Monte Carlo methods for Inverse Problems

Marko Laine

ILMATIETEEN LAITOS
METEOROLOGISKA INSTITUTET
FINNISH METEOROLOGICAL INSTITUTE

HY 2016-02-24

Outline

Introduction - sources of uncertainties

Markov chain Monte Carlo – MCMC

Some MCMC theory

Toy example: Stochastic Lorenz 95 model

Remote sensing of greenhouse gases. From ozone to methane and carbon dioxide.

Sources of uncertainties in modelling

uncertainty	*source*	*methods*
Observation	instrument noise, sampling, representation, retrieval	sampling design retrieval method
Parameter	estimation, calibration, tuning	optimal estimation MCMC
Model formulation	approximate physics, numerics, resolution, sub-grid scale processes	model diagnostics model selection averaging Gaussian processes
Initial value	state space models	Kalman filter assimilation

MCMC for complex model

CPU demanding models, atmosphere, climate, weather

- Efficient adaptive MCMC.
- Parallel chains, "tricks" like early rejection.

High dimension of the unknown, inverse problems, profile estimation

- Regularization by smoothness priors.
- Dimension reduction.

Chaotic behaviour of models, model error.

- Assimilation, Kalman filter.
- Models are right, but not very accurate.

High dimensional spaces are very empty

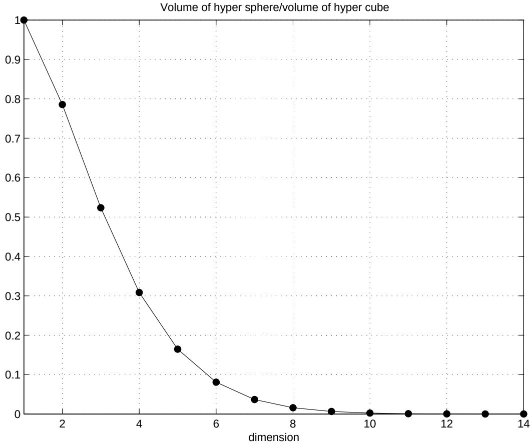
The plot shows the volume of a hyper sphere

$$\frac{2\pi^{d/2} r^d}{d\Gamma(d/2)}$$

divided by the volume of a hyper cube

$$(2r)^d.$$

Random walk type methods are needed to explore the space of statistical significant probability. Otherwise we will always be lost at some distant corners.



Terminology

- Observations y_t, model states x_t, parameters θ.
- State space representation.

$$\begin{aligned} y_t &= \mathcal{F}(x_t, \theta) + \epsilon_t && \text{obsevation equation} \\ x_t &= \mathcal{M}(x_{t-1}, \theta) + E_t && \text{model evolution} \end{aligned}$$

- Hierarchical statistical model

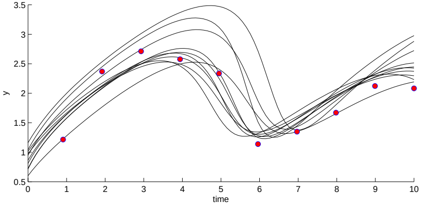
$$\begin{aligned} & p(y_t|x_t, \theta) && \text{observation model} \\ & p(x_t|x_{t-1}, \theta) && \text{process model} \\ & p(\theta) && \text{parameter model} \end{aligned}$$

- Bayes formula.

$$p(x_{1:n}, \theta|y_{1:n}) \propto \prod_{t=1}^{n} p(y_t|x_t, \theta) p(x_t|x_{t-1}, \theta) p(\theta)$$

Statistical analysis by simulation

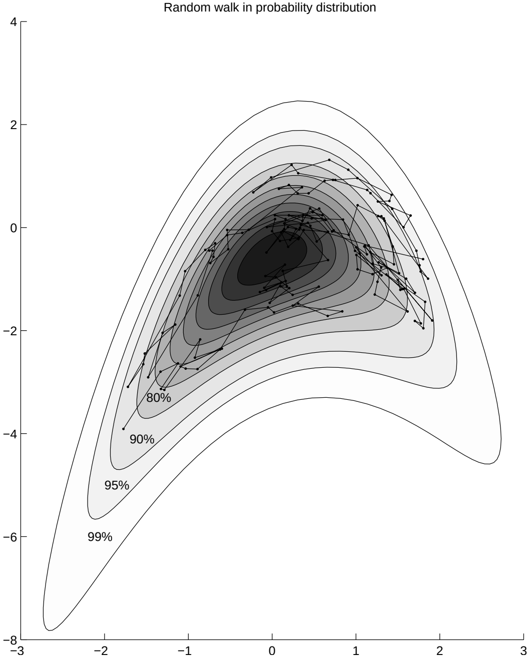
- We are interested in the uncertainty distribution of the unknown model parameter vector θ given the observational data y and the model: $p(\theta|y, M)$.
- This distribution is typically analytically intractable.
- We can still simulate observations from $p(y|\theta, M)$.



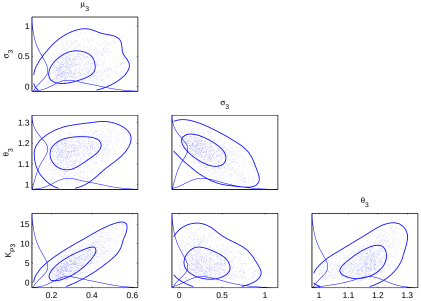
- Statistical inference is used to define what is a good fit. Parameters that are consistent with the data and the modelling uncertainty are accepted.

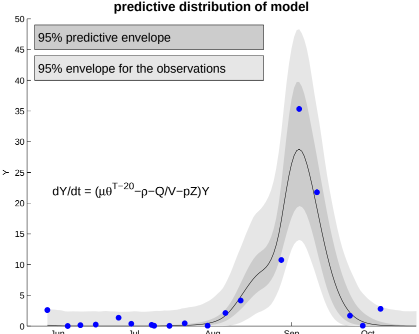
Markov chain Monte Carlo – MCMC

- Simulate the model while sampling the parameters from a *proposal distribution*.
- Accept (or weight) the parameters according to a suitable goodness-of-fit criteria depending on *prior information* and error statistics defining the *likelihood function*.
- The resulting chain is a sample from the Bayesian *posterior distribution* of parameter uncertainty.



Posterior distributions





While sampling the model using the MCMC, we get:

- Posterior distribution of model parameters.
- Posterior distribution of model predictions.
- Posterior distribution for model comparison.

Terminology for modelling with MCMC methods

The observation model (in non state space form) is

$$y = f(x|\theta) + \epsilon,$$
$$\text{observations} = \text{model} + \text{error}.$$

Likelihood function for independent Gaussian errors corresponds to a simple quadratic *cost function*, with

$$\begin{aligned} p(y|\theta) &\propto \exp\left\{-\frac{1}{2}\frac{\sum_i^n (y_i - f(x_i|\theta))^2}{\sigma^2}\right\} \\ &= \exp\left\{-\frac{1}{2}\frac{SS(\theta)}{\sigma^2}\right\}, \end{aligned}$$

where $SS(\theta) = -2\log(p(y|\theta))$, the log-likelihood in "sum-of-squares" cost function format. For calculating the *posterior*, we also need to account $SS_{\mathrm{pri}}(\theta) = -2\log(p(\theta))$, the *prior* "sum-of-squares".

Metropolis-Hastings algorithm

Random walk Metropolis-Hastings algorithm with Gaussian *proposal distribution* (and Gaussian likelihood).

- Propose new parameter value $\theta_{\mathrm{prop}} = \theta_{\mathrm{curr}} + \xi$, where $\xi \sim N(0, \Sigma_{\mathrm{prop}})$ is drawn from the *proposal distribution*.
- Accept θ_{prop} with probability α,

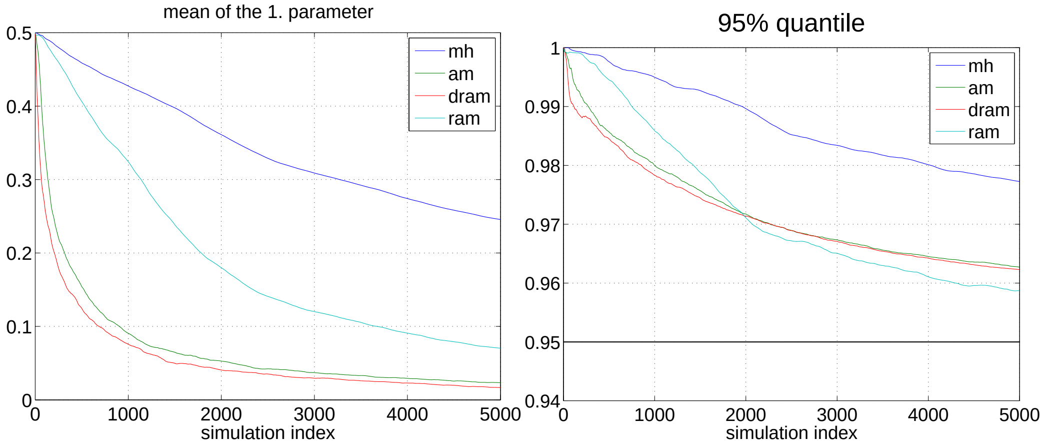
$$\begin{aligned}\alpha(\theta_{\mathrm{curr}}, \theta_{\mathrm{prop}}) = 1 \wedge \exp\Bigg\{ & -\frac{1}{2}\left(\frac{SS(\theta_{\mathrm{prop}}) - SS(\theta_{\mathrm{curr}})}{\sigma^2}\right) \\ & -\frac{1}{2}\Big(SS_{\mathrm{pri}}(\theta_{\mathrm{prop}}) - SS_{\mathrm{pri}}(\theta_{\mathrm{curr}})\Big)\Bigg\}\end{aligned}$$

- Efficient proposal distribution ⇒ *adaptive tuning* of Σ_{prop} by AM and DRAM algorithms.

Haario, et al, Stat.Comp. 2006.

Short chains and adaptation

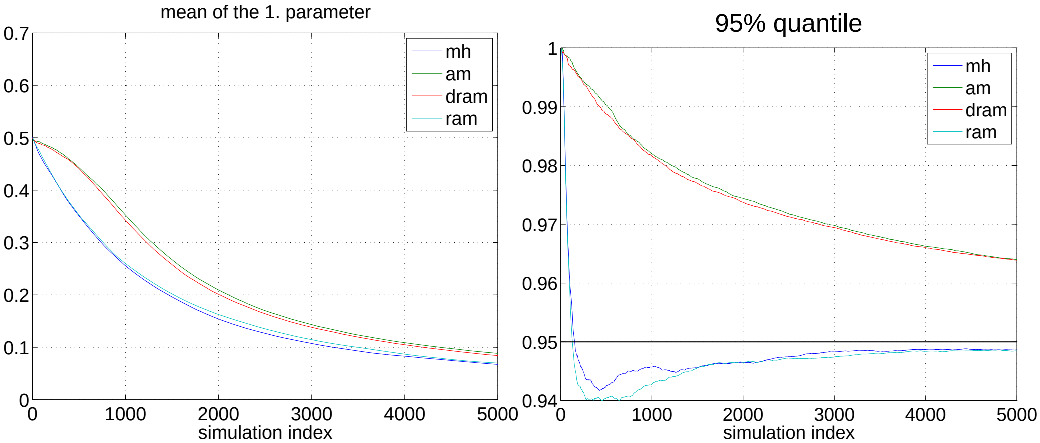
- It is important to make short chains as efficient as possible. Efficient: produce estimates with small *Monte Carlo error*.



Short MCMC chain repeated 1000 times with different algorithms, Gaussian 10 dimensional target and too large initial covariance.

Short chains and adaptation

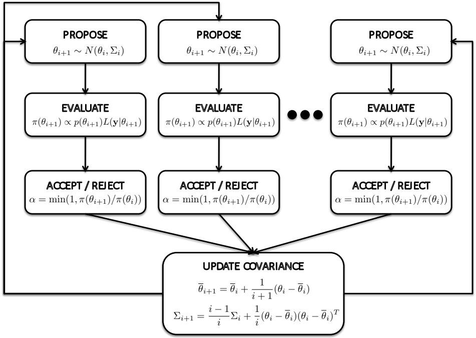
- But, adaptation might slow the convergence.



Same as in the previous slide, but now with more optimal initial proposal, Gaussian 10 dimensional target, near optimal initial covariance.

Faster MCMC: parallel chains

- Random walk MCMC is by nature sequential, and it is generally more efficient to run one long chain than many short independent chains.
- In *parallel* adaptive MCMC, the adaptation is done over the points in all chains and they share one common adapted proposal covariance.
- Communication between the chains can be asynchronous.



PROPOSE

$\theta_{i+1} \sim N(\theta_i, \Sigma_i)$

EVALUATE

$\pi(\theta_{i+1}) \propto p(\theta_{i+1}) L(\mathbf{y}|\theta_{i+1})$

ACCEPT / REJECT

$\alpha = \min(1, \pi(\theta_{i+1})/\pi(\theta_i))$

PROPOSE

$\theta_{i+1} \sim N(\theta_i, \Sigma_i)$

EVALUATE

$\pi(\theta_{i+1}) \propto p(\theta_{i+1}) L(\mathbf{y}|\theta_{i+1})$

ACCEPT / REJECT

$\alpha = \min(1, \pi(\theta_{i+1})/\pi(\theta_i))$

PROPOSE

$\theta_{i+1} \sim N(\theta_i, \Sigma_i)$

EVALUATE

$\pi(\theta_{i+1}) \propto p(\theta_{i+1}) L(\mathbf{y}|\theta_{i+1})$

ACCEPT / REJECT

$\alpha = \min(1, \pi(\theta_{i+1})/\pi(\theta_i))$

UPDATE COVARIANCE

$$\overline{\theta}_{i+1} = \overline{\theta}_i + \frac{1}{i+1}(\theta_i - \overline{\theta}_i)$$

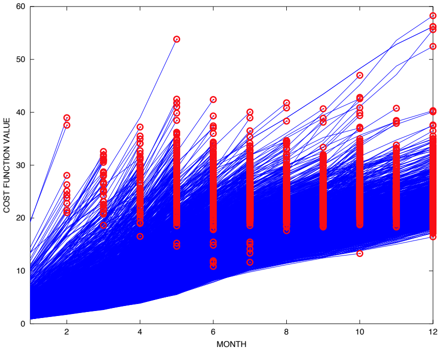
$$\Sigma_{i+1} = \frac{i-1}{i}\Sigma_i + \frac{1}{i}(\theta_i - \overline{\theta}_i)(\theta_i - \overline{\theta}_i)^T$$

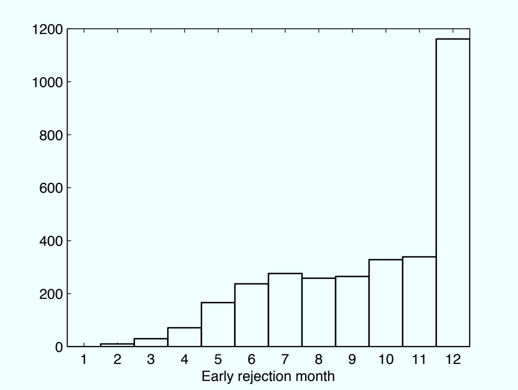
Faster MCMC: early rejection

Idea: evaluate the likelihood in parts and check after each part if the proposed parameter value can be rejected.

Cumulative cost function evaluated after each month

during one year climate model simulation





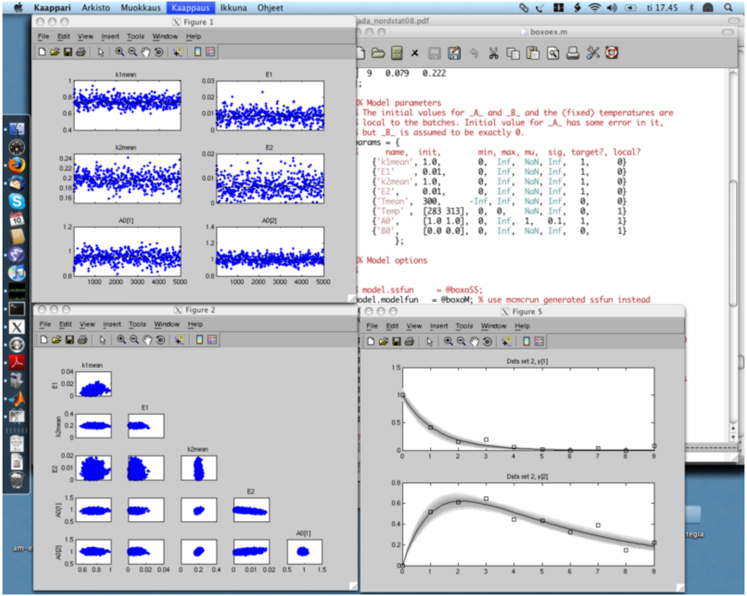
time to stop the simulation

proportion of stopped runs by month

This simple trick saved 10%–80% of CPU time in different test cases.

Solonen, et al, Bayesian Analysis, 2012.

MCMC Toolbox for Matlab



```
] 9   0.079   0.222
];

%% Model parameters
% The initial values for _A_ and _B_ and the (fixed) temperatures are
% local to the batches. Initial value for _A_ has some error in it,
% but _B_ is assumed to be exactly 0.
params = {
%       name,  init,         min, max, mu,  sig, target?, local?
   {'k1mean', 1.0,          0,  Inf,  NaN, Inf,   1,       0}
   {'E1'    , 0.01,         0,  Inf,  NaN, Inf,   1,       0}
   {'k2mean', 1.0,          0,  Inf,  NaN, Inf,   1,       0}
   {'E2',     0.01,         0,  Inf,  NaN, Inf,   1,       0}
   {'Tmean',  300,       -Inf,  Inf,  NaN, Inf,   0,       0}
   {'Temp' ,  [283 313],  0,  0,      NaN, Inf,   0,       1}
   {'A0',     [1.0 1.0],  0,  Inf,    1,   0.1,   1,       1}
   {'B0',     [0.0 0.0],  0,  Inf,    NaN, Inf,   0,       1}
         };

%% Model options
%

% model.ssfun     = @boxoSS;
model.modelfun   = @boxoM; % use mcmcrun generated ssfun instead
```

http://helios.fmi.fi/~lainema/mcmc/

MCMC toolbox for matlab

```
model.ssfun = @mycostfun

data = load(’datafile.dat’);

parameters = {
  {’par1’, 2.3 }
  {’par2’, 1.2 }
};

options.nsimu  = 5000;
options.method = ’am’;

[results,chain] = mcmcrun(model,data,parameters,options);

mcmcplot(chain,[],results)
```

Some MCMC theory

- Let θ be a parameter vector having values in a parameter space Θ, indexing family of possible probability distributions

$$p(y|\theta)$$

  describing our observations y.

- If $p(\theta)$ is a prior probability density describing our prior beliefs about θ, then Bayes formula gives the posterior π in terms of likelihood and prior

$$\pi(\theta) := p(\theta|y) = \frac{p(y|\theta)p(\theta)}{\int p(y|\theta)p(\theta)\, d\theta}$$

- In Markov chain Monte Carlo methods (MCMC) we construct a Markov chain which has our parameter space Θ as its state space and π as its limiting stationary distribution.
- That means we have a way of sampling values from posterior distribution π and therefore make Monte Carlo inference about θ in form of sample averages and density estimates.

Markov chain Monte Carlo – MCMC

- A Markov chain is described by a transition kernel $P(\theta, d\theta')$ that gives for each state θ the probability distribution for the chain to move to state $d\theta'$ in next step. For ease of exposition let us assume that for each distribution the corresponding density function exists and denote the transition density as $p(\theta, \theta')$.
- MCMC methods produce chains that are aperiodic, irreducible and fulfill a reversibility condition, also called 'detailed balance equation':

$$\pi(\theta)p(\theta, \theta') = \pi(\theta')p(\theta', \theta) \quad \theta, \theta' \in \Theta.$$

- If π is the initial distribution of the starting state, then the intensity of going from state θ to state θ' is same as that of going from θ' to θ.
- Direct consequence of the reversibility is

$$\int \pi(\theta)p(\theta, \theta')\, d\theta = \pi(\theta'), \text{ for all } \theta' \in \Theta$$

that means that π is in fact the stationary distribution of the chain and we can use a sample from the chain as a random sample from distribution π.

The Metropolis-Hastings algorithm

- In Metropolis-Hastings algorithm we generate a Markov chain with transition density

$$p(\theta,\theta') = q(\theta,\theta')\alpha(\theta,\theta'), \quad \theta \neq \theta'$$
$$p(\theta,\theta) = 1 - \int q(\theta,\theta')\alpha(\theta,\theta')\, d\theta$$

  for some *proposal density* q and for *acceptance probability* α.

- The chain is reversible if and only if

$$\pi(\theta)q(\theta,\theta')\alpha(\theta,\theta') = \pi(\theta')q(\theta',\theta)\alpha(\theta',\theta).$$

- Which leads to choose α as

$$\alpha(\theta,\theta') = \min\left\{1, \frac{\pi(\theta')q(\theta',\theta)}{\pi(\theta)q(\theta,\theta')}\right\}.$$

- Usually $\Theta \subset \mathbb{R}^d$, but the reversibility condition can be formulated in more general state space.

Notes:

- In MH algorithm we need to calculate the posterior ratio $\pi(\theta')/\pi(\theta)$ in the formula for α but Bayes formula gives this in terms of likelihood and prior as

$$\frac{p(y|\theta')p(\theta')}{p(y|\theta)p(\theta)}$$

  and the constant of proportionality disappears.
- We need some theory of Markov chains but with important simplifications: we know by construction that the stationary distribution π exists. Also we are able to choose the initial distribution as we like. That gives us simple ways to prove important ergodic properties of the MH chain: The law of large numbers that gives us permission to use sample averages as estimates and the central limit theorem which gives us the convergence rate for the algorithms.

The algorithm

1. Choose initial values θ_0 and proposal density q.
2. Using current value of the chain θ_i propose a new value θ' using proposal distribution $q(\theta_i, \cdot)$.
3. Generate a random number u uniform on $[0, 1]$ and accept the new value if

$$u \leqslant \min\left\{1, \frac{\pi(\theta')q(\theta', \theta_i)}{\pi(\theta_i)q(\theta_i, \theta')}\right\}.$$

4. If accepted set $\theta_{i+1} = \theta'$, if not $\theta_{i+1} = \theta_i$.
5. Go to (ii) until enough values have been sampled.

Nonlinear model fitting

- Consider a nonlinear model describing observations y by control variables x and parameter vector θ:
$$y = f(x, \theta) + \epsilon, \quad \epsilon \sim N(0, I\sigma^2).$$
- In 'classical' theory we find the optimal θ by minimizing the sum of squares
$$SS = \sum_{i=1}^{n} (y_i - f(x_i, \theta))^2$$
which leads to a nonlinear minimization problem. Confidence regions for θ are usually obtained by linearizing the likelihood function and by asymptotic arguments.
- In Bayesian approach we can get a similar fit by using non-informative uniform priors for the parameter θ and the inference is done with the posterior distribution of θ obtained with MCMC.
- For the error variance a convenient choice for prior information is thru 'precision' $\tau = \sigma^{-2}$:
$$p(\tau) \sim \Gamma(\frac{n_0}{2}, \frac{n_0}{2} S_0^2)$$

Implementation

Just need to write a routine to return the sum-of-squares for given parameter and data.

1. Initialization. Initial values for θ, n_0 and S_0^2. Proposal distribution q. Adaptive strategies for q useful here.
2. MH step: Generate a new value for θ from q and calculate sum-of-squares for it, SS_{new}. New value is accepted if $SS_{\mathrm{new}} < SS_{\mathrm{old}}$ or if

   $$u \leqslant \exp\left\{-\frac{1}{2\sigma^2}(SS_{\mathrm{new}} - SS_{\mathrm{old}})\right\}$$

   where $u \sim U(0,1)$ and σ^2 is current value of the error variance.
3. Update σ^{-2} with a draw from

   $$\Gamma(\frac{n_0 + n}{2}, \frac{n_0 S_0^2 + SS}{2}).$$

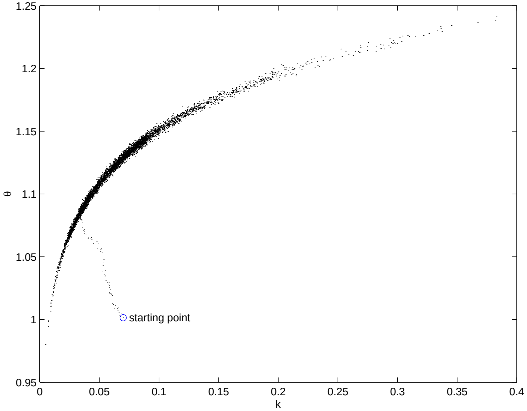
4. Go to (ii) until enough values have been sampled.

Example: Oxygen consumption in lake Tuusulanjärvi

- Model:

$$\frac{d[O_2]}{dt} = -k[O_2]\theta^{T-20}$$

- Data: measured oxygen concentrations and temperatures during ice season.
- Parameters: k, oxygen consumption rate, θ, temperature coefficient.
- On right: a plot of the generated MCMC chain for the two parameters.



Reversible jump Metropolis-Hastings algorithm

- The detailed balance equation can also be formulated in general state space. For the Metropolis-Hastings algorithm to work it must only accept states from where there is a positive probability to do a reversible move back to the original state.
- For the reversible jump Metropolis-Hastings algorithm the state space is written as

  $$E = \left\{ (k, \theta^{(k)}), k \in \mathcal{K}, \theta^{(k)} \in \Theta_k \right\},$$

  where $\mathcal{K}$ is enumerable model space and $\theta^{(k)} \in \Theta_k$ is the parameter space of model k. The dimension of $\theta^{(k)}$ can vary with k.
- The posterior distribution can be factorized as

  $$\pi(\theta^{(k)}, k) = \pi(\theta^{(k)}|k)\pi(k)$$

  and we might be interested in the posterior probabilities of different models $\pi(k)$ of draw some conditional or marginal conclusions about different models in terms of $\pi(\theta^{(k)}|k)$ or $\pi(\theta^{(k)})$.

Implementing RJMCMC

- We need a reversible move between models i and j. This is accomplished by a bijective function g_{ij} that transforms the parameters

$$g_{ij}(\theta^{(i)}, u^{(i)}) = (\theta^{(j)}, u^{(j)}),$$

and retains the dimensions

$$d(\theta^{(i)}) + d(u^{(i)}) = d(\theta^{(j)}) + d(u^{(j)}).$$

- The inverse transform $g_{ij}^{-1} =: g_{jk}$ gives the move to the other direction. Variables u and u' are random quantities used in proposing change in the components and as extra components when going to a higher dimension.
- If $q_{ij}(\theta^{(i)}, u^{(i)})$ is the probability density for the proposed move and $p(i,j)$ is the probability for the move $i \to j$ the accepting probability can be written as

$$\alpha_{ij}(\theta^{(i)}, \theta^{(j)}) =$$

$$\min\left\{1, \frac{\pi_j(\theta^{(j)})p(j,i)q_{ji}(\theta^{(j)}, u^{(j)})}{\pi_i(\theta^{(i)})p(i,j)q_{ij}(\theta^{(i)}, u^{(i)})} \left|\frac{\partial(\theta^{(j)}, u^{(j)})}{\partial(\theta^{(i)}, u^{(i)})}\right|\right\}. \quad (**)$$

Step of the algorithm:

When being in model k_i with parameter vector $\theta_i^{(k_i)}$:

1. Choose a new model j by drawing it from distribution $p(i, \cdot)$. Propose a value for the parameter $\theta^{(j)}$ by generating u from distribution $q_{k_i j}(\theta_i^{(k_i)}, u)$.
2. Accept the move with probability (**): $k_{i+1} = j$ and $\theta_{i+1}^{(k_{i+1})} = \theta^{(j)}$.
3. If the move is not accepted, stay in the current model: $k_{i+1} = k_i$ and $\theta_{i+1}^{(k_{i+1})} = \theta_i^{(k_i)}$.

In step (i) it is also possible to choose stay in the current model and do a standard Metropolis-Hastings step.

RJMCMC Example

Model 1: $(\theta_1, \theta_2) \in \mathbb{R}^2$, Model 2: $\theta \in \mathbb{R}$.
Move 1 → 2

$$g_{12}((\theta_1, \theta_2)) = \left(\frac{\theta_1 + \theta_2}{2}, \frac{\theta_1 - \theta_2}{2} \right) = (\theta, u).$$

Move 2 → 1

$$g_{21}(\theta, u) = (\theta + u, \theta - u).$$

Make a move to another model with probability $\frac{1}{2}$. Draw a random number u from distribution q. Ratios used in acceptance are now $(\theta_1, \theta_2) \to \theta$:

$$\frac{\pi_2(\theta)\frac{1}{2}q(u)}{\pi_1(\theta_1, \theta_2)\frac{1}{2}} \left| \frac{\partial(\theta, u)}{\partial(\theta_1, \theta_2)} \right| = \frac{\pi_2(\frac{\theta_1+\theta_2}{2})q(\frac{\theta_1-\theta_2}{2})}{\pi_1(\theta_1, \theta_2)} \frac{1}{2}.$$

and for $\theta \to (\theta_1, \theta_2)$:

$$\frac{\pi_1(\theta + u, \theta - u)}{\pi_2(\theta)q(u)} 2.$$

This is reasonable if the ratio π_1/π_2 is easy to calculate.

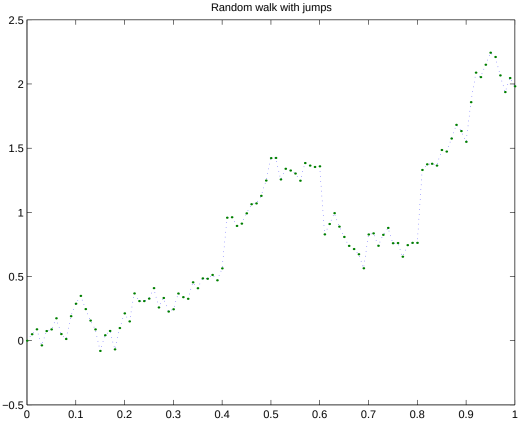
RJMCMC Example (cont.)

Standard 1 dimensional random walk, with 'extra' jumps or changes of level. Let

$$\theta^{(k)} = (b_1, c_2, \dots, b_k, c_k)$$

be the places and sizes of these jumps. The likelihood for the model is

$$p(x|b, c) \propto \exp\left\{-\frac{1}{2d}\sum_{i=1}^{n}(\Delta x_i - c_i 1_{\{b_i\}})^2\right\},$$



RJMCMC Example (cont.)

Let the prior for the locations of the jumps be uniform over the observational interval and the prior for the number of jumps be Poisson. Let prior sizes of the jumps be Gamma with unknown sign and suppose that sizes and places are independent.

$$p(k) = \frac{\lambda^k}{k!} e^{-\lambda}$$

$$p(|c|) = \Gamma(c, \alpha, \beta) = \frac{x^{\alpha-1} e^{-x/\beta}}{\beta^{\alpha}\Gamma(\alpha)}$$

Reversible jump MH: At every step we have 3 alternatives

1. Stay at the current model and do a standard MH-step componentwise for every step present.
2. Add one step at random location and propose a size for the jump there.
3. Remove one step by random choice among present jumps.

RJMCMC Example (cont.)

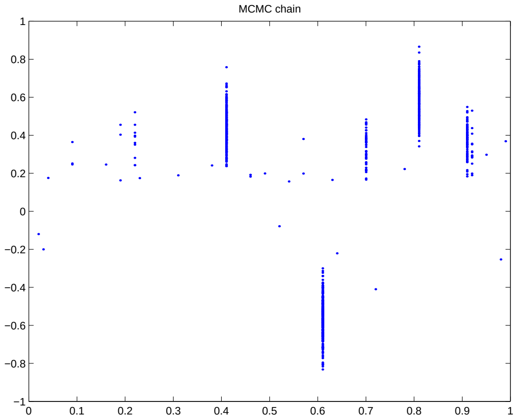
This gives us the acceptance probabilities

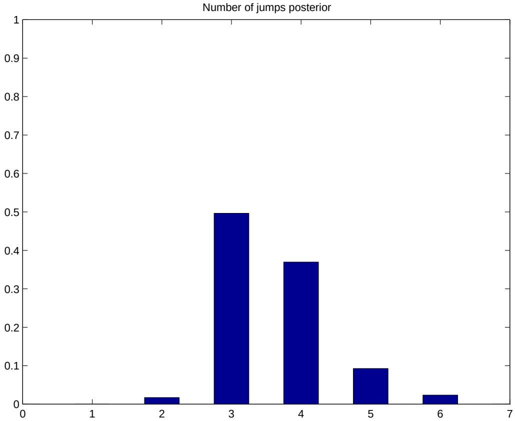
$$
\begin{aligned}
\alpha_{k,k+1}(b_{(k+1)}, c) &= \min\left\{1,\ \exp\left(\frac{\Delta x^2_{(k+1)}}{2d} - \frac{|c|}{\beta}\right) \frac{|c|^{\alpha-1}\lambda(n-k)\sqrt{2\pi d}}{2\beta^{\alpha}\Gamma(\alpha)(k+1)^2}\right\}, \\
\alpha_{k,k-1}(b_{(k)}, c) &= \min\left\{1,\ \exp\left(\frac{-\Delta x^2_{(k)}}{2d} + \frac{|c|}{\beta}\right) \frac{2k^2\beta^{\alpha}\Gamma(\alpha)}{(n-k+1)\lambda|c|^{\alpha-1}\sqrt{2\pi d}}\right\}, \\
\alpha_b(b_{(i)}, b_{(j)}) &= \min\left\{1,\ \exp\left\{\frac{c(\Delta x_{(j)} - \Delta x_{(i)})}{d}\right\}\right\},
\end{aligned}
$$

and

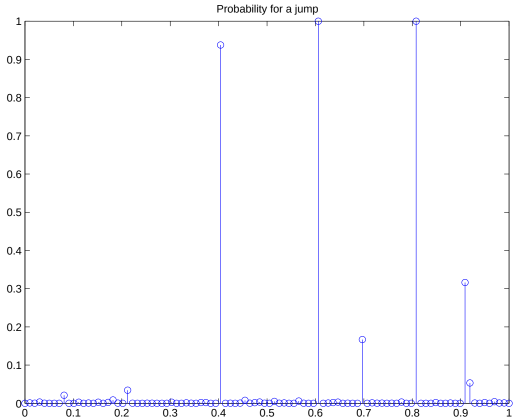
$$
\alpha_c(c, c') = \min\left\{1,\ \exp\left(\frac{-1}{2d}\left(2\Delta x_{(j)}(c' - c) - c^2 + (c')^2\right)\right)\left|\frac{c'}{c}\right|^{\alpha-1} \exp\left(\frac{|c| - |c'|}{\beta}\right)\right\}.
$$

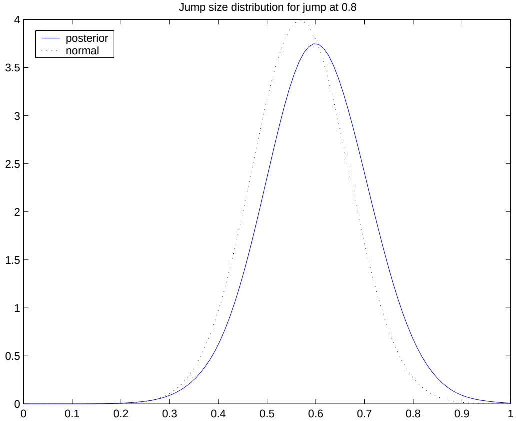
RJMCMC Example (cont.)



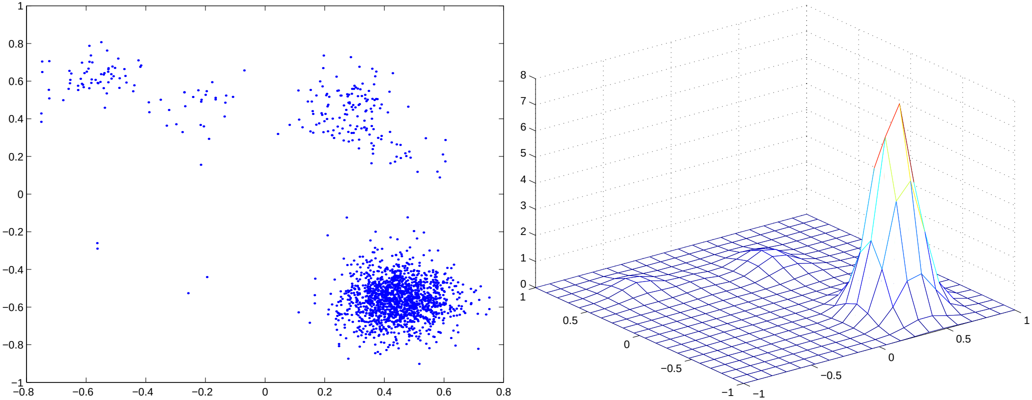


RJMCMC Example (cont.)


Probability for a jump


Jump size distribution for jump at 0.8

RJMCMC Example (cont.)



Generated chain and 2d density for the first two jumps.

Toy example: Stochastic Lorenz 95 model

- The next example demonstrates the difficulties in using non-linear models to describe dynamical systems. Chaotic behaviour, initial values, parameter estimation.
- It is a common bench mark model for studying data assimilation, e.g. state estimation with large dimensional model states combined with (relatively) limited amount of observations, such as in numerical weather prediction.
- Without going into details, some solutions to the parameter estimation problem ("tuning of the model") are presented.

Stochastic Lorenz 95 model

Nature:

$$\frac{dx_k}{dt} = -x_{k-1}\left(x_{k-2} - x_{k+1}\right) - x_k + F - \frac{hc}{b} \sum_{j=J(k-1)+1}^{Jk} y_j$$

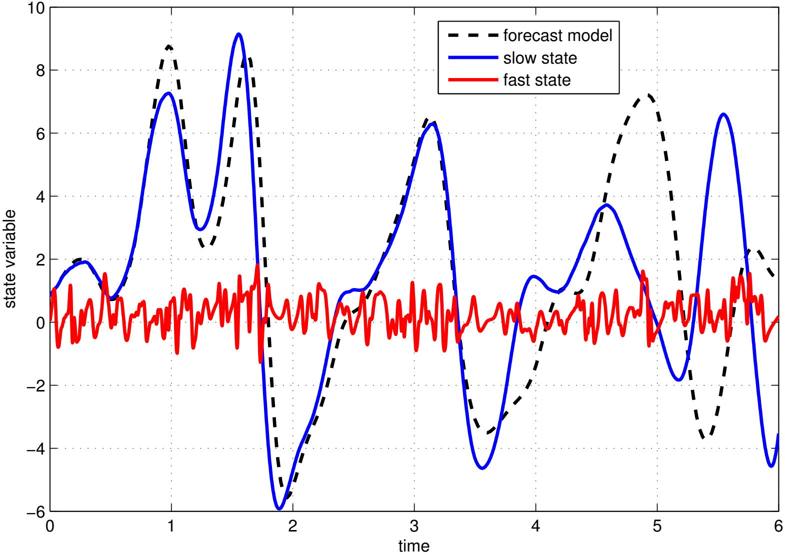
$$\frac{dy_j}{dt} = -cby_{j+1}\left(y_{j+2} - y_{j-1}\right) - cy_j + \frac{c}{b}F + \frac{hc}{b} x_{1+\lfloor \frac{j-1}{J} \rfloor}$$

Forecast model:

$$\frac{dx_k}{dt} = -x_{k-1}\left(x_{k-2} - x_{k+1}\right) - x_k + F - g(x_k, \theta).$$

with $k = 1 \dots K, j = 1 \dots JK$, $K = 40$, $J = 8$, $F = 10$, $h = 1$, $c = b = 10$ and $g(x_k, \theta) = \theta_0 + \theta_1 x_k$.

Stochastic Lorenz 95 model

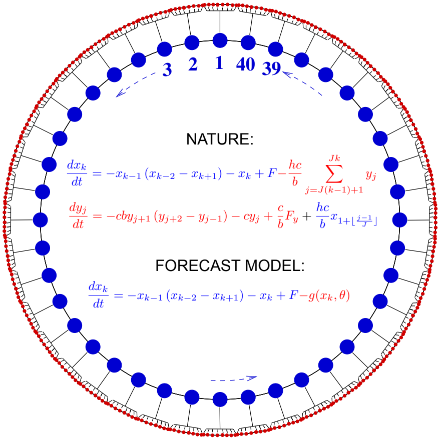


Stochastic Lorenz 95 model

We have 40 slow state variables -●- and 320 fast state variables -●-, whose effect is parametrized in the *forecast model*.

Good test case to study:

- Estimation methodologies.
- Different parameterizations $g(x_k, \theta)$.
- Modeling error.
- Filtering and ensemble methods.



NATURE:

$$\frac{dx_k}{dt} = -x_{k-1}\left(x_{k-2} - x_{k+1}\right) - x_k + F - \frac{hc}{b}\sum_{j=J(k-1)+1}^{Jk} y_j$$

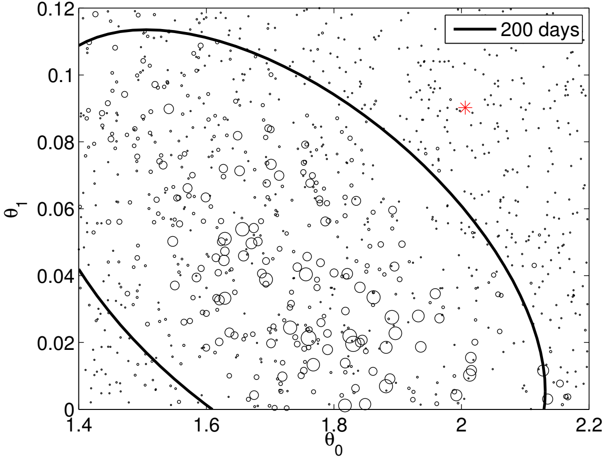
$$\frac{dy_j}{dt} = -cby_{j+1}\left(y_{j+2} - y_{j-1}\right) - cy_j + \frac{c}{b}F_y + \frac{hc}{b}x_{1+\lfloor\frac{j-1}{J}\rfloor}$$

FORECAST MODEL:

$$\frac{dx_k}{dt} = -x_{k-1}\left(x_{k-2} - x_{k+1}\right) - x_k + F - g(x_k, \theta)$$

Wilks, D.: Effects of stochastic parametrizations in the Lorenz '96 system, *Quart. J. Roy. Meteor. Soc.*, **131(606)**, 389–407, 2005. doi:10.1256/qj.04.03

Attempt 1: Lorenz 95 MCMC by summary statistics

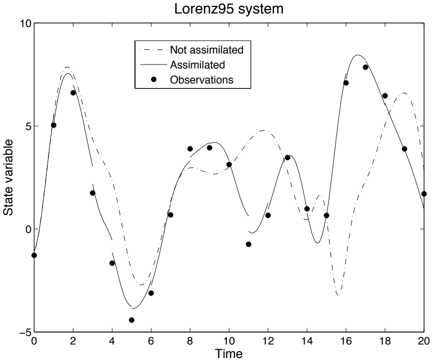


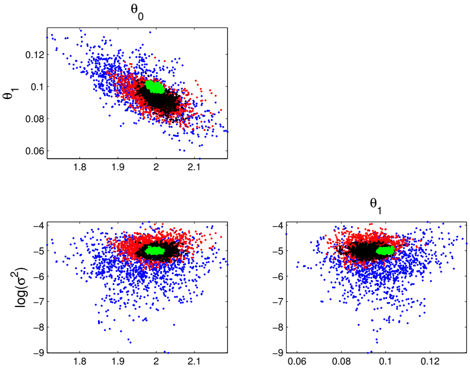
- Chaoticity: small perturbation in initial values or in parameters causes large changes in model trajectories.
- Estimation of static parameters is difficult, and even not very well defined problem.

Hakkarainen, et al, NPG 2012.

Attempt 2: Lorenz 95 with Kalman filter likelihood

- Using Kalman filter for defining the likelihood by model short term predictions while accounting for the model error and chaotic behavior.

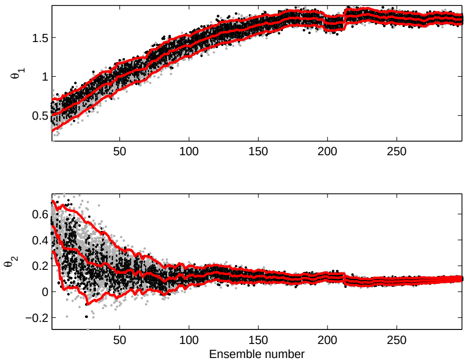




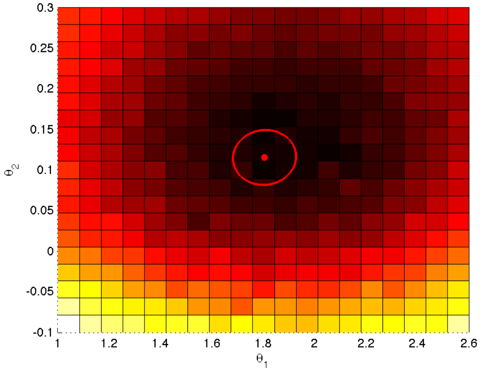
Right: scatter plot of parameter pairs from MCMC runs using 10 (blue), 20 (red), 50 (black) and 500 (green) day simulations.

Attempt 3: Lorenz 95 model with EPPES

Ensemble Kalman filter for states, parameter perturbations from proposal distribution that is adapted. On left, each column of points corresponds to proposed parameter values in one time window of the sequential estimation procedure. On right, forecast skill is calculated over a grid, with an ellipse and dot showing the final estimated parameter proposal.



parameter evolution



"6 day" forecast skill

Laine, et al, Q.J.R.Met.Soc. 2012.

Remote sensing of greenhouse gases. From ozone to methane and carbon dioxide.

Separate slides by Johanna Tamminen / FMI