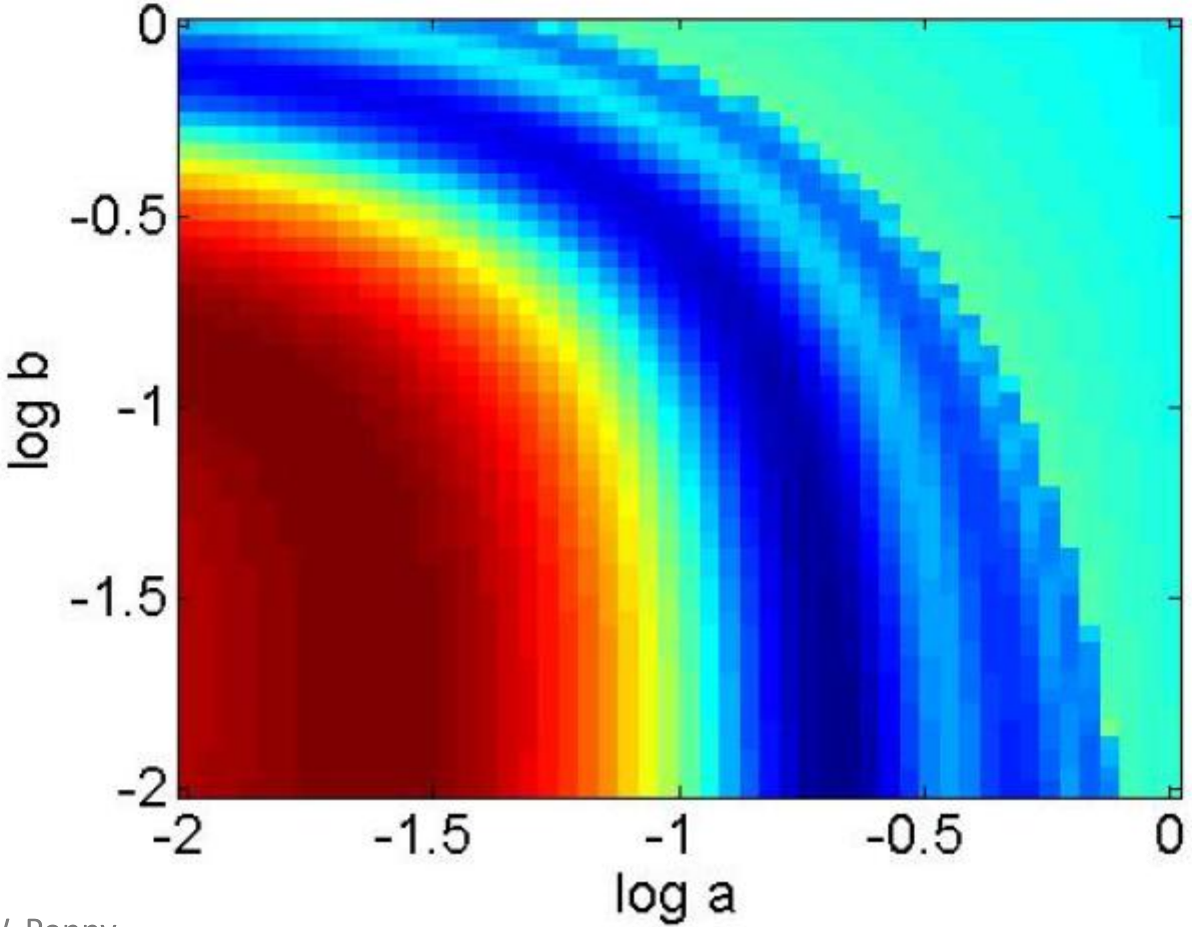


64,000 samples from VL posterior



MCMC – example

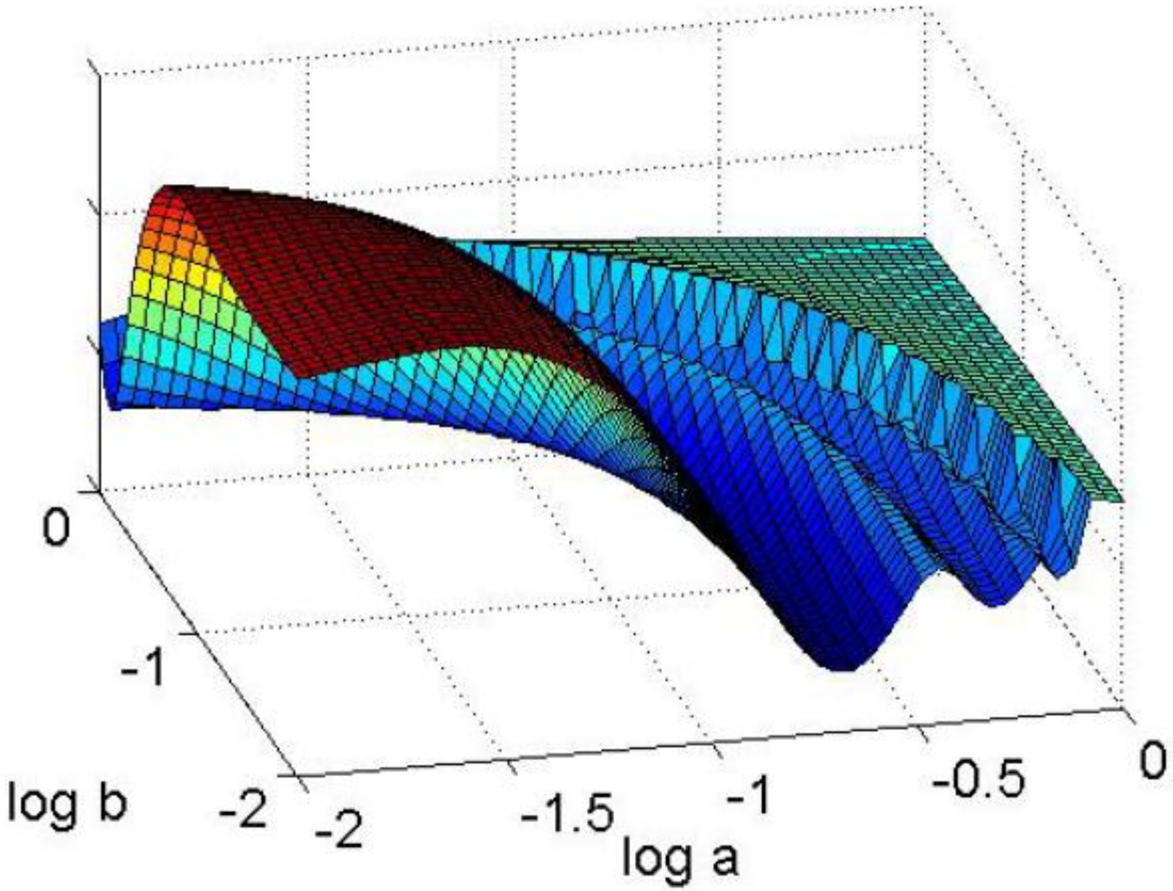
A plot of $\log[p(y|\theta)p(\theta)]$



W. Penny

MCMC – example

A plot of $\log[p(y|\theta)p(\theta)]$

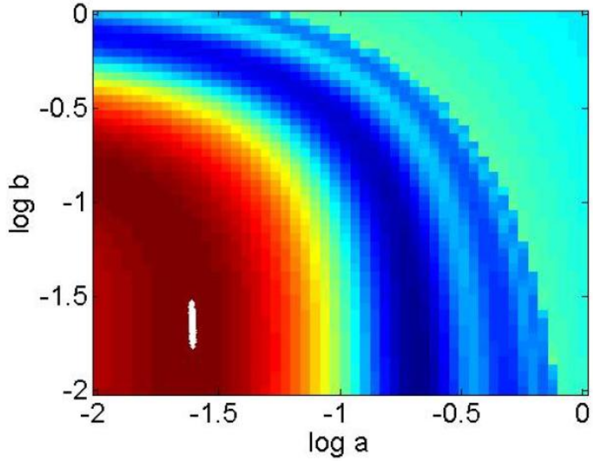


W. Penny

MCMC – example

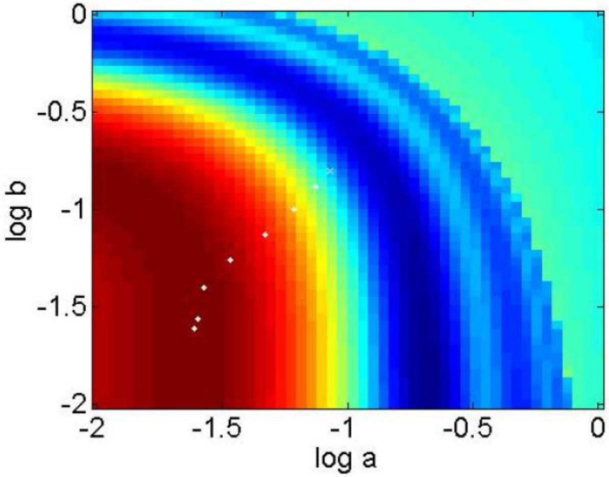
Metropolis-Hastings

2000 samples

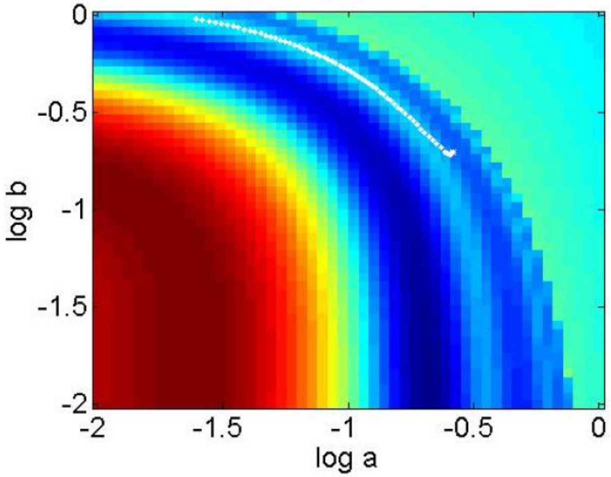


Variational-Laplace

Global maxima



Local maxima



4 Model comparison

Model evidence

The model evidence is not straightforward to compute, since this computation involves integrating out the dependence on model parameters

$$p(y|m) = \int p(y|\theta, m)p(\theta|m)d\theta.$$

Once computed two models can be compared via the Bayes factor

$$B_{12} = \frac{p(y|m_1)}{p(y|m_2)}$$

Prior arithmetic mean

The simplest approximation to the model evidence

$$p(y|m) = \int p(y|\theta, m)p(\theta|m)d\theta.$$

is the Prior Arithmetic Mean

$$p_{PAM}(y|m) = \frac{1}{S}\sum_{s=1}^{S} p(y|\theta_s, m)$$

where the samples θ_s are drawn from the prior density.

A problem with this estimate is that most samples from the prior will have low likelihood. A large number of samples will therefore be required to ensure that high likelihood regions of parameter space will be included in the average.

Posterior harmonic mean

A second option is the Posterior Harmonic Mean

$$p_{PHM}(y|m) = \left[\frac{1}{S} \sum_{s=1}^{S} \frac{1}{p(y|\theta_s, m)} \right]^{-1}$$

where samples are drawn from the posterior (eg. through MH sampling).

A problem with the PHM is that the largest contributions come from low likelihood samples which results in a high-variance estimator.

Savage-Dickey ratio

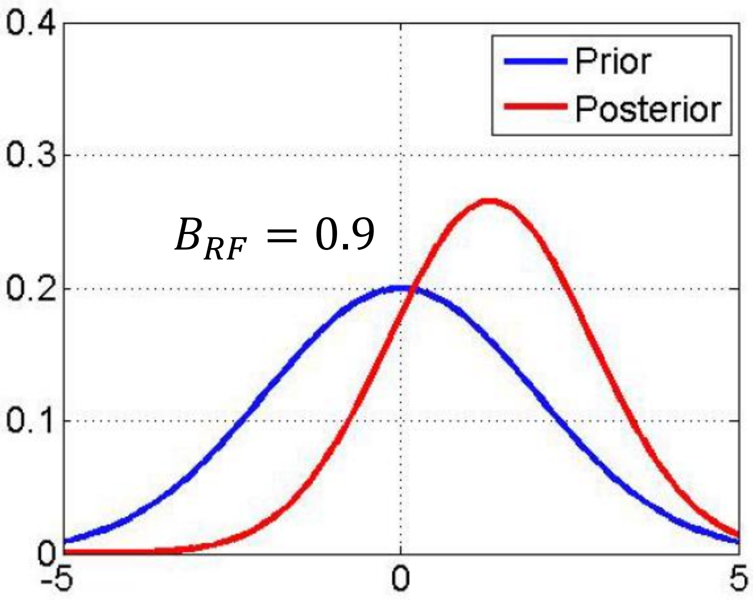
In many situations we wish to compare models that are nested. For example:

m_F: full model with parameters $\theta = (\theta_1, \theta_2)$

m_R: reduced model with $\theta = (\theta_1, 0)$

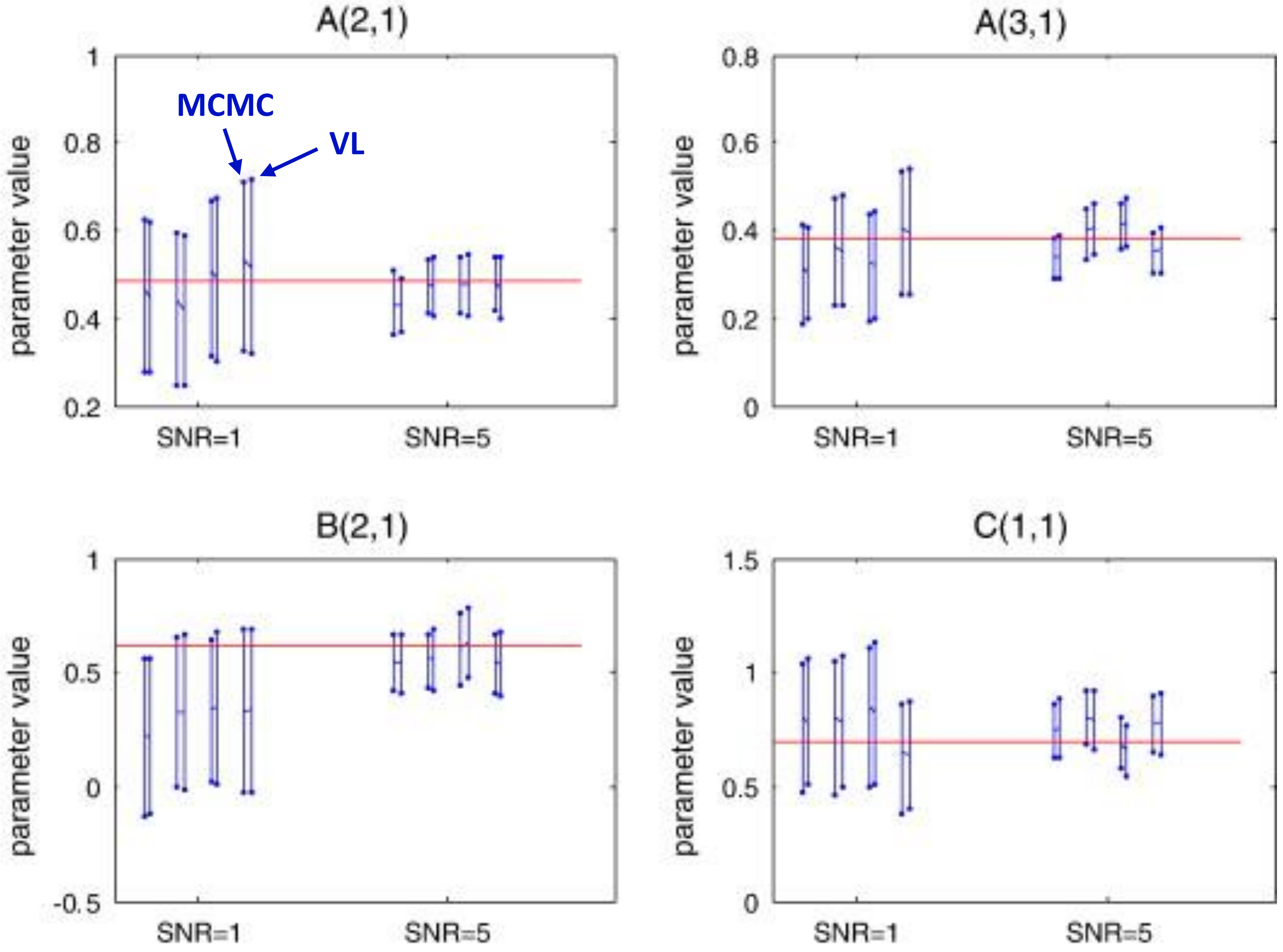
In this case, we can use the Savage-Dickey ratio to obtain a Bayes factor without having to compute the two model evidences:

$$B_{RF} = \frac{p(\theta_2 = 0|y, m_F)}{p(\theta_2 = 0|m_F)}$$



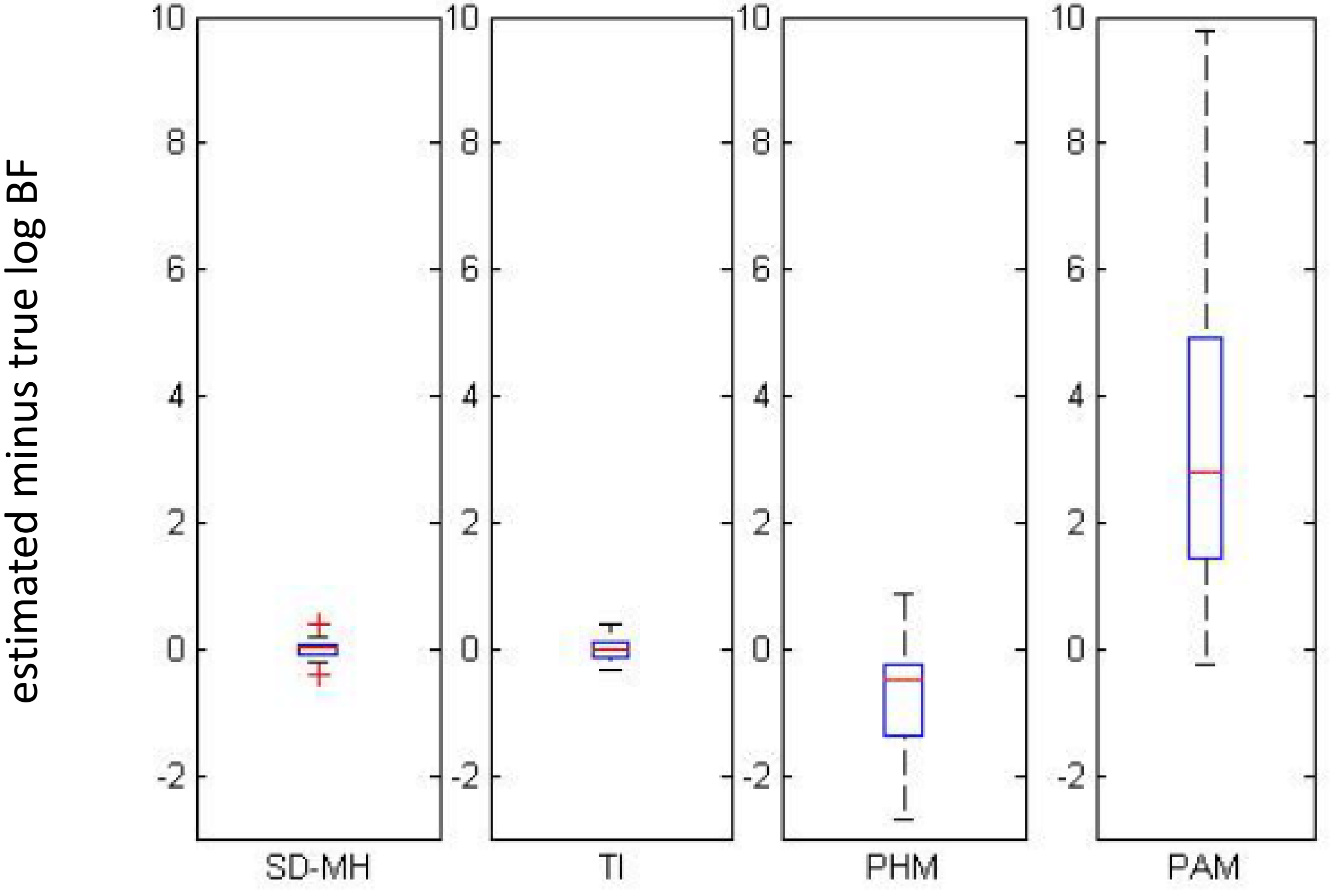
W. Penny

Comparison of methods



Chumbley et al. (2007) *NeuroImage*

Comparison of methods



Chumbley et al. (2007) *NeuroImage*