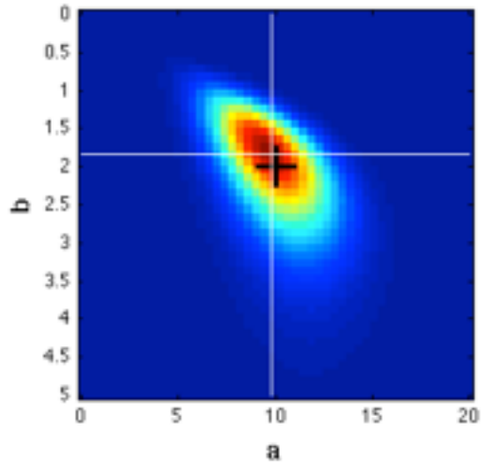
Bayesian modelling

Saad Jbabdi

- Conditioning and marginalisation
- Examples of Bayesian inference
- Approximate inference
  - Analytic
  - Sampling

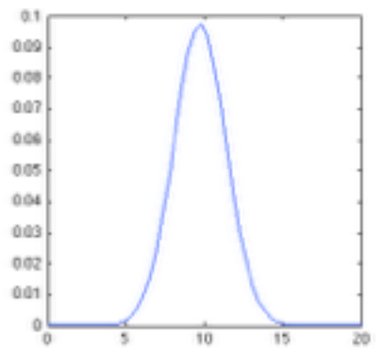
p(A, B)

Joint distribution

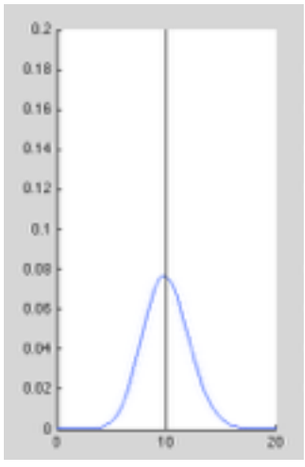


$\int p(A,B)dAdB=1$

Conditioning p(A | B)



Marginalisation p(A, ~~B~~)



$p(A)=\int p(A,B)dB$

- Relationship between **joint** probability and **conditional** probability

$$p(A, B) = p(A \mid B) * p(B)$$ product rule

- Relationship between **joint** probability and **marginal** probability

$$p(A) = \sum_B \{ p(A, B) \}$$ sum rule

Bayes "theorem"

$$p(A, B) = p(A \mid B) * p(B)$$ product rule

$$p(A, B) = p(B \mid A) * p(A)$$ product rule

$$p(A \mid B) = p(B \mid A) * p(A) / p(B)$$

Bayesian modelling and inference

y : data

a : parameters of a model y=F(a)

$$\underbrace{p(a|y)}_{\text{infer}} = \underbrace{p(y|a) * p(a)}_{\text{model}} / p(y)$$

Bayesian modelling and inference

$$p(a | y) = p(y | a) * p(a) / p(y)$$

- model = prior and likelihood
- Bayesian inference = find the posterior

simple.m

A simple example

$$y = a + \text{noise}$$

I measure one noisy data point

what is the probability distribution of a (given y)?

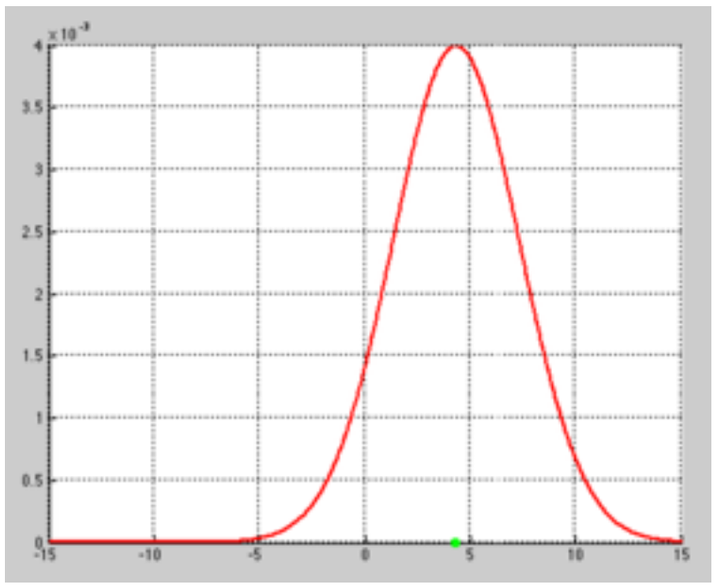
simple.m

A simple example

$$y = a + \text{noise}$$

assumption : noise is $N(0,s^2)$

now, we have a likelihood:



a

$$p(y|a) = N(y|a,s^2)$$

$$p(y|a) = (2\pi s^2)^{-1} \exp\left(-\frac{(y-a)^2}{2s^2}\right)$$

simple.m

A simple example

$$y = a + \text{noise}$$

$$p(y|a) = N(y|a,s^2) \qquad \text{likelihood}$$

$$p(a) = N(a|a_0,s_0^2) \qquad \text{prior}$$

$$p(a|y) = N(y|a,s^2)\ N(a|a_0,s_0^2)/p(y)$$

(product of 2 Gaussians)

$$p(a|y) \sim \exp(-0.5(y-a)^2/s^2)*\exp(-0.5(a-a_0)^2/s_0^2)$$

$$= \exp(-0.5*(a-a_p)^2/s_p^2+\text{const})$$

$$\sim N(a|a_p,s_p^2)$$

simple.m

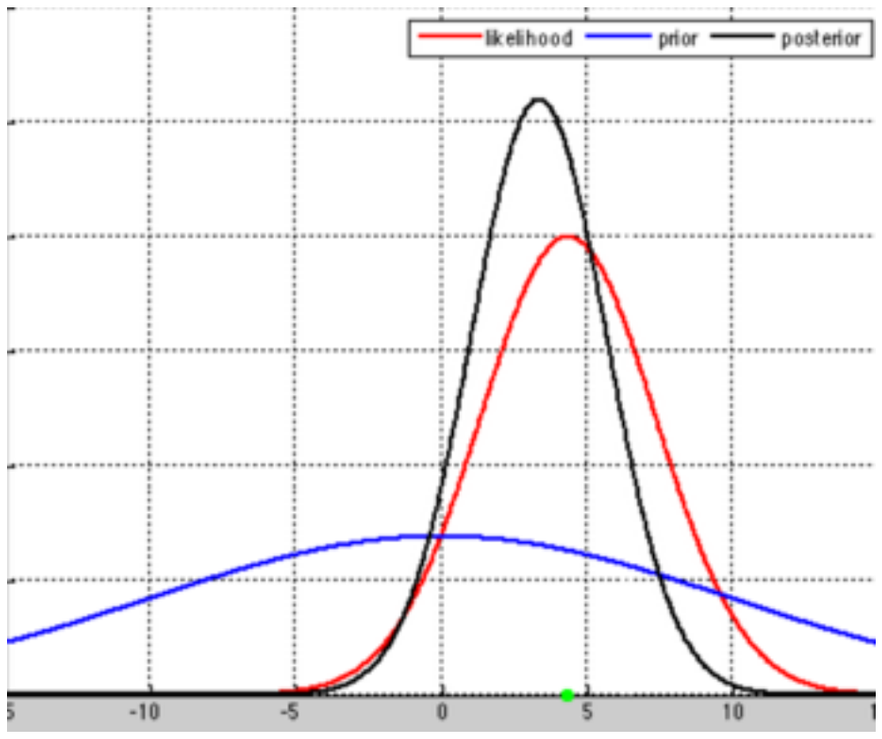
$p(y|a) = N(y|a,s^2)$ likelihood

$p(a) = N(a|a_0,s_0^2)$ prior

$p(a|y) = N(a|a_p,s_p^2)$ posterior

$$
\begin{aligned}
a_p &= s_p^2(\beta y + \beta_0 a_0) \\
s_p^2 &= (\beta + \beta_0)^{-1}
\end{aligned}
$$

precision (1/variance)



simple.m

multiple data points?

extra assumption: noise is i.i.d.
(independent identically distributed samples)

$$p(y_1,y_2,\ldots,y_n|a) = p(y_1|a)\, p(y_2|a) \ldots p(y_n|a)$$

independent

$$p(y_1,y_2,\ldots,y_n|a) = N(y_1|a,s^2)\, N(y_2|a,s^2) \ldots N(y_n|a,s^2)$$

identically distributed

simple.m

A simple example

y = a + noise

I measure N data points

what is the probability distribution of a?

$$\prod_{i=1}^{N} \exp(-\frac{(y_i - a)^2}{2s^2}) \longrightarrow \exp(-\sum_{i=1}^{N} \frac{(y_i - a)^2}{2s^2})$$

$p(a|y) \sim p(y|a)*p(a) = \exp(-0.5\sum(y_i-a)^2/s^2)*\exp(-0.5(a-a_0)^2/s_0^2)$

$= \exp(-0.5*(a-a_p)^2/s_p^2+\text{const})$

$\sim N(a|a_p,s_p^2)$

simple.m

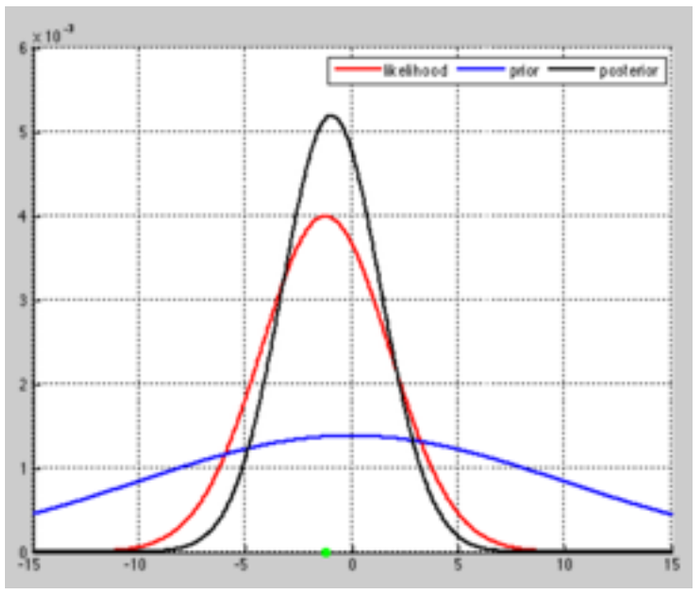
A simple example

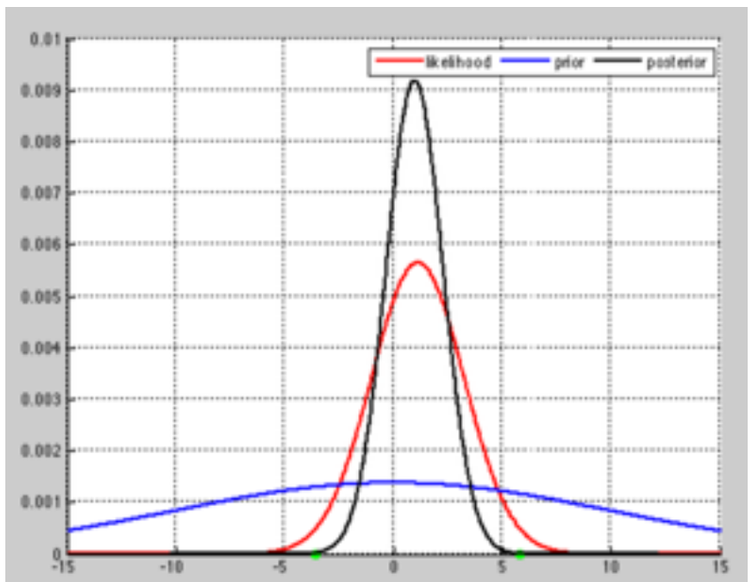
y = a + noise

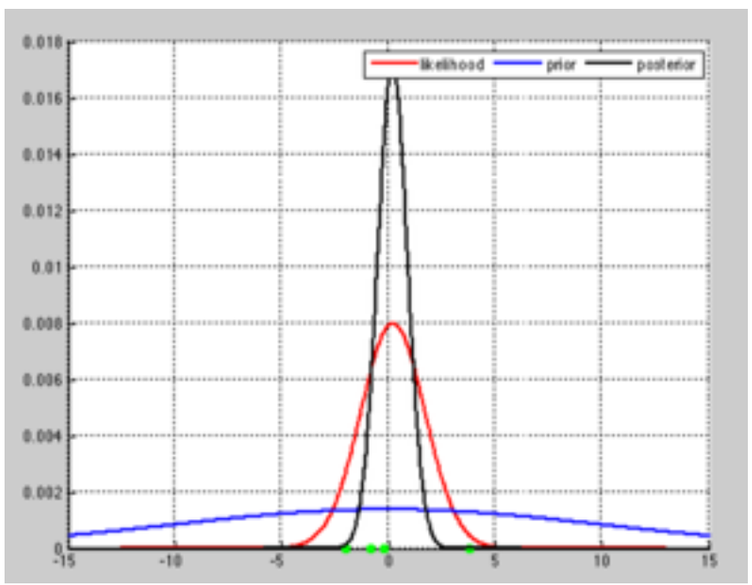
$$p(a|y) \quad \propto \quad \mathrm{N}\left(a_p, s_p^2\right)$$

$$\begin{aligned} a_p &= s_p^2 * \left(n\beta \frac{\sum y_i}{n} + \beta_0 a_0 \right) \\ s_p^2 &= (n\beta + \beta_0)^{-1} \end{aligned}$$

precision (1/variance)







simple.m

Regression

$$y = a*x$$

I measure one noisy data point

what is the probability distribution of a?

simple.m

Regression

$$y = a*x$$

noise model: $y = a*x + N(0,s^2)$

likelihood: $p(y \mid a) = N(a*x, s^2)$

prior: $p(a) = N(a|a_0, s_0^2)$

Bayes: $p(a \mid y) \sim p(y \mid a) * p(a)$

$$p(a|y) \sim p(y|a)*p(a) = \exp(-0.5\sum(y_i-ax_i)^2/s^2)*\exp(-0.5(a-a_0)^2/s_0^2)$$
$$= \exp(-0.5*(a-a_p)^2/s_p^2+\text{const})$$
$$\sim N(a|a_p,s_p^2)$$

Regression

simple.m

$$p(a \mid y) \propto N(a_p, s_p)$$

$$a_p = \frac{bx^T y + b_0 a_0}{bx^T x + b_0}$$

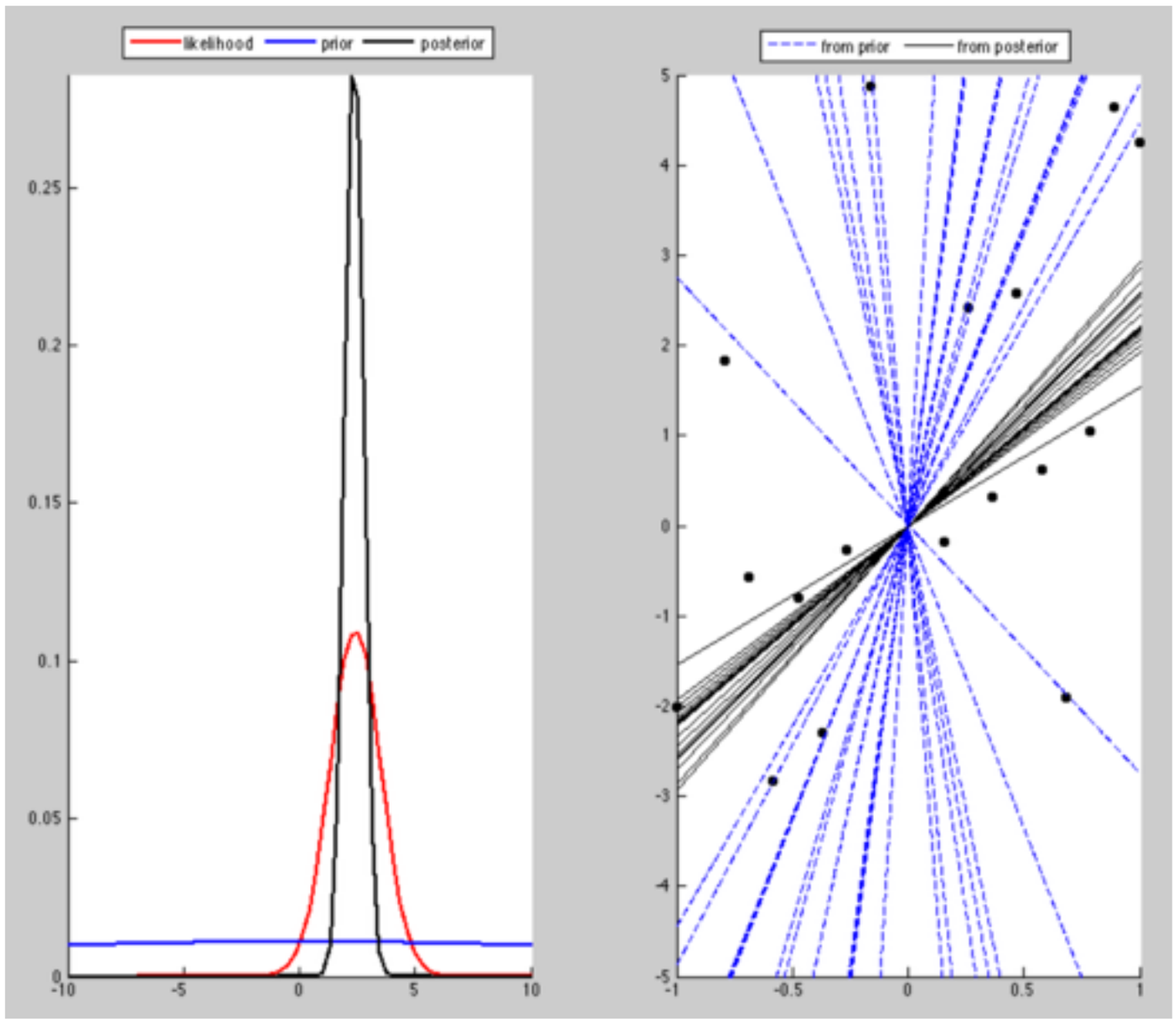
$$s_p = (bx^T x + b_0)^{-1}$$

$b = 1/s^2$

$b_0 = 1/s_0^2$

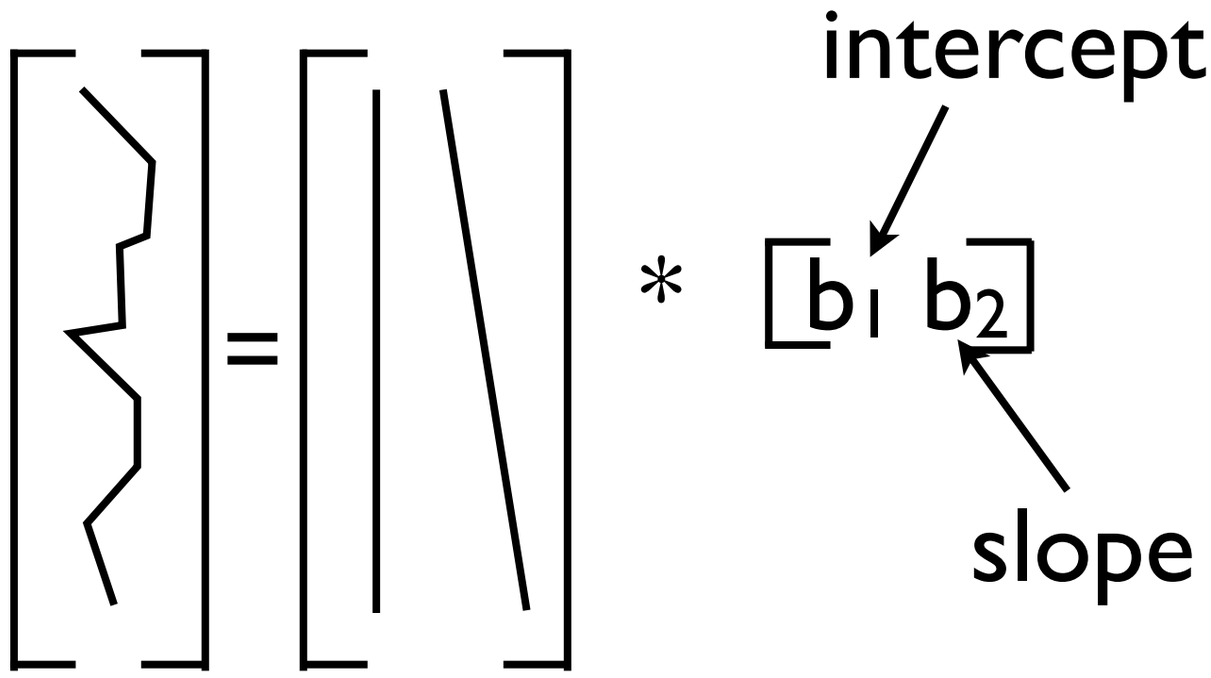
$(X^T X)^{-1} X^T Y$

least squares regression!



Sequential learning

$$Y = X * b$$



seqlearn.m

Sequential learning

$$p(Y \mid b) = p(y_1|b)*p(y_2|b)*\ldots*p(y_n|b)$$

i.i.d. noise

$$p(y_i|b)=N(y_i|(Xb)_i,s^2)$$

$$p(b \mid Y) = p(y_n|b)*p(y_{n-1}|b)*\ldots*p(y_2|b)*p(y_1|b) * p(b)$$

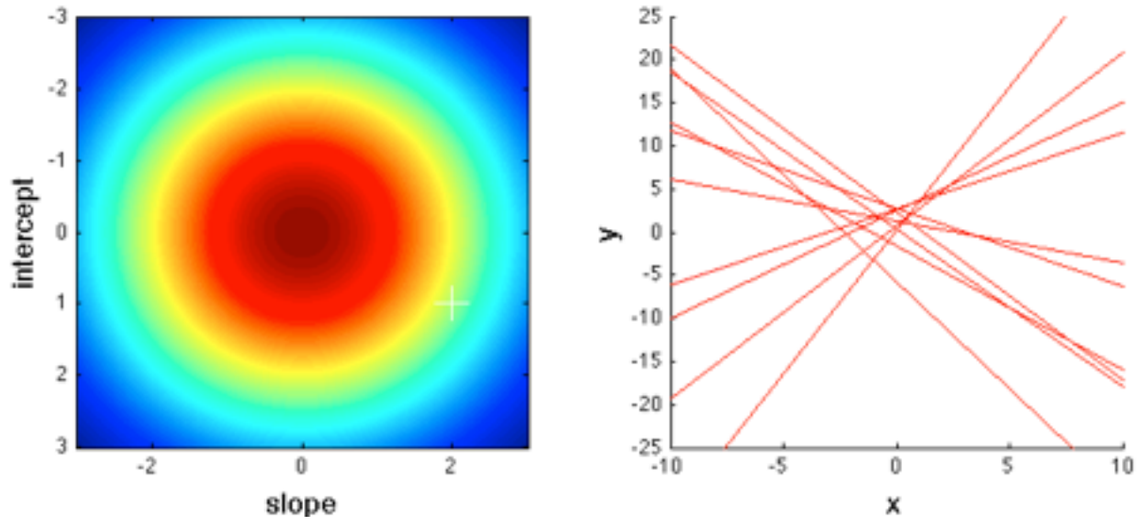
$p(b|y_1)$

$p(b|y_1,y_2)$

posterior given y_1 becomes prior for y_2, etc.etc.

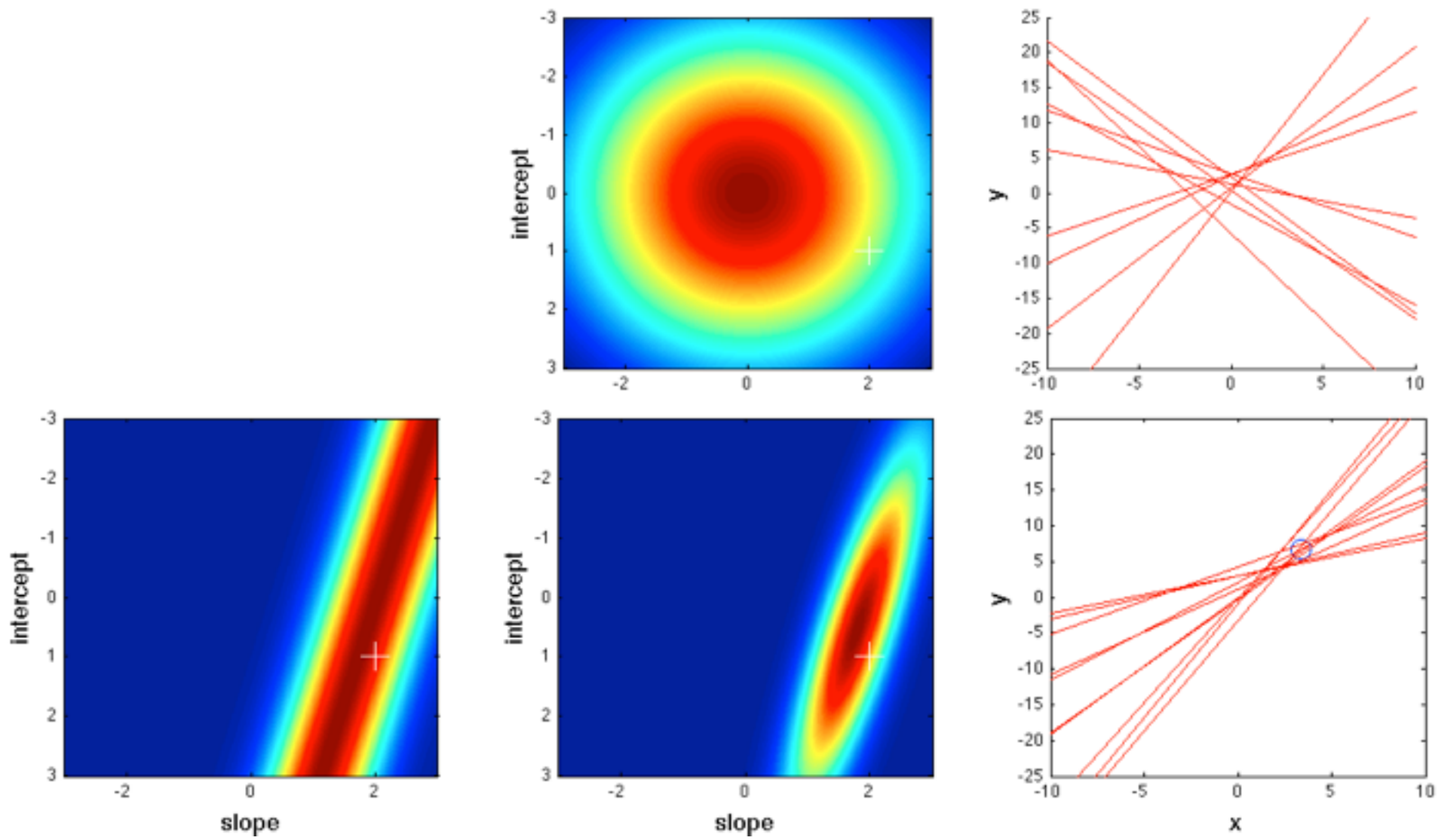
Sequential learning

seqlearn.m



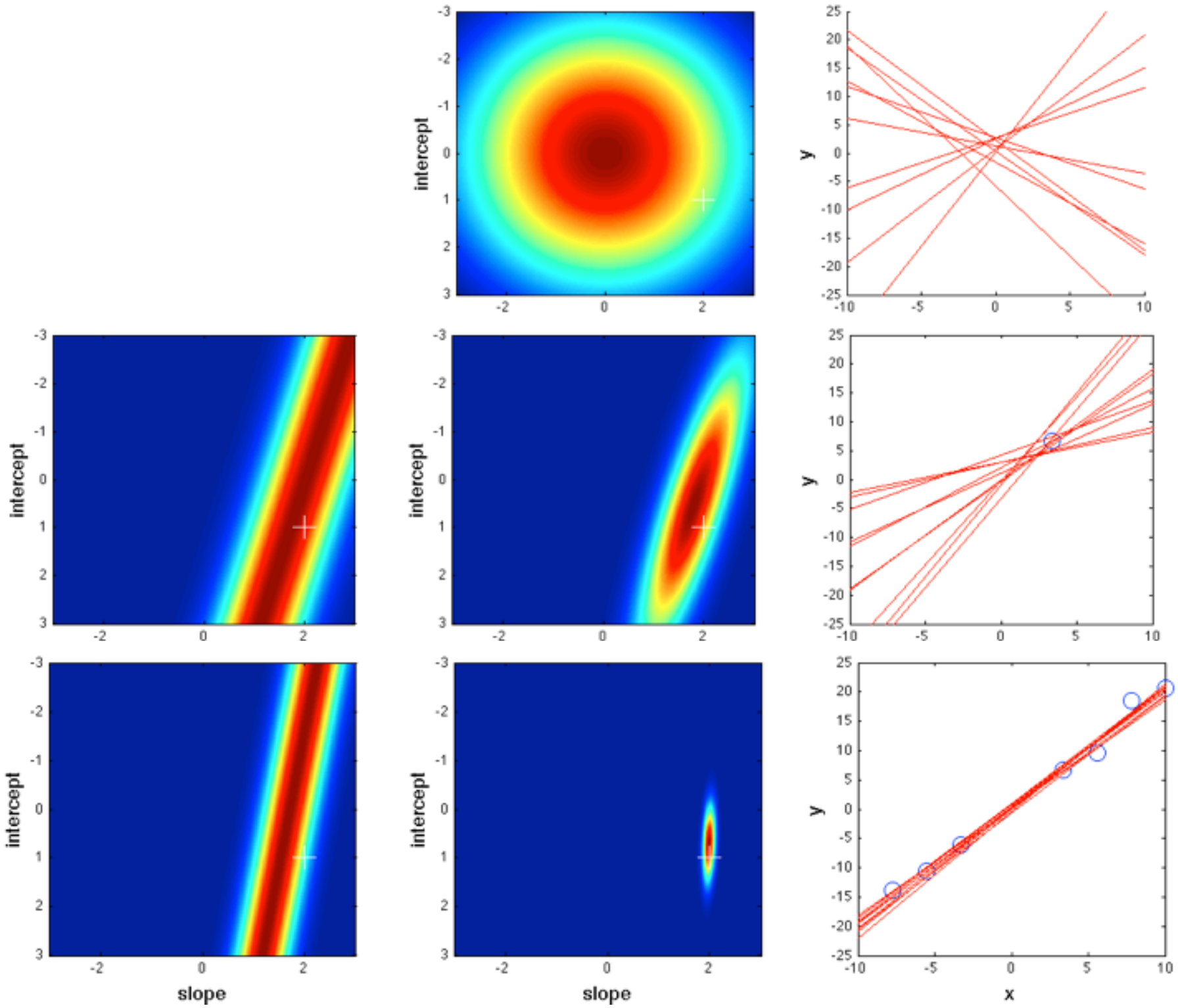
Sequential learning

seqlearn.m



Sequential learning

seqlearn.m



seqlearn.m

Sequential learning

The posterior distribution is the prior for the next data point

nonlin.m

Nonlinear model

$$y = a*\exp(-b*x) = f(a,b)$$

noise model: $y = f(a,b) + N(0,s^2)$

likelihood: $p(y|a,b) = N(y|f(a,b),s^2)$

prior: $p(a,b) = N(a|a_0,s_{a0}^2)*N(b|b_0,s_{b0}^2)$

posterior: $p(a,b|y) \sim \exp[\ -0.5(y-f(a,b))^2/s^2\ ]*p(a,b)$

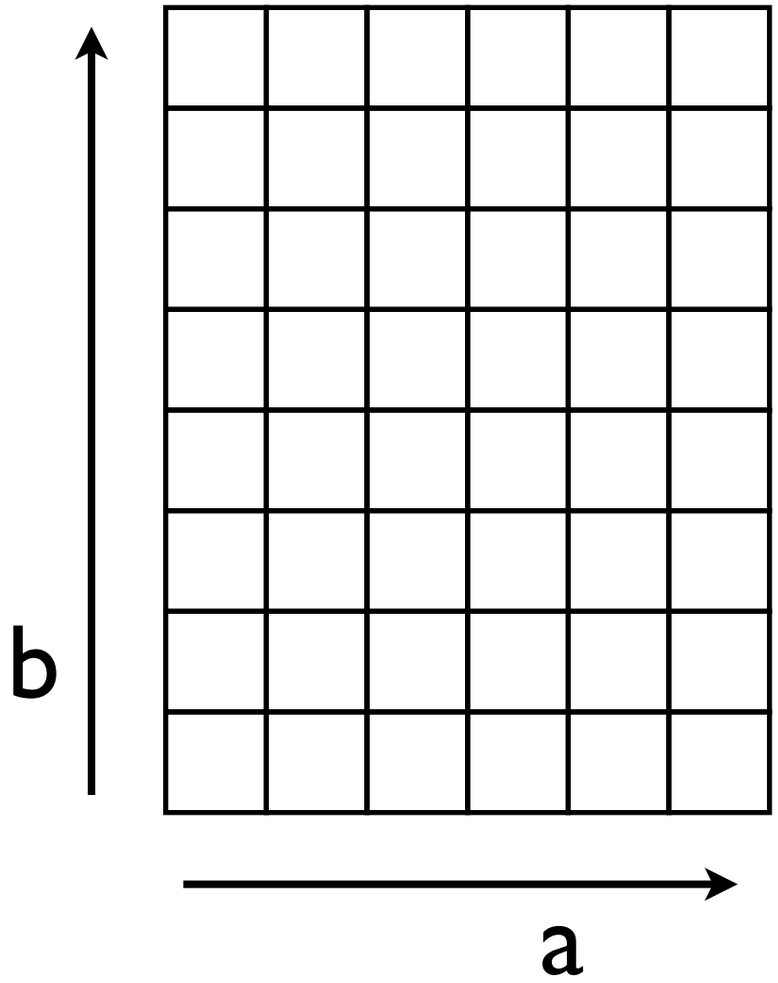
not a Gaussian... what do we do?

nonlin.m

Grid evaluation

$$y = a*\exp(-b*x)$$

$$\text{posterior: } p(a,b|y) = \text{Const}*\exp[-0.5(y-f(a,b))^2/s^2]*p(a,b)$$



calculate p(a,b|y) (to a constant)
normalise by sum over the whole grid

nonlin.m

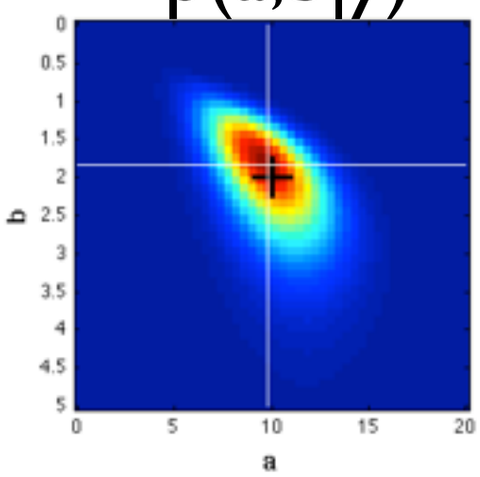
Grid evaluation

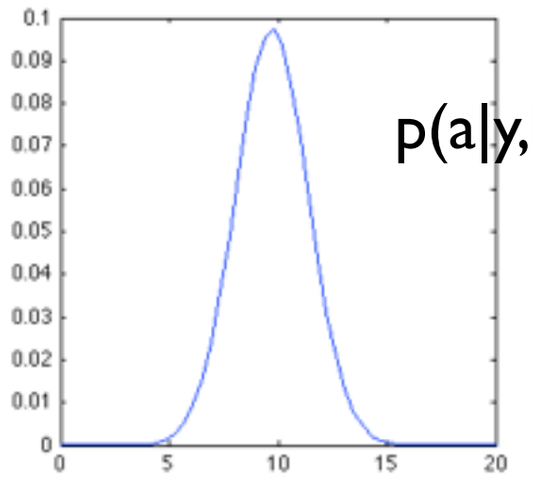
$y = a*\exp(-b*x)$

$x=1:10$

full posterior

$p(a,b|y)$



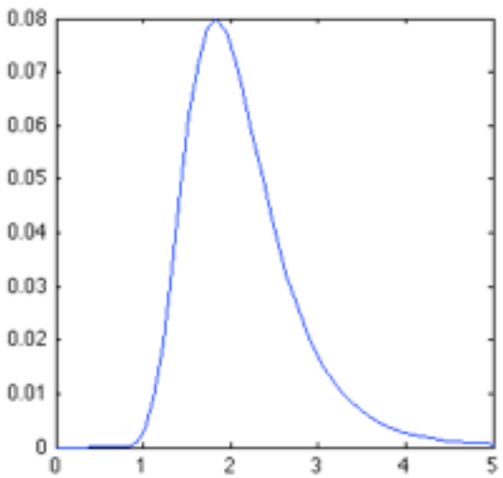


$a=10$

$b=2$

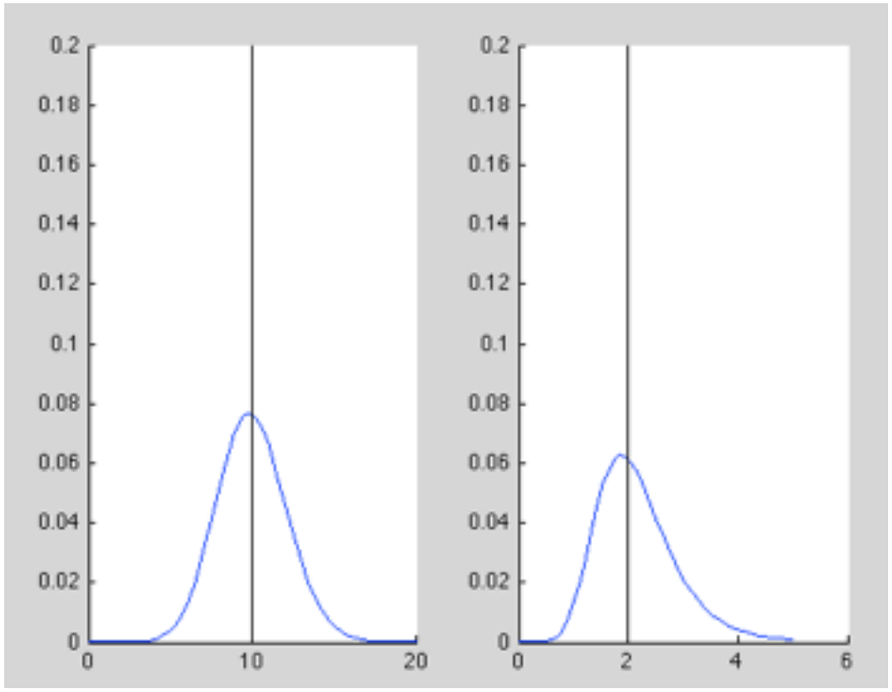
$s^2=3$

conditionals



$p(b|y,a=10)$

$p(a|y)$ $p(b|y)$



marginal posteriors

nonlin.m

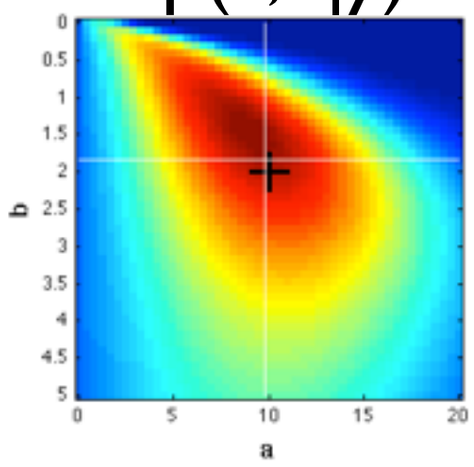
Grid evaluation

$y = a*\exp(-b*x)$

$x=1:10$

full posterior

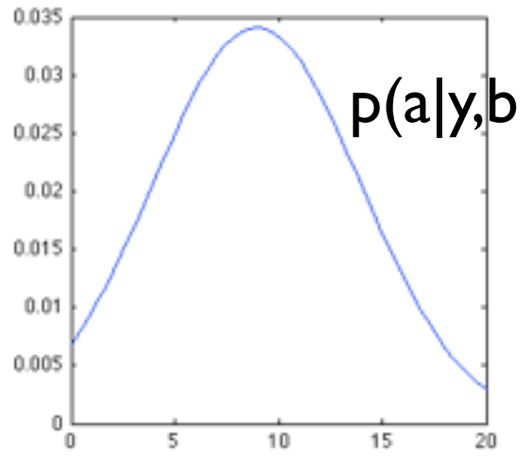
$p(a,b|y)$



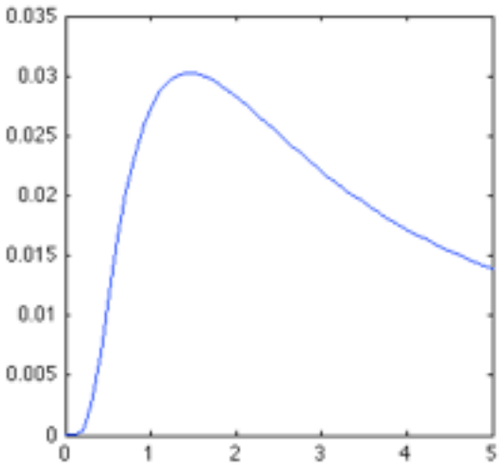
$a=10$

$b=2$

$s^2=10$

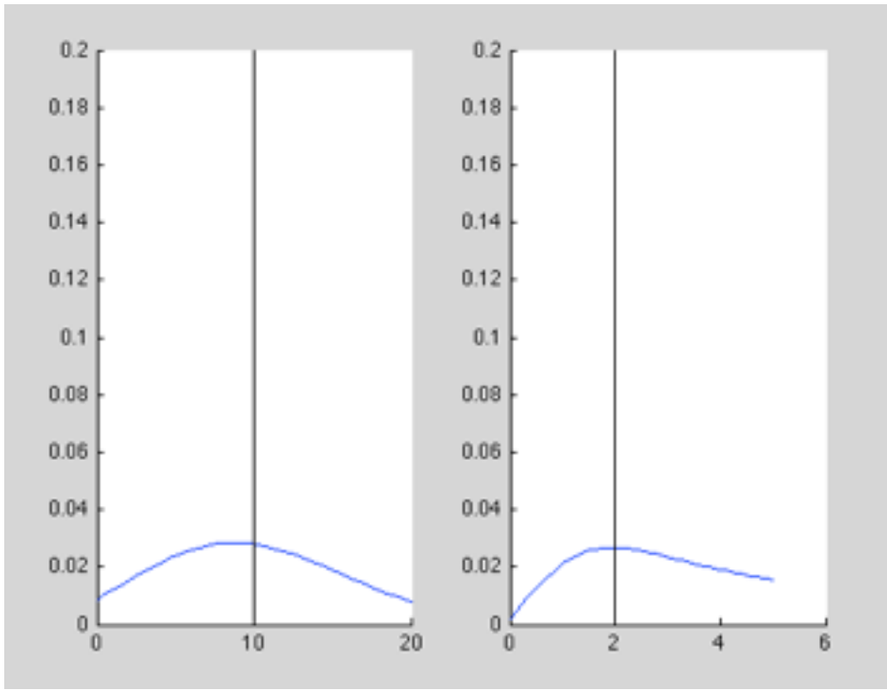


conditionals



$p(b|y,a=10)$

$p(a|y)$ $p(b|y)$



marginal posteriors

Grid search

but what if we had MANY parameters?
e.g. 5 parameters, 100 bins each = 100^5 values
to calculate (and store)

solutions:
- approximate with an easier distribution
- sample directly from the distribution (e.g. MCMC)

nonlin.m

Laplace Approximation

posterior: $p(\mathbf{x}|y) \sim \exp(-.5* (y-f(\mathbf{x}))^2/s^2) * p(\mathbf{x})$

$-\log p(\mathbf{x}|y) \sim (y-f(\mathbf{x}))^2/s^2$ ignore priors

$-\log p(\mathbf{x}|y) \sim g(\mathbf{x})/s^2$

$g(\mathbf{x}) \sim g(\mathbf{x}_m)+g'(\mathbf{x}_m)(\mathbf{x}-\mathbf{x}_m)+.5*g''(\mathbf{x}_m)(\mathbf{x}-\mathbf{x}_m)^2 +...$ (Taylor theorem)

$g(\mathbf{x}) \sim g(\mathbf{x}_m)+.5*g''(\mathbf{x}_m)(\mathbf{x}-\mathbf{x}_m)^2$

$\mathbf{x}_m$ = maximum likelihood

constant

quadratic, i.e. $p(\mathbf{x})$ is like a Gaussian

nonlin.m

Laplace Approximation

$\log(p(a,b|y) \sim g(\mathbf{x})/s^2 \sim \text{const} + 0.5 * g''(\mathbf{x}_m)(\mathbf{x}-\mathbf{x}_m)^2/s^2$

posterior: $p(a,b|y) \sim N(y \mid (a_m;b_m), s^2 H^{-1})$

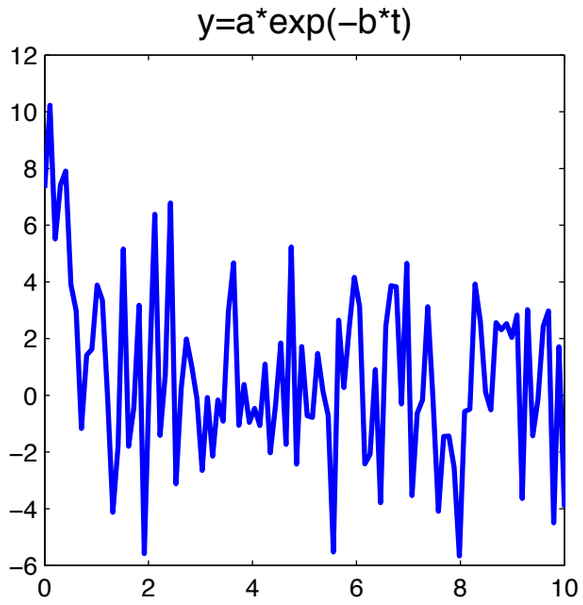
$$H = \begin{bmatrix} g_{aa} & g_{ab} \\ g_{ba} & g_{bb} \end{bmatrix}$$

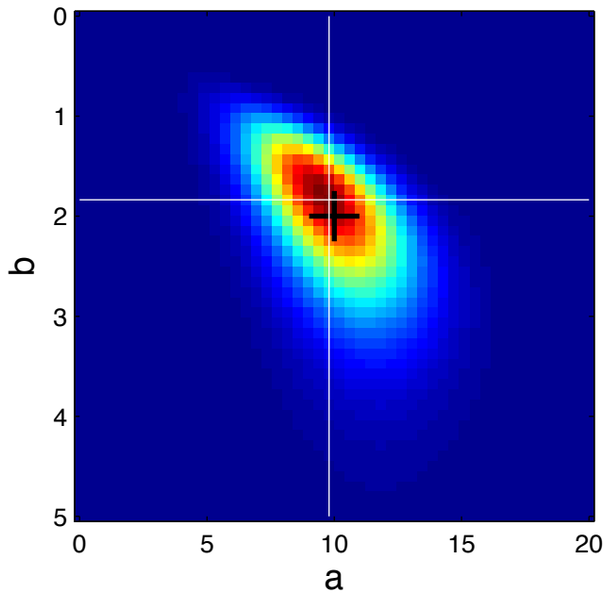
$g = (y-f(a,b))^2$

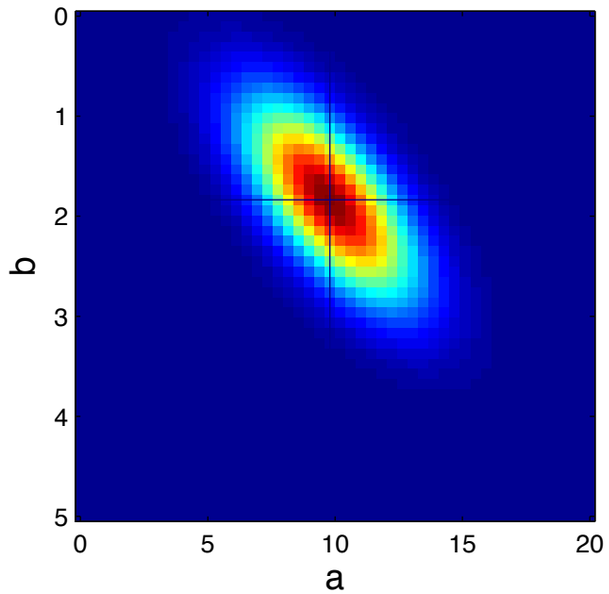
Laplace approx in one sentence:
"curvature of the posterior PDF around its max"

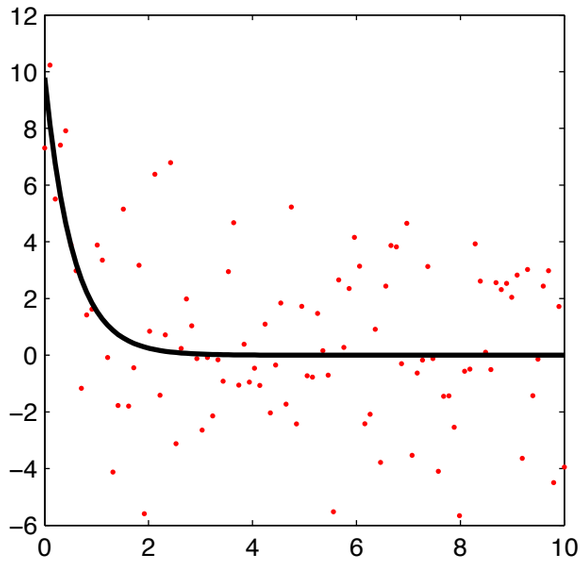
nonlin.m

Laplace Approximation

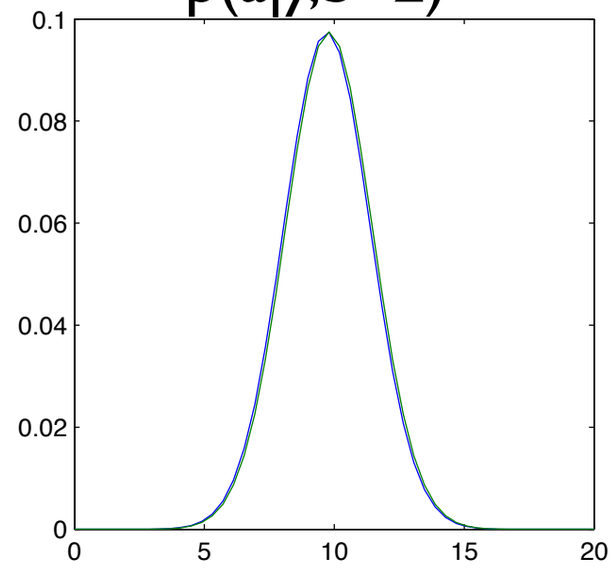




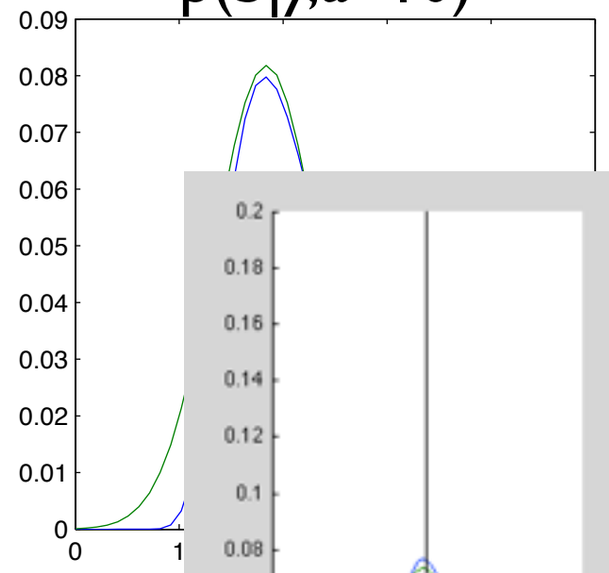




p(a|y,b=2)



p(b|y,a=10)



approximate full posterior by a “local” Gaussian
All marginals and conditionals are Gaussians

Cannot work if posterior is complicated

Variational Bayes (VB)

q(a) instead of p(a|y)

approximate with simple q(a) with unknown "parameters"

$$KL = \int q(a)\log[p(a|y)/q(a)]$$

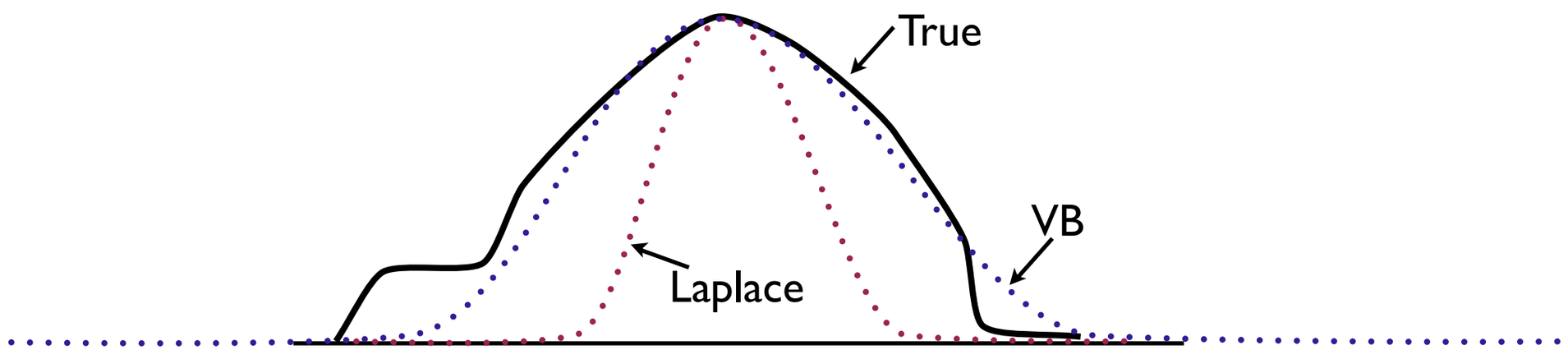
Kullback Leibler
(distance between true and approximate posterior)

$$F = \int q(a)\log[p(y|a)p(a)/q(a)]$$

define Free Energy

$$F+KL=\log[p(y)]$$

constant ---> minimize KL wrt q by maximizing F



The kind of maths that you typically have to do...

$$
\begin{aligned}
\ln q^*_\mu(\mu) &= \mathrm{E}_\tau[\ln p(\mathbf{X}|\mu,\tau) + \ln p(\mu|\tau) + \ln p(\tau)] + C \\
&= \mathrm{E}_\tau[\ln p(\mathbf{X}|\mu,\tau)] + \mathrm{E}_\tau[\ln p(\mu|\tau)] + \mathrm{E}_\tau[\ln p(\tau)] + C \\
&= \mathrm{E}_\tau[\ln \prod_{n=1}^{N} \mathcal{N}(x_n|\mu,\tau^{-1})] + \mathrm{E}_\tau[\ln \mathcal{N}(\mu|\mu_0,(\lambda_0\tau)^{-1})] + C_2 \\
&= \mathrm{E}_\tau[\ln \prod_{n=1}^{N} \sqrt{\frac{\tau}{2\pi}} e^{-\frac{(x_n-\mu)^2\tau}{2}}] + \mathrm{E}_\tau[\ln \sqrt{\frac{\lambda_0\tau}{2\pi}} e^{-\frac{(\mu-\mu_0)^2\lambda_0\tau}{2}}] + C_2 \\
&= \mathrm{E}_\tau[\sum_{n=1}^{N} \left(\frac{1}{2}(\ln\tau - \ln 2\pi) - \frac{(x_n-\mu)^2\tau}{2}\right)] + \mathrm{E}_\tau[\frac{1}{2}(\ln\lambda_0 + \ln\tau - \ln 2\pi) - \frac{(\mu-\mu_0)^2\lambda_0\tau}{2}] + C_2 \\
&= \mathrm{E}_\tau[\sum_{n=1}^{N} -\frac{(x_n-\mu)^2\tau}{2}] + \mathrm{E}_\tau[-\frac{(\mu-\mu_0)^2\lambda_0\tau}{2}] + \mathrm{E}_\tau[\sum_{n=1}^{N} \frac{1}{2}(\ln\tau - \ln 2\pi)] + \mathrm{E}_\tau[\frac{1}{2}(\ln\lambda_0 + \ln\tau - \ln 2\pi)] + C_2 \\
&= \mathrm{E}_\tau[\sum_{n=1}^{N} -\frac{(x_n-\mu)^2\tau}{2}] + \mathrm{E}_\tau[-\frac{(\mu-\mu_0)^2\lambda_0\tau}{2}] + C_3 \\
&= -\frac{\mathrm{E}_\tau[\tau]}{2}\{\sum_{n=1}^{N}(x_n-\mu)^2 + \lambda_0(\mu-\mu_0)^2\} + C_3
\end{aligned}
$$

Noise trick

- so far we have assumed that we know the variance of the noise s^2
- what to do when we don't want to make that assumption?
  - Estimate it like the other parameters
  - Marginalise wrt the noise variance

Noise trick

Let us assume a Gaussian noise model.

$$p(y|x,s^2) = N(f(x),s^2)$$

If we want to estimate s^2, we need a prior on s^2

Get one from a hat: $p(s^2)=1/s^2$

$$p(x|y) = \int p(x|y,s^2)p(s^2)\, ds^2$$ Marginalisation

whiteboard calculation of:

$$p(x|y) = \int N(y|f(x),s^2)p(s^2)\, ds^2$$

gives:

$$p(x|y) \propto |y\text{-}f(x)|^{-n}$$

- When we don't care about a parameter, but we still know it's "there", it is often useful to "integrate it out" (marginalise)

likelihood

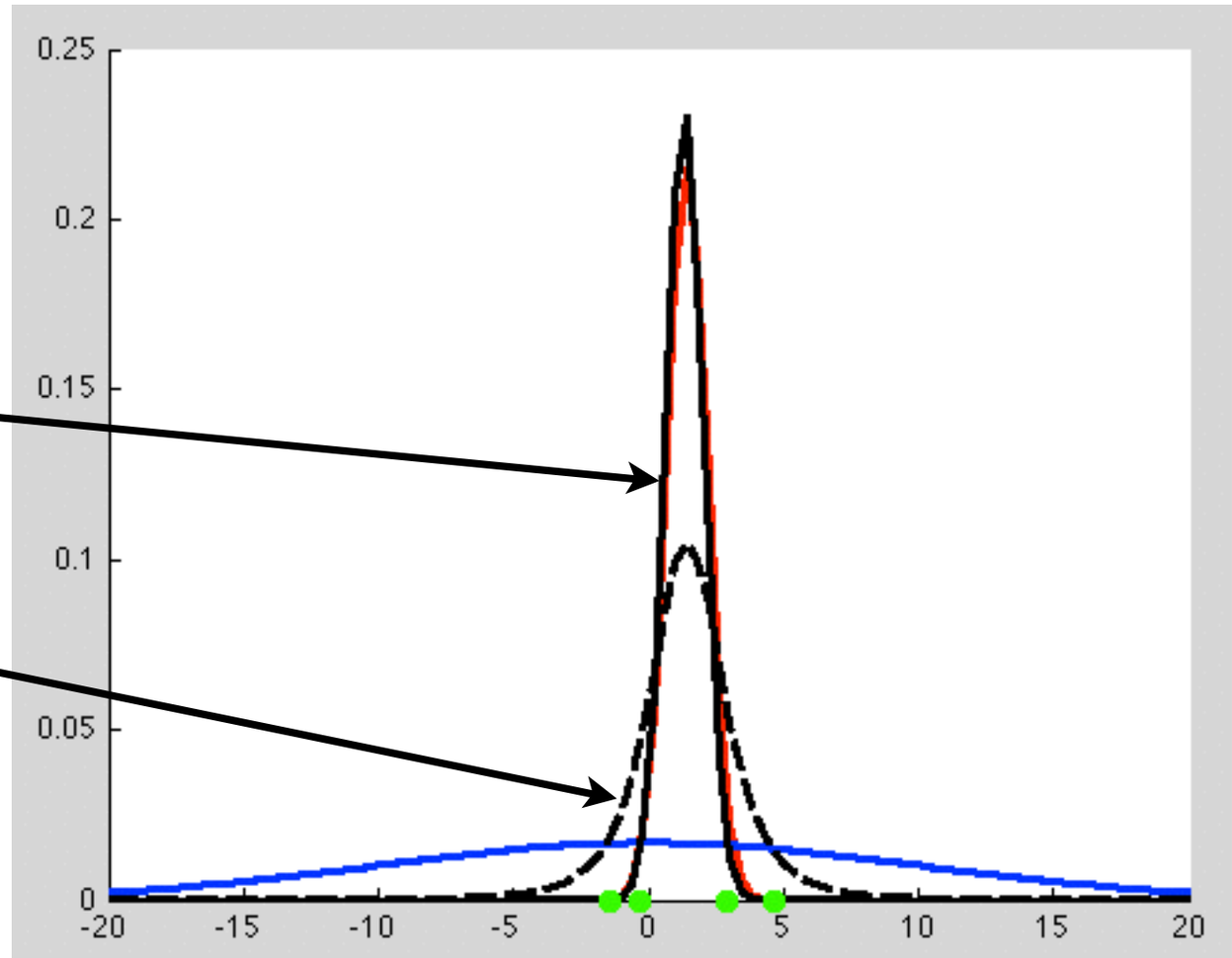
prior

posterior

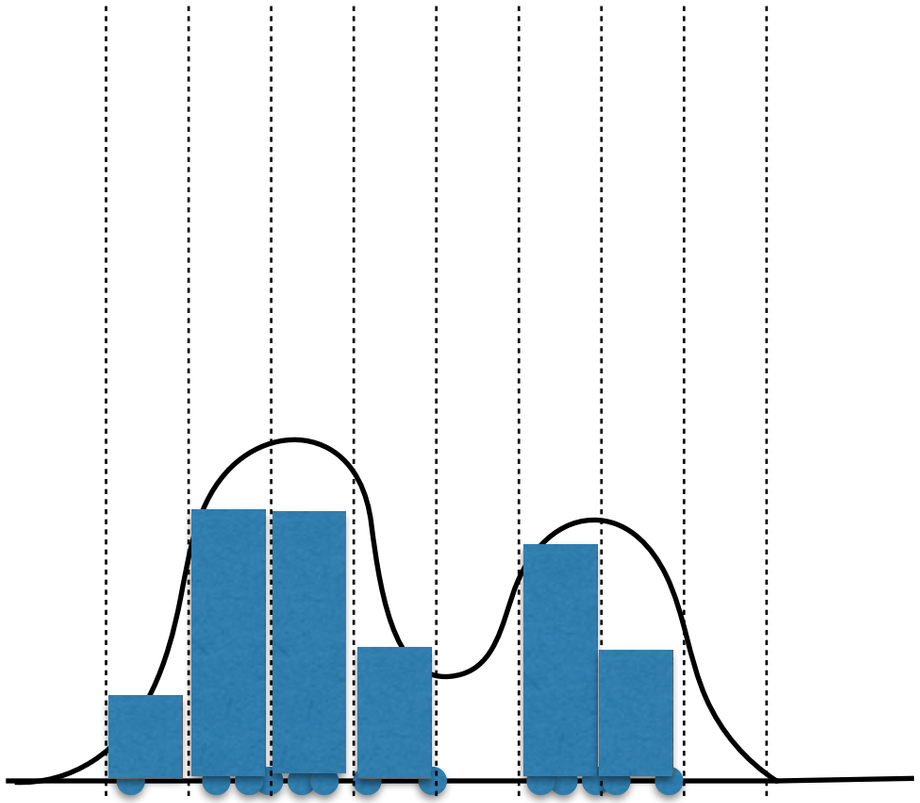
y = a + noise

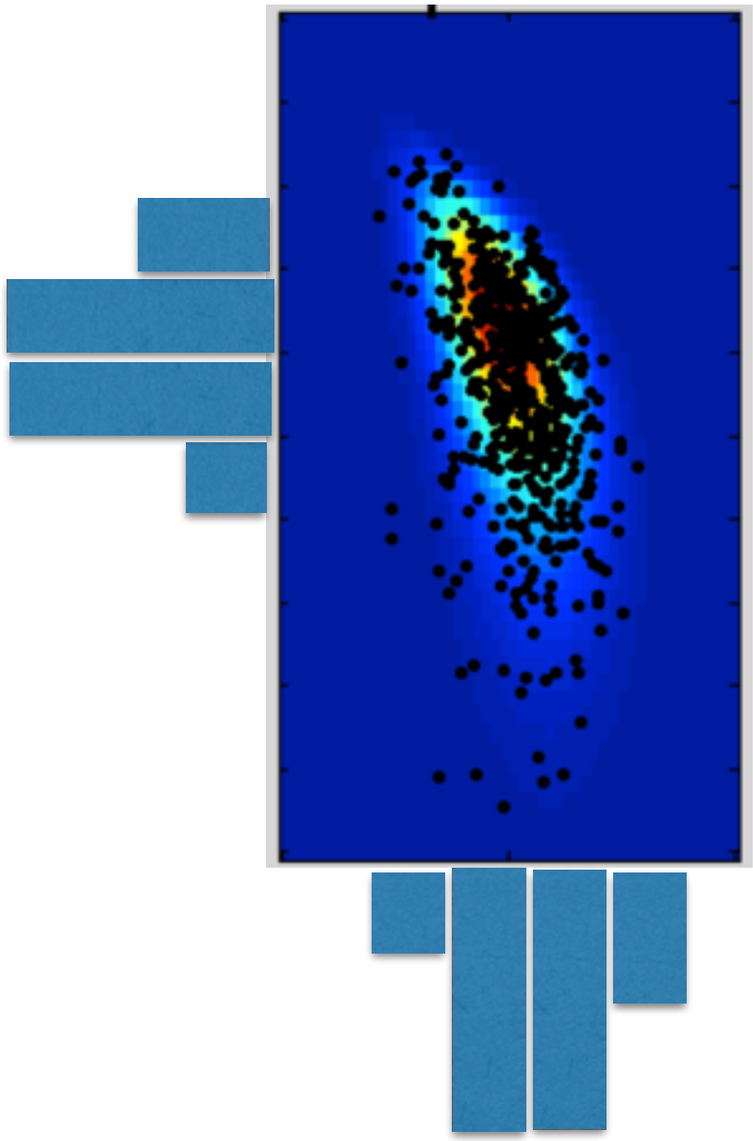
$p(a|y) \propto N(a,m_p,s_p^2)$

$p(a|y) \propto |y-a|^{-n}$



Sampling





Gibbs sampling

$$y = a*x$$

noise model: $y = a*x + N(0,s^2)$

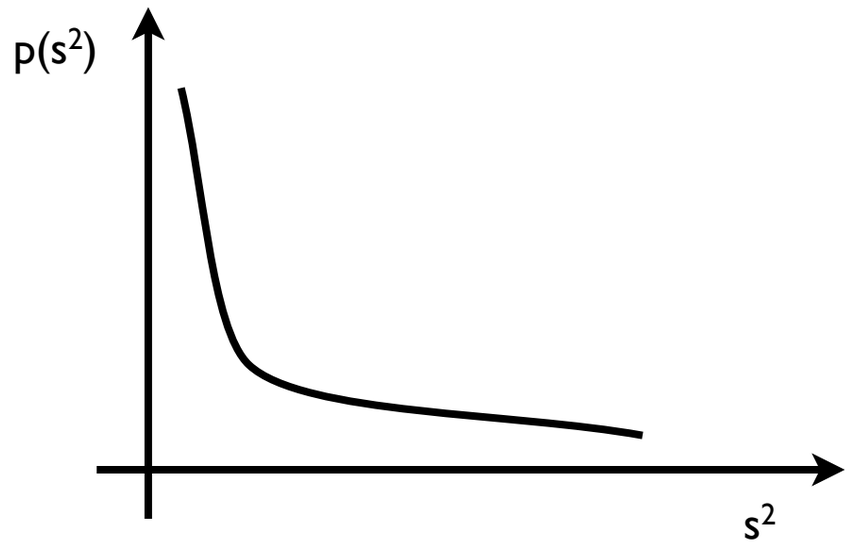
what if we want to learn about this one?

$$p(a,s^2|y) \sim p(y|a,s^2)\ p(a)\ p(s^2)$$

likelihood: $p(y|a,s^2) = N(y|a*x,s^2)$

$$p(a) = N(a|a_0,s_0^2)$$

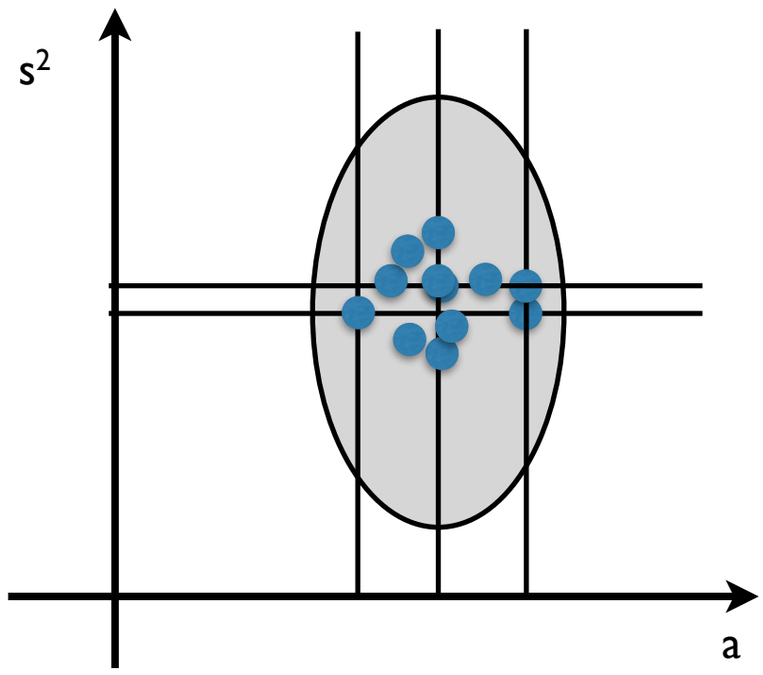
$$p(s^2) = 1/s^2$$



Gibbs sampling

iterate: sample from $p(a|y,s^2)$

sample from $p(s^2|y,a)$



Gibbs sampling

$p(a|y,s^2) = N(a_p,s_p)$ we can easily sample from this

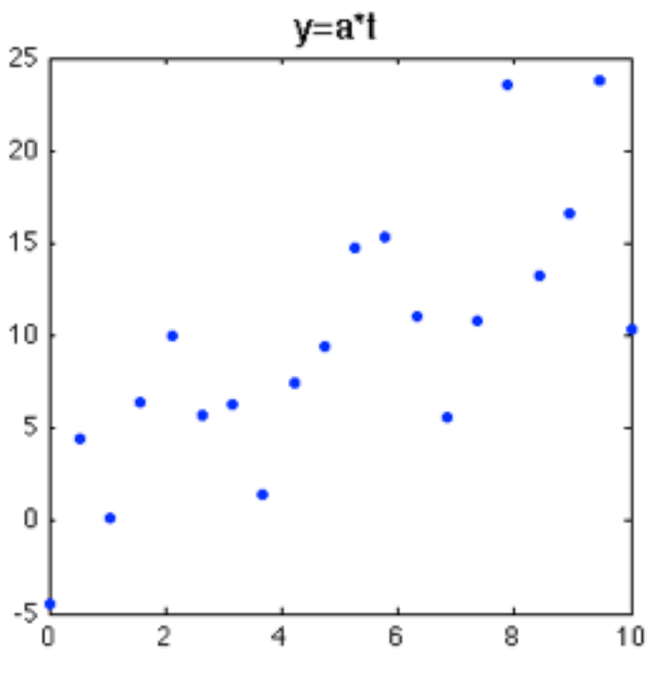
$p(s^2|y,a) \propto p(y|a,s^2)p(s^2)$ product rule

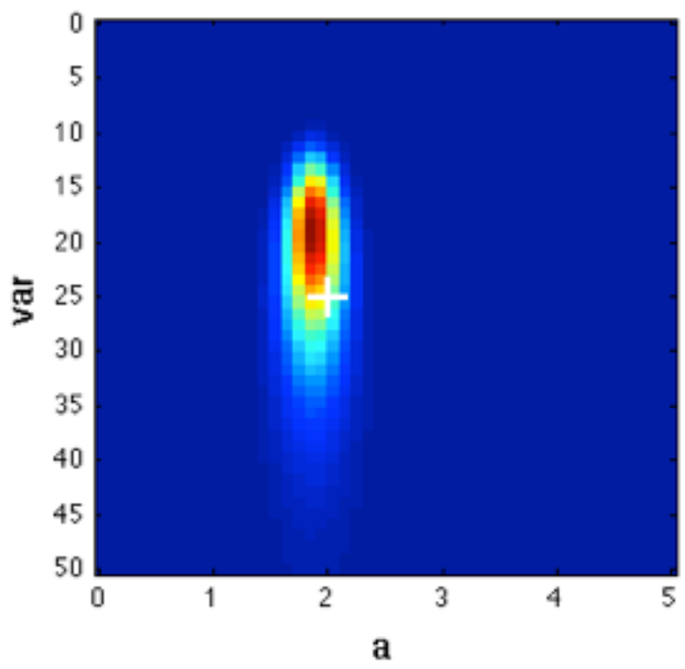
$$p(s^2|y,a) \propto (s^2)^{-n/2} \exp \frac{-\sum(y_i - ax_i)^2}{2s^2} \frac{1}{s^2}$$

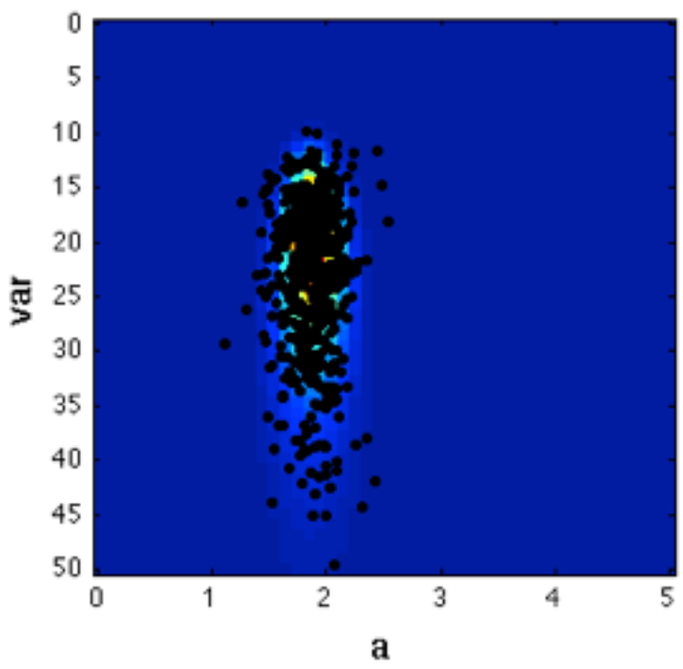
$$\text{inv} - \Gamma(s^2|n/2, \sum(y_i - ax_i)^2))$$

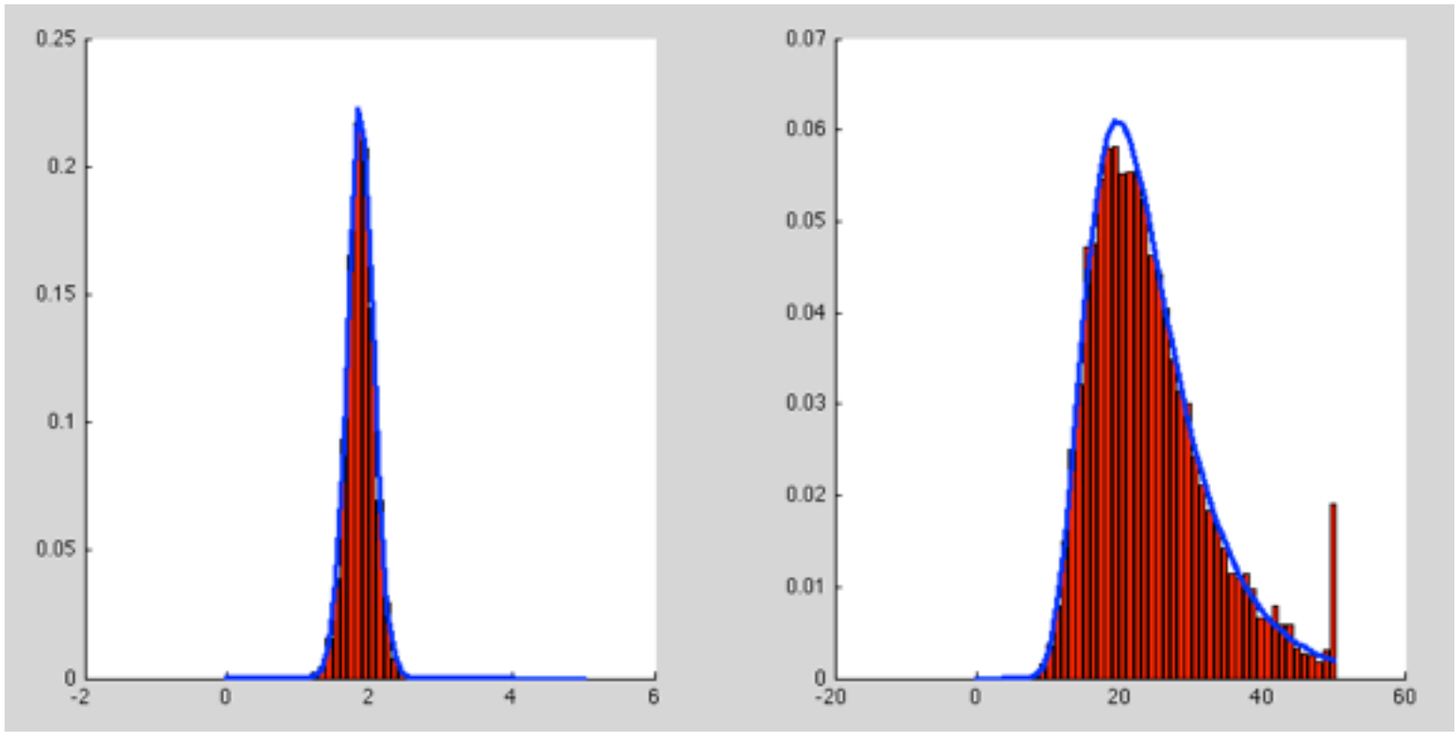
gibbs.m

Gibbs sampling

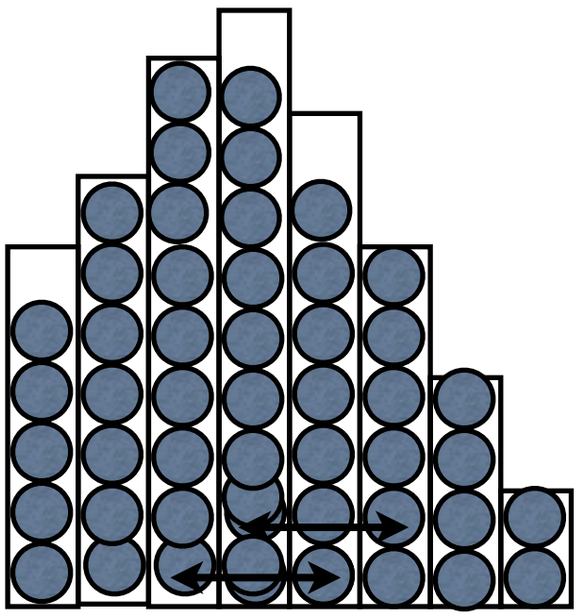








Metropolis Hastings



$a_1 \xrightarrow[q]{} a_2 \rightarrow a_3 \rightarrow a_4 \rightarrow a_4 \rightarrow a_5 \rightarrow a_6 \rightarrow a_7 \rightarrow a_8 \rightarrow a_9$

$a_{n+1} \sim q(a_n)$ sample from q (typically Gaussian)

Metropolis Hastings

$a_{n+1} \sim q(a_n)$ sample from q (typically Gaussian)

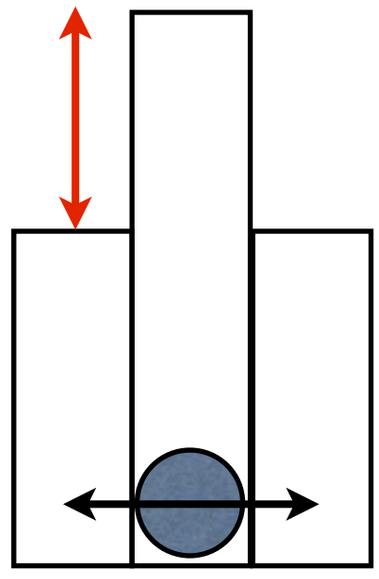
if $p(a_{n+1}|y) > p(a_n|y)$

move!

if $p(a_{n+1}|y) < p(a_n|y)$

stay with probability proportional to ratio:

$$\frac{p(a_{n+1}|y)}{p(a_n|y)}$$



Final residence map proportional to full posterior

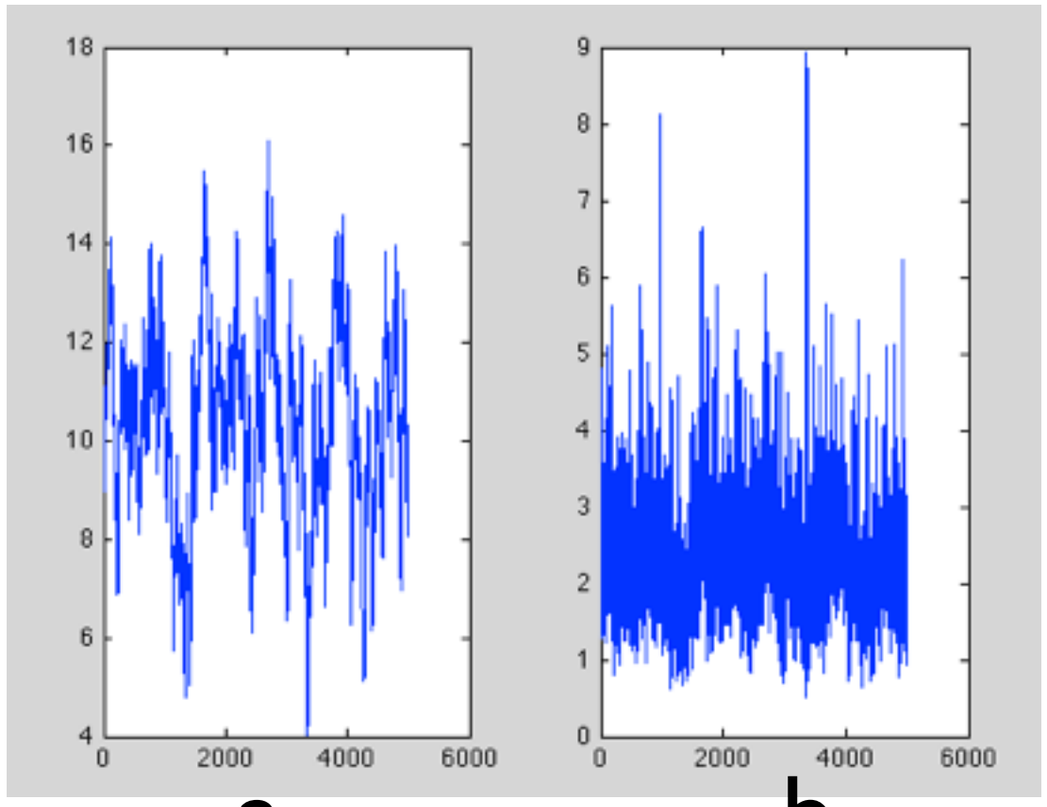
Metropolis Hastings Code

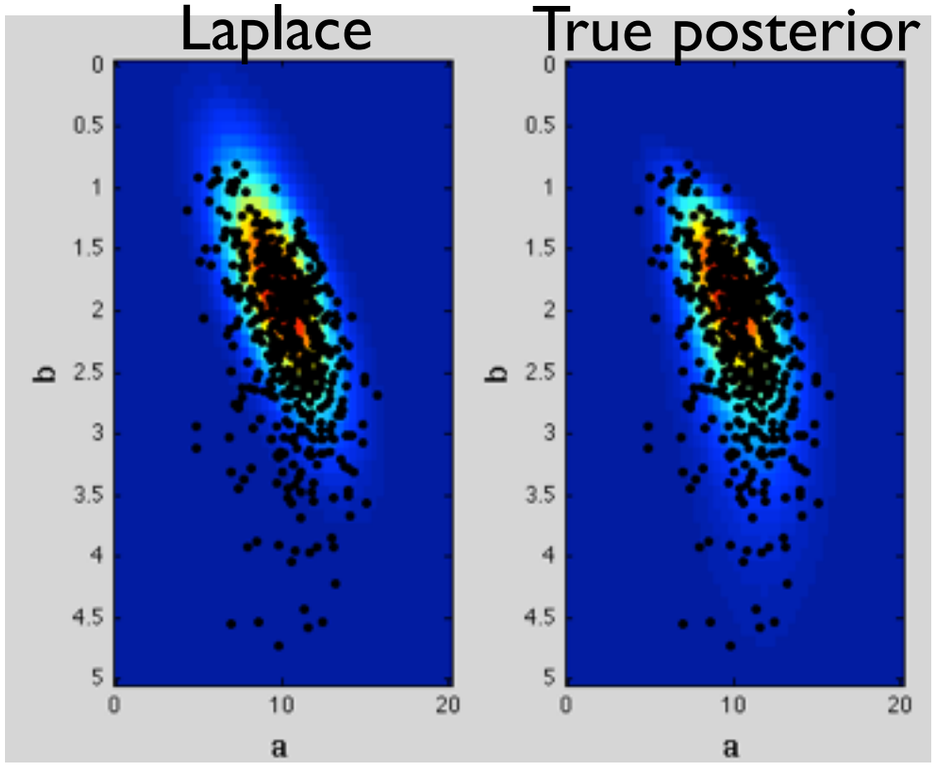
```
function samples = mh(varargin)
% res = mh(y,x0,@genfunc,[LB,UB,params])
%
% Compulsory Parameters
%   y        = data (Nx1)
%   x0       = initial parameters (Px1)
%   @genfunc = generative model
%
%   Output is nsamples*P where nsamples=njumps/sampleevery
%
% Example:
%   % define a forward model (here y=a*exp(-bx))
%   myfun=@(x,c)(exp(-x(1)*c)+x(2));
%   % generate some noisy data
%   true_x = [1;2];
%   c=linspace(1,10,100);
%   y=myfun(true_x,c) + .05*randn(1,100);
%   % estimate parameters
%   x0=[1;2]; % you can get x0 using nonlinear opt
%   samples=mh(y,x0,@(x)(myfun(x,c)));
%   figure,plot(samples)
%
% Other Parameters
%   LB = lower bounds on x (default=[-inf]*ones(size(x0)))
%   UB = upper bounds on x (default=[+inf]*ones(size(x0)))
%   params.burnin = #burnin iterations (default=1000)
%   params.njumps = #jumps (default=5000)
%   params.sampleevery = keep every n jumps (default=10)
%   params.update = update proposal every n jumps (default=20)
%
%
% S. Jbabdi 01/12
```

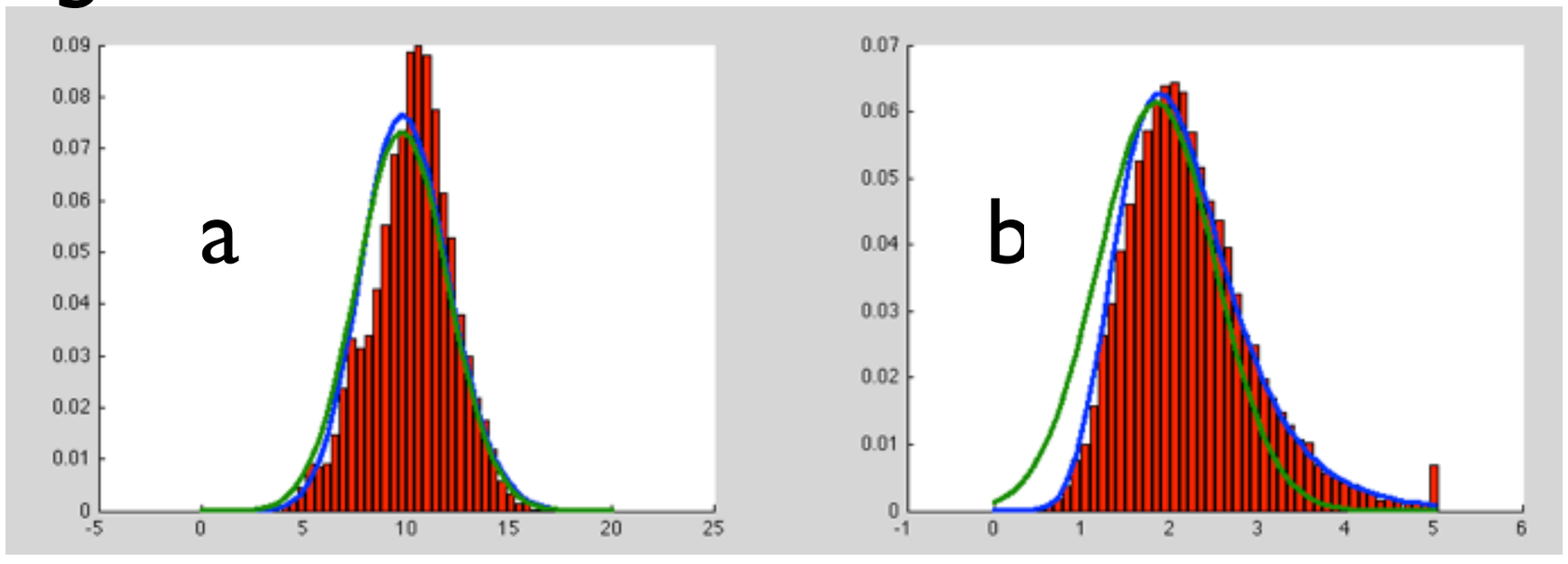
nonlin.m

Metropolis Hastings

$$y = f(a, b) = a\mathrm{e}^{-bt}$$







Metropolis Hastings

- extremely flexible
  - all sorts of priors possible
  - all sorts of models (linear/nonlinear)
  - high dimensional problems
- computationally intensive
- not easy to monitor convergence

Example

simple_mri_example.m

$$Y = M_0 * \exp(-TE/T_2) * (1 - \exp(-TR/T_1))$$

known: data, measurements; experimental parameters

unknown: model parameters

Example

simple_mri_example.m

$$Y = M_0 * \exp(-TE/T_2) * (1-\exp(-TR/T_1))$$

Let's call these **x**

$$Y_i = F_i(M_0, T_2, T_1)$$

each 'i' is a different combination (TE,TR)

Example

simple_mri_example.m

$$Y = M_0 * \exp(-TE/T_2) * (1 - \exp(-TR/T_1))$$

$$Y_i = F_i(\mathbf{x})$$

“generative model”

Example

simple_mri_example.m

$$Y = M_0 * \exp(-TE/T_2) * (1-\exp(-TR/T_1))$$

$$Y_i = F_i(\mathbf{x}) + \text{noise}$$

noise trick:

$$p(\mathbf{x}|Y) \propto |Y-F(\mathbf{x})|^{-n}$$

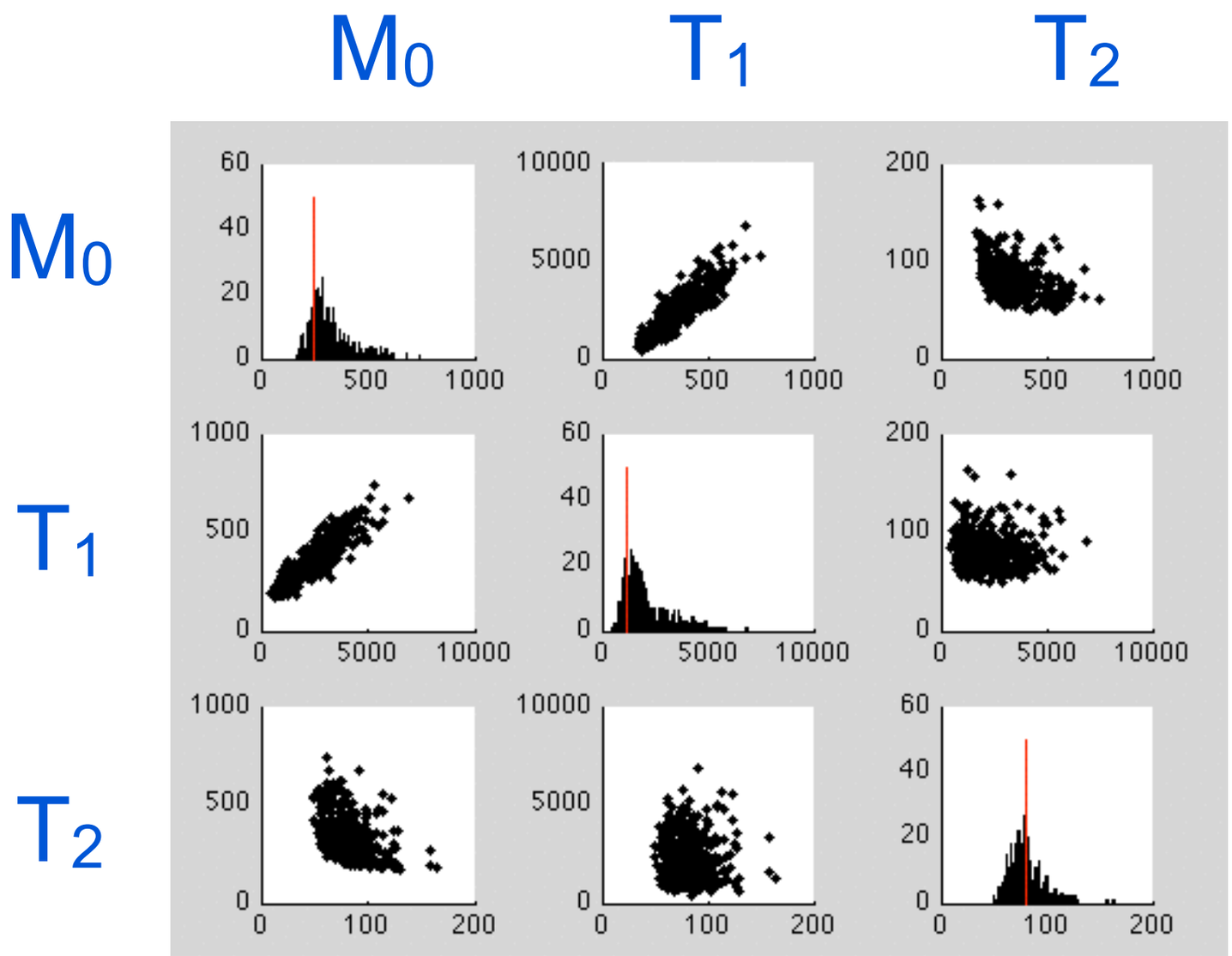
straight into MH!

TR = 1, 1.6, 2.3, 4 s

TE = 30, 47, 65, 82, 100 ms

simple_mri_example.m

$Y = M_0 * \exp(-TE/T_2) * (1-\exp(-TR/T_1))$



SNR=20

Lesson: Bayes can tell us about model structure (e.g. correlation between parameters) and estimability **given data**

Bayesian Inference

- Exact inference

  Analytic integrals (gaussian - or nasty)
  Grid evalutation (1-5 parameters, max)

- Sampling

  Metropolis Hastings (the most flexible, please use my tool!)
  Gibbs sampling (need conditional posteriors to be nice)

- Analytic approximations

  Laplace Approximation (local = curvature of posterior)
  Variational Bayes (Kullback Leibler = approximate whole distribution)

ONBI - Bayesian modelling practical. 2014/15

Practical Overview

This practical requires Matlab. Go through the instructions and execute the listed commands in the Matlab command window (you can copy-paste). Raise your hand if you need help, but perhaps try first the "help" command in Matlab if you are unsure about Matlab syntax issues.

Contents:

Linear models

This practical will cover most of the examples that were discussed in the lecture, and you should be able to reproduce all the figures that were shown in the lecture.

We start with the simplest possible example of Bayesian inference. Imagine that we are measuring a certain quantity (e.g. temperature using a thermometer, actually let us say we ARE measuring temperature just for concreteness). We are going to assume that there exists a true unknown temperature, and that our measurement device is noisy. Therefore, we are going to make several measurements and we are going to use Bayesian inference.

We start by writing a generative model. Assuming we are measuring the temperature directly, the generative model can simply write:

```
y = a + noise;
```

In the above equation, `y` is the data, `a` is a parameter of the model (the true unknown temperature), and the noise is assumed to be additive.

We further need to make some distributional assumption on the noise. Most commonly, the noise is assumed Gaussian with mean zero and standard deviation `s`.