Nonlinear models

Will Penny

Bayesian Inference Course,
WTCN, UCL, March 2013

Nonlinear Regression

We consider the framework implemented in the SPM function *spm-nlsi-GN.m*. It implements Bayesian estimation of nonlinear models of the form

$$y = g(w) + e$$

where $g(w)$ is some nonlinear function of parameters w, and e is zero mean additive Gaussian noise with covariance C_y. The likelihood of the data is therefore

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

The error *precision* matrix is assumed to decompose linearly

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

where Q_i are known precision basis functions and λ are hyperparameters eg $Q = I$, noise precision $s = \exp(\lambda)$.

Priors

We allow Gaussian priors over model parameters

$$p(w) = \mathrm{N}(w; \mu_w, C_w)$$

where the prior mean and covariance are assumed known.

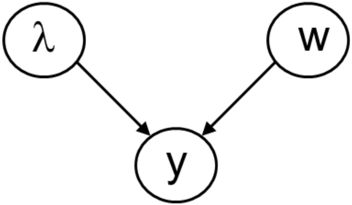
The hyperparameters are constrained by the prior

$$p(\lambda) = \mathrm{N}(\lambda; \mu_\lambda, C_\lambda)$$

This is not Empirical Bayes.

Generative Model

VL Generative Model



$$p(y, w, \lambda) = p(y|w, \lambda)p(w)p(\lambda)$$

Energies

The above distributions allow one to write down an expression for the joint log likelihood of the data, parameters and hyperparameters

$$L(w,\lambda) = \log[p(y|w,\lambda)p(w)p(\lambda)]$$

It splits into three terms

$$\begin{aligned} L(w,\lambda) &= \log p(y|w,\lambda) \\ &+ \log p(w) \\ &+ \log p(\lambda) \end{aligned}$$

Joint Log Likelihood

The joint log likelihood is composed of sum squared precision weighted prediction errors and entropy terms

$$\begin{aligned} L(w,\lambda) &= -\frac{1}{2}e_y^T C_y^{-1} e_y - \frac{1}{2}\log|C_y| - \frac{N_y}{2}\log 2\pi \\ &- \frac{1}{2}e_w^T C_w^{-1} e_w - \frac{1}{2}\log|C_w| - \frac{N_w}{2}\log 2\pi \\ &- \frac{1}{2}e_\lambda^T C_\lambda^{-1} e_\lambda - \frac{1}{2}\log|C_\lambda| - \frac{N_\lambda}{2}\log 2\pi \end{aligned}$$

where prediction errors are the difference between what is expected and what is observed

$$\begin{aligned} e_y &= y - g(m_w) \\ e_w &= m_w - \mu_w \\ e_\lambda &= m_\lambda - \mu_\lambda \end{aligned}$$

VL Posteriors

The Variational Laplace (VL) algorithm, implemented in *spm-nlsi-GN.m*, assumes an approximate posterior density of the following factorised form

$$\begin{aligned} q(w,\lambda|y) &= q(w|y)q(\lambda|y) \\ q(w|y) &= \mathrm{N}(w; m_w, S_w) \\ q(\lambda|y) &= \mathrm{N}(\lambda; m_\lambda, S_\lambda) \end{aligned}$$

This is a fixed-form variational method.

Variational Energies

The approximate posteriors are estimated by minimising the Kullback-Liebler (KL) divergence between the true posterior and these approximate posteriors. This is implemented by maximising the following (negative) variational energies

$$\begin{aligned} I_w &= \int L(w,\lambda) q(\lambda) d\lambda \\ I_\lambda &= \int L(w,\lambda) q(w) dw \end{aligned}$$

Gradient Ascent

This maximisation is effected by first computing the gradient and curvature of the variational energies at the current parameter estimate, $m_w(old)$. For example, for the parameters we have

$$
\begin{aligned}
j_w(i) &= \frac{dI_w}{dm_w(i)} \\
H_w(i,j) &= \frac{d^2 I_w}{dm_w(i)dm_w(j)}
\end{aligned}
$$

where i and j index the ith and jth parameters, j_w is the gradient vector and H_w is the curvature matrix. The estimate for the posterior mean is then given by

$$m_w(new) = m_w(old) + \Delta m_w$$

Adaptive Step Size

The change is given by

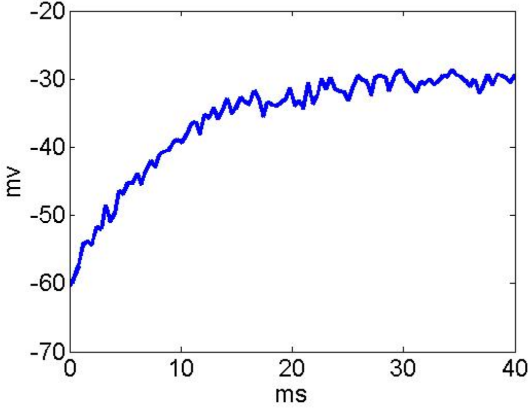
$$\Delta m_w = -H_w^{-1} j_w$$

which is equivalent to a Newton update (Press et al. 2007).

This implements a step in the direction of the gradient with a step size given by the inverse curvature. Big steps are taken in regions where the gradient changes slowly (low curvature).

Approach to Limit Example

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



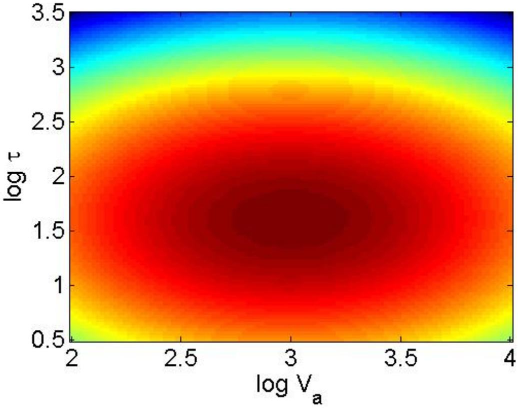
$$V_a = 30, \tau = 8$$

Noise precision

$$s = \exp(\lambda) = 1$$

Prior Landscape

A plot of $\log p(w)$ where $w = [\log \tau, \log V_a]$

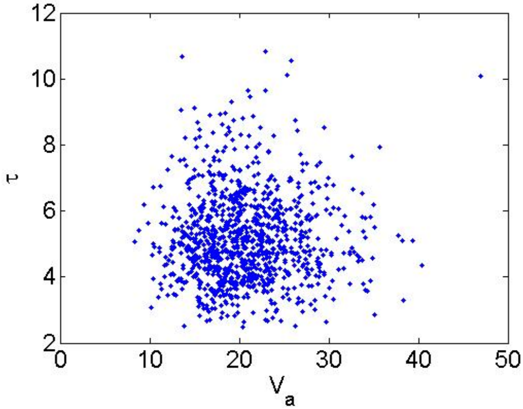


$$\mu_w = [3, 1.6]^T, C_w = diag([1/16, 1/16]);$$

Samples from Prior

The true model parameters are unlikely apriori

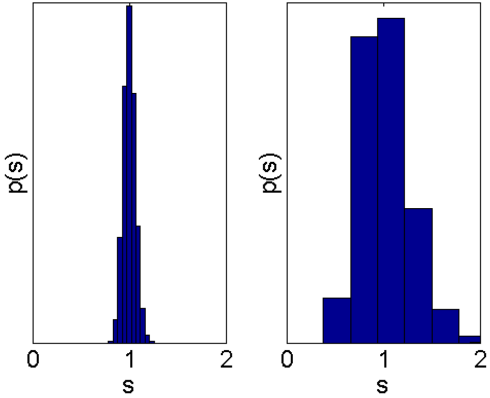
$$V_a = 30, \tau = 8$$



Prior Noise Precision

$Q = I$. Noise precision $s = \exp(\lambda)$ with

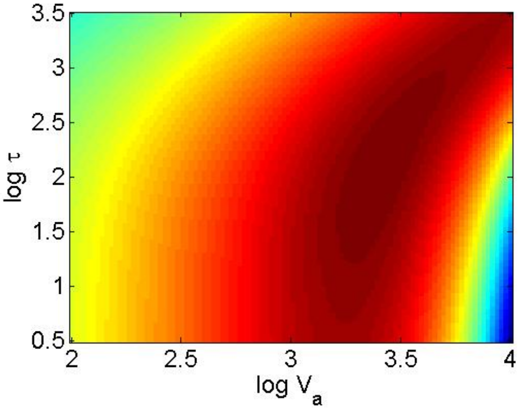
$$p(\lambda) = \mathrm{N}(\lambda; \mu_\lambda, C_\lambda)$$



with $\mu_\lambda = 0$. We used $C_\lambda = 1/16$ (left) and $C_\lambda = 1/4$ (right). True noise precision, $s = 1$.

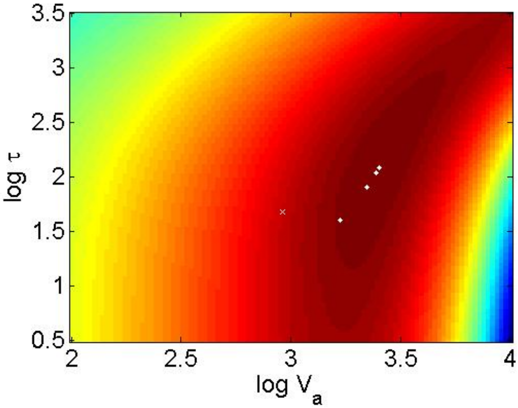
Posterior Landscape

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



VL optimisation

Path of 6 VL iterations (x marks start)



Investigate further using *matlab/lif*.

Oscillator Example

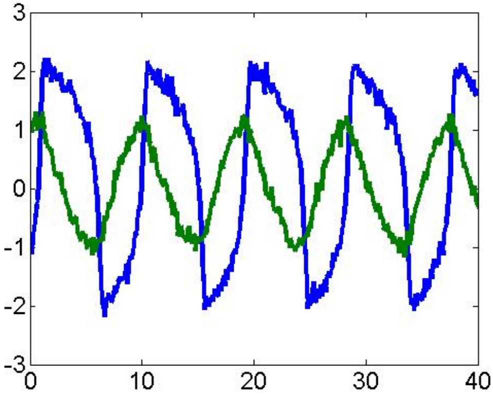
This example is based on a differential equation describing the evolution of a voltage variable, v, and a recovery variable, r

$$\begin{aligned} \dot{v} &= c[v - \frac{1}{3}v^3 + r + I] \\ \dot{r} &= -\frac{1}{c}[v - a + br] \end{aligned}$$

This is used in statistics as an example of a difficult optimisation algorithm with multiple local maxima Ramsay et al. (2007).

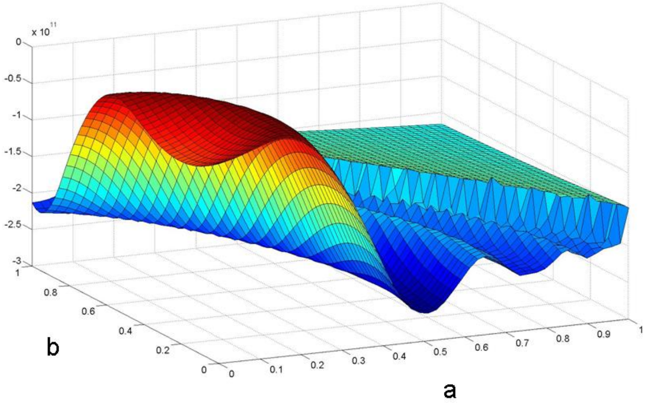
Oscillator Example

For $a = 0.2$, $b = 0.2$, $c = 3$ and $I = 0$



Oscillator Example

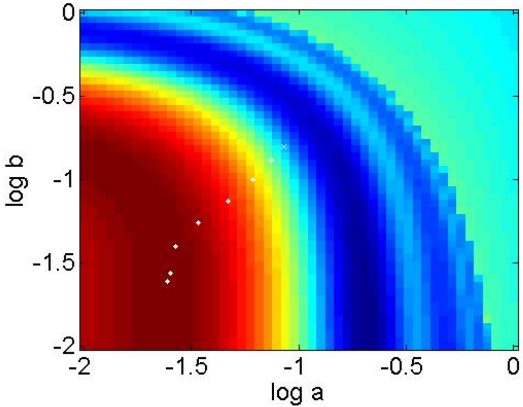
A plot of $\log[p(y|w)p(w)]$



Parameters $w = [a, b]$. Fix $l = 0$, $c = 3$.

Oscillator Example

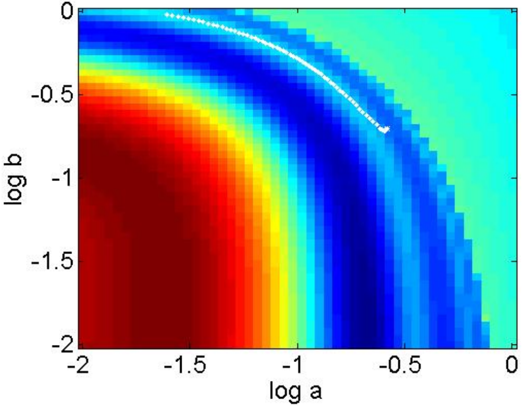
A plot of $\log[p(y|w)p(w)]$



Global maxima

Oscillator Example

Local maxima



Potential solutions

There are a number of potential solutions

- Increase the dimension of the space (from a,b to a,b,c).
- Fit data in the frequency domain rather than time domain
- Fit other features of the data
- Use sampling methods

Metropolis-Hastings

MH creates as series of random points $(w(1), w(2), ...)$ whose distribution converges to the target distribution of interest. For us, this is the posterior density $p(w|y)$. Each sequence can be considered a random walk whose stationary distribution is $p(w|y)$.

MH makes use of a proposal density $q(w'; w)$ which is dependent on the current state vector w. For symmetric q (such as a Gaussian) samples from the posterior density can be generated as follows.

MH update

First, start at some point $w(0)$ in parameter space. Then generate a proposal w' using the density q. This proposal is then accepted according to the standard Metropolis-Hastings procedure.

That is, with probability $\min(1, r)$ where

$$r = \frac{p(y|w^{'})p(w^{'})}{p(y|w)p(w)}$$

If the step is accepted we set $w(n+1) = w'$. If it is rejected we set $w(n+1) = w(n)$.

Adaptive proposal density

We use a zero mean Gaussian proposal density with covariance C_s. This covariance is initialised to

$$C_s = \sigma C_w$$

where C_w is the prior covariance and $\sigma = 1$.

We then use a three stage procedure comprising (i) scaling, (ii) tuning and (iii) sampling steps in which the scaling and tuning stages are used to optimize the proposal covariance C_s.

The first two stages are regarded as a burn-in phase and samples from this period are later discarded. At the end of this C_s is fixed and sampling proper begins.

Scaling

The proposal covariance is given by

$$C_s = \sigma C_w$$

In the scaling step σ is adjusted as follows.

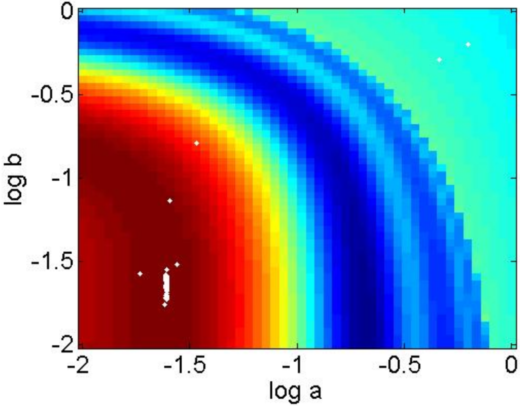
If the acceptance rate, as measured over the last $n_s = 100$ proposals, is less than 20 per cent then σ is halved.

If the acceptance rate is greater than 40 per cent σ is doubled.

Otherwise, σ remains unchanged.

MH - Scaling

Init: $[-0.2, -0.2]$. Then 1000 samples



Tuning

The tuning step makes use of adaptive estimation of a covariance matrix C_{tune} based on a Robbins-Monro update.

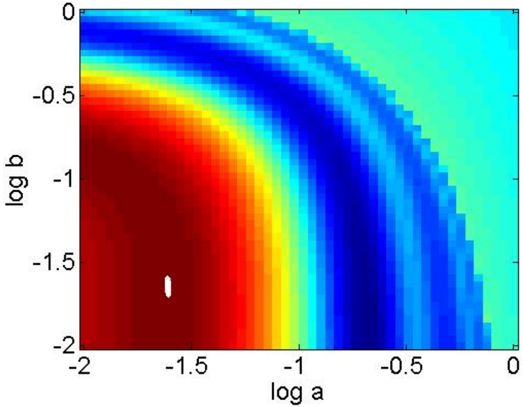
At the beginning of the tuning stage we set $C_{tune} = C_s$. We then update according to

$$\begin{aligned} \mu_t &= \mu_{t-1} + \frac{1}{n_t}(x_t - \mu_t) \\ \Delta C_{tune} &= \frac{1}{n_t}[(x_t - \mu_t)(x_t - \mu_t)^T - C_{tune}(t-1)] \end{aligned}$$

where n_t is the number of elapsed iterations in the tuning period. At the end of tuning set $C_s = C_{tune}$.

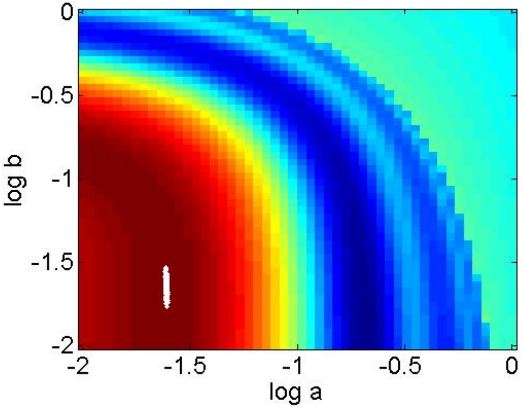
MH - Tuning

1000 samples



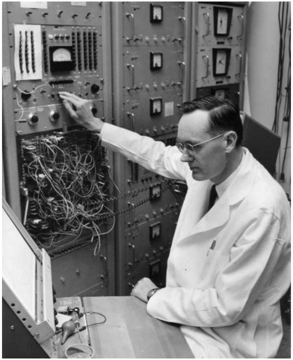
MH - Sampling

2000 samples



Potential solutions

Accept that a nonlinear dynamical system model has such a rich repertoire of behaviour, that a model cannot be specified by a dynamical equation alone. One must also specify the range of allowable parameters.



$$
\begin{aligned}
\dot{v} &= c[v - \frac{1}{3}v^3 + r + I] \\
\dot{r} &= -\frac{1}{c}[v - a + br]
\end{aligned}
$$

Fitzhugh-Nagumo

I is input current.

$$\begin{aligned} \dot{v} &= c[v - \frac{1}{3}v^3 + r + I] \\ \dot{r} &= -\frac{1}{c}[v - a + br] \end{aligned}$$

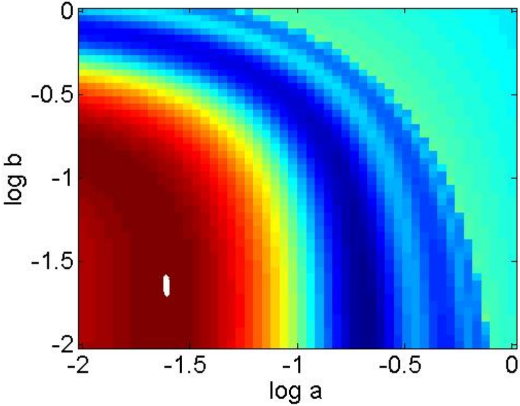
For $I = 0$ the cell should not spike (need stable fixed point at $v = 0$).

For I above some threshold there should be an unstable fixed point around which a limit cycle emerges (spiking).

Fitzhugh-Nagumo

This occurs if these 3 conditions are satisfied

- $1 - \frac{2b}{3} < a < 1$
- $0 < b < 1$
- $b < c^2$



References

R. Fitzhugh (1961) Impulses and physiological states in theoretical models of nerve membrane. Biophysical Journal, 1:445-466.

K. Friston et al. (2007) Variational Free Energy and the Laplace Approximation. Neuroimage 34(1), 220-234.

A. Gelman et al. (1995) Bayesian data analysis. Chapman and Hall.

W. Press et al. (2007) Numerical recipes in C: the art of scientific computing. 3rd Edition, Cambridge University Press.

Ramsay et al. (2007) Parameter estimation for differential equations: a generalized smoothing approach. J. Roy. Stat. Soc. B, 69(5),741-796.

B. Ermentrout and D. Terman. Mathematical Foundations of Neuroscience. Springer, 2010.