Kuopio June 04

BAYESIAN INFERENCE AND MARKOV CHAIN MONTE CARLO BY EXAMPLE

GEOFF K. NICHOLLS

Abstract. In these lectures I illustrated the Bayesian inference and MCMC methods Colin Fox and Antonietta Mira had taught. Their lectures are available at `venda.uku.fi/research/FIPS/BMIP/`.

Key words and phrases. Bayesian inference, MCMC, examples, lecture notes.
I thank Jari Kaipio and Ville Kolehmainen for inviting me to give these lectures.

Inverse Problems. Inverse problems are statistical problems. In the examples below, we are given the problem of recovering the parameters θ of a differential equation from noisy measurements of the solution. Difference equations $n_{t+1} = F(n_t;\theta)$, ordinary differential $dX/dt = F(X;\theta)$ and partial differential equations $L_\theta u = 0$ all crop up in practice. The forward map from parameters θ to solution, $n_t(\theta)$, $X(t;\theta)$ and $u(x,t;\theta)$ is deterministic, and nonlinear in θ. The stochastic part of the observation process is the measurement of the solution, ($y_t \sim \Pi(n_t)$ for the blowfly population counts and $y_i \sim N(u(x_i,T;\theta),s^2)$ for the bacteria population density measurements).

We give one example in which the forward map is not determined from the solution of some physically motivated equation. In the radiocarbon dating example, the forward map $\mu(\theta)$ is simply a look-up table. This is similar to the situation when the forward map is determined from the solution of an equation, since the solution is typically evaluated by a numerical scheme which acts, from the point of view of the statistical inference for θ, as a black box. The difference between a black box and a look-up table is not great.

The physicist's forward map is often a very accurate description of the modelled reality. This is the case in electrical impedance imaging, as described in Kaipio et al. (2000). The physical forward map is very well represented by the complete electrode model, and numerical predictions of the solution are in good agreement with measurements made under controlled conditions. In this case the plan for the statistical inference is very clear (even if it is hard to carry out). More often the model misspecification error is significant. Meyer and Christensen (2000) model these errors by including some "model-noise" in the difference equation $n_{t+1} = F(n_t;\theta) + \epsilon_t$, ordinary differential $dX = F(X;\theta)dt + dB(t)$ and partial differential equations $L_\theta u = B(x,t)$, with for example ϵ_t a Gaussian Markov chain and $B(t)$ and $B(x,t)$ Gaussian fields. This allows the time (and space) evolution of the physical system to depart from deterministic evolution, and thereby accommodate the effect of unmodeled background events. The variance of the model-noise process is added to the collection θ of parameters we should estimate from the data. Judd (2003) points out that if we allow "model-noise" we can get a good fit to the data even when the deterministic part of the dynamics is qualitatively wrong. In the examples below, no example involving "model-noise" is given. It is common practice to include such noise nowadays. For a good example involving differential equations, see Andersen et al. (2003).

My examples are not uniformly "good". The only example in which something close to best practice is illustrated from beginning to end is the radiocarbon dating example. In the other examples I take the kind of shortcuts which are OK for teaching purposes, to keep things simple.

Basic Bayesian inference.

The data. The simplest path to Bayesian inference for a data set starts with data exploration. Plot the numbers and get to know them. Does the observation model seem plausible? Crude tests and common sense checking can reveal important details. In these lectures I have focused more on the technical issues, and omitted this stage. For example, in the Blowfly data analysis, I take a Poisson model for the fly-counting error $y_t|n_t$. The time evolution of the blowfly population size is quite densely sampled, so this model could be tested by fitting a smooth curve $n(t)$ (for example, a straight line) to short subsequences of the data and considering the distribution of $y_t|n(t)$, which should be close to Poisson. Try it yourself and prepare to be unimpressed.

Likelihood. Once we have a model of the forward map, $\mu(\theta)$ from parameters θ to solution $\mu(\theta)$, and a model of the stochastic part of the measurement process, from solution to data y, we have a likelihood $f(y|\mu(\theta))$ which we write below as $L(y|\theta)$. Notice that the data depends of the parameter through the value of $\mu(\theta)$ only. The setup here is very nicely set out in Geman and Geman (1984) for the case of image reconstruction. Those authors summarize the posterior using the mode of the distribution, which they estimate using Gibbs sampling, and simulated annealing. That paper is still well worth reading. As Colin pointed out, the posterior mean is often a more reliable than the mode. We focus on sampling the posterior distribution, rather than maximising it.

Prior probability. Prior knowledge is summarized in a density $p(\theta)$. This function models a state of knowledge about the unknown true value of θ. When we carry out Bayesian analysis we probe the data with various different prior beliefs about the parameter. To what extent does the data impose certain conclusions about properties of θ? We repeat the entire analysis for several choices of prior representing states of knowledge of interest. These will usually include priors which are informative, and non-informative with respect to the key scientific hypotheses. This point is illustrated in the radiocarbon dating example. It is usually a good idea to draw samples from the prior and check (by making histograms) that the samples do indeed represent the state of knowledge the prior is intended to model.

Posterior probability. The posterior density $\pi(\theta|y)$ represents the sum of the knowledge available. It is typically related to the variables of interest through

$$\pi(\theta|y) \propto L(y|\theta)p(\theta).$$

It is sometimes necessary to introduce auxiliary variables, representing missing data, in order to get simple relations for L and p. This point is illustrated in the radiocarbon dating example.

Reporting results. The problem remaining is to summarize the posterior density and report results. It is often possible to present results in a very simple form. For example, suppose a scientist has a hypothesis about the unknown true value of θ. The hypothesis can often be expressed as the statement that θ lies in some set A say. The probability that the hypothesis is true, given the data and modelling assumptions is $p_A = \mathsf{E}\{\mathbb{I}_{\theta \in A}\}$, the expectation of the indicator function $\mathbb{I}$ for the event $\theta \in A$. We evaluate p_A (with an error) and report it to the scientist.

Expectations are integrals, $\mathsf{E}\{f(\theta)\} = \int f(\theta)\pi(\theta|y)$, (and histograms are expectations) so our summarizing the posterior depends on being able to evaluate such integrals. As Antonietta explained, Monte Carlo is a convenient way to do this. Suppose that for $t = 1, 2 \ldots N$, $\theta^{(t)} \sim \pi$ (meaning, $\theta^{(t)}$ is a random sample, distributed according to the posterior distribution, which in turn means that $\theta^{(t)}$ is generated in such a way that the probability to find $\theta^{(t)}$ in some set A is p_A). Under certain conditions the $\theta^{(t)}$ satisfy a CLT. If $\hat{f}_N = N^{-1} \sum_{t=1}^{N} f(\theta^{(t)})$ and τ_f is the integrated autocorrelation time Antonientta discussed, then $\hat{f}_N$ tends in distribution to $\mathbf{N}(\mathsf{E}\{f(\theta)\}, \mathrm{var}(f)\tau_f/N)$ as N gets large. That is just what we want, $\hat{f}_N$ estimates $\mathsf{E}\{f(\theta)\}$.

Markov Chain Monte Carlo.

The algorithms. We can use Markov chain Monte Carlo to generate the $\theta^{(t)} \sim \pi(\theta|y)$. In order to specify the MCMC we specify an algorithm determining the stochastic map or update from $X_t = \theta^{(t)}$ to $X_{t+1} = \theta^{(t+1)}$, consecutive states of the Markov chain. The update is designed so that the Markov chain X_t is reversible with respect to the target distribution (which in our case is $\pi(\theta|y)$, the posterior) and irreducible. It follows that the distribution of X_t converges to $\pi(\theta|y)$ from any start, and thereby generates the samples we need. We will state the MCMC algorithm in two forms, first in the usual Metropolis Hastings form, due to Metropolis et al. (1953) and Hastings (1970), and second in the form due to Green (1995).

Metropolis Hastings. Fix a proposal density $q(\theta, \theta')$ (*ie*, q is the probability density to propose θ' if the current state of the Markov chain is θ). Choose q so that it (is as simple as possible and) generates an irreducible Markov chain on the θ-support of $\pi(\theta|y)$, and make sure that any proposal can be reversed, so that $q(\theta, \theta') > 0 \Rightarrow q(\theta', \theta) > 0$ for all pairs θ and θ' in the support of $\pi(\theta|y)$. I assume the chain is started in a state $X_0 = \theta^{(0)}$ in the support of $\pi(\theta|y)$. Let $X_t = \theta$. X_{t+1} is determined in the following way. Draw $\theta' \sim q(\theta, \theta')$. Let

$$\alpha(\theta, \theta') = 1 \wedge \frac{\pi(\theta'|y)q(\theta', \theta)}{\pi(\theta|y)q(\theta, \theta')}.$$

Set $X_{t+1} = \theta'$ with probability $\alpha(\theta, \theta')$ and otherwise set $X_{t+1} = \theta$.

Metropolis Hastings Green. In the second, reversible-jump form (which is simply related to the first in the fixed dimension setting for θ in which we will use it), introduce a new random variate u, a density $q(u)$ and a mapping $\theta' = \phi(\theta, u)$ which is invertible in the sense that $\theta = \phi(\theta', u')$ has a unique solution for u', and differentiable. Now, let $X_t = \theta$. X_{t+1} is determined in the following way. Draw $u \sim q(u)$. Set $\theta' = \phi(\theta, u)$ and let

$$\alpha(\theta, \theta') = 1 \wedge \frac{\pi(\theta'|y)q(u')}{\pi(\theta|y)q(u)} \left| \frac{\partial(\theta', u')}{\partial(\theta, u)} \right|$$

where the last factor in vertical bars denotes the Jacobian of the transformation from (θ, u) to (θ', u'). Set $X_{t+1} = \theta'$ with probability $\alpha(\theta, \theta')$ and otherwise set $X_{t+1} = \theta$.

Normalizing Constants. The MCMC method has the advantage that we do not need to evaluate the overall normalizing constant for the product $L(y|\theta)p(\theta)$ since $\pi(\theta|y)$ appears only as a ratio $\pi(\theta'|y)/\pi(\theta|y)$. This means that we can in addition drop any factors in $L(y|\theta)$ and $p(\theta)$ which do not depend on θ, and this is typically a significant simplification. On the other hand we may find that the likelihood depends on θ through an intractable normalizing constant. The prior may suffer from a related disease. When this happens MCMC may be impossible, or very difficult. An example is given in the lossy image example, in which the Ising model is used as an image prior.

Mixtures of updates. The updates above determine the transition kernel $P(x, dx')$ stationary with respect to $\pi(\theta|y)$. We usually design several proposal densities $q_i(x, x')$ (or stochastic maps $\phi_i(x, u)$) for $i = 1, 2 \ldots M$, and choose one at random (map i with probability ξ_i say) at the start of each update. We have then a collection of transition kernels $P_i(x, dx')$ and an overall kernel $P(x, dx') = \sum_{i=1}^{M} \xi_i P_i(x, dx')$. If each kernel is reversible with respect to $\pi(\theta|y)$ and $P(x, dx')$ is irreducible then X_t converges in distribution to $\pi(\theta|y)$ from any start, without the need for any single kernel $P_i(x, dx')$ to generate a Markov chain with unique equilibrium $\pi(\theta|y)$.

Choosing transformations for MCMC efficiency. In order to get rapid MCMC convergence and decorrelation in equilibrium one needs to make an appropriate choice for the proposal density $q(\theta, \theta')$. Typically we start with a simple choice, and mix in other updates as described in the preceding paragraph. Given an algorithm with M updates, we plot some statistics $f(X_t)$ of the output and (if possible) calculate integrated autocorrelation times (see Antonietta's notes, and Sokal (1989) and Geyer (1992)). In the examples below we pay particular attention to difficulties associated with singularities and near-symmetries of the equilibrium distribution. Once we have identified a statistic that we care about that mixes poorly we can add a move (move $M + 1$) designed specifically to alter

the value of the poorly behaved statistic. Thus in the radiocarbon example we add a move that mixes first the average of the date parameters and then one that mixes the difference between the oldest and most recent date parameters.

Rather than choosing a density $q(\theta, \theta')$ to define the new update it is often convenient to define a transformation $\theta' = \phi(\theta, u)$ parameterized by one or more variables u. By randomizing u we define an update, and the Green formalism gives the appropriate acceptance probability (to ensure $\pi(\theta|y)$ is stationary). Notice that we are using the reversible jump formalism, even though the dimension of θ and θ' may be equal, because it frees us to make transformations on subspaces $\theta' = \phi(\theta, u)$ parameterized by u.

Debugging. Once we have a working MCMC algorithm, we need to debug it. The first thing we usually do is "switch off the likelihood" (so that the acceptance probability becomes

$$1 \wedge \frac{p(\theta')q(\theta', \theta)}{p(\theta)q(\theta, \theta')},$$

and the equilibrium of the Markov chain is the prior). We form estimates $\hat{f}_N$ of the expectations $\mathsf{E}\{f(\theta)\}$ of statistics of interest (along with standard error estimates, $\mathrm{var}(\hat{f}_N) \simeq \mathrm{var}(f)\tau_f/N$, see above) under the prior and compare them with independently calculated results (closed form expectations are often available for the prior, or there may be some simple alternative simulation scheme for the prior). Reliable error estimates are important at this point, since we need to be able to decide whether the difference between $\hat{f}_N$ and $\mathsf{E}\{f(\theta)\}$ is significant.

One debugging test, which we should use when we have M MCMC proposal schemes, is to make several MCMC runs on the full posterior, varying the proportions of moves (*ie*, vary the ξ_i, $i = 1, 2, ..., M$) between runs. Varying the proportions in this way should not alter the equilibrium distribution of the MCMC. If one of the updates has a bug it will not preserve $\pi(\theta|y)$ and we should find that MCMC runs show inconsistent expectations as we vary move proportions.

Blowflies

A large number of blowflies (Lucilia cuprina) were kept in a glass tank. A fixed quantity of raw meat was placed in the tank every day. The flies were counted every second day. The population was observed to go through the boom and bust cycles visible in Figure 1. If you are interested see

`www.math.auckland.ac.nz/~nicholls/707/`

For $t = 1, 2 \ldots T$, let n_t denote the population size at generation t. Let r and b be real positive parameters. Consider the following model for the time evolution of the population size,

$$n_{t+1} = \frac{r n_t}{1 + b^4 n_t^4}$$

The model makes physical sense for r about one (though it could be as large as ten or even conceivably one hundred) and b of the order of $1/n_t$ (so around 0.001 or perhaps somewhat smaller).

The following data

t	1	2	3	4	5	6
# Flies y_t	1225	825	4752	774	3547	1549

has been pulled from the full data set, sub-sampling so that time is counted in something like fly generations. The counts y_t are not precise. We will suppose they are subject to Poisson statistics. Let N_t denote the unknown true population size at time t. Let Y_t denote the outcome of an observation (*ie* the result of counting the flies at time t). The probability to count y_t flies given that there were in fact n_t is

$$\Pr\{Y_t = y_t | N_t = n_t\} = \frac{n_t^{y_t}}{y_t!} e^{-n_t}$$

and is independent from one measurement to the next.

The unknown parameters of interest are the initial population size (*ie*, n_1), and b and r. We will estimate the parameters and quantify the uncertainty in their values. The problem breaks down into a few familiar steps.

(1) Express the posterior distribution for the parameters of interest in terms of model probability densities
(2) Write some code to evaluate those densities
(3) Design an MCMC program to simulate the posterior distribution
(4) Convince yourself it works

So, step one. Write down an expression for the posterior probability density for n_1, b and r given the data $y = [y_1, y_2 \ldots y_6]$ above. We will need to specify priors for n_1, b and r.

Now, n_1, b and r must be positive, so an adequate space of states is

$$\Omega = [0, \infty)^3.$$

Let $P(y|n_1, b, r)$ denote the likelihood function, *ie* the probability to observe y if the unknown true parameter values were n_1, b and r. Let $P(n_1, b, r)$ denote the prior preference function for n_1, b and r.

When n_t is small $n_{t+1}/n_t \simeq r$ so r is the number of viable offspring per adult in ideal conditions. Considerations of this kind could lead to informative priors for n_1, b and r. I will assume complete ignorance and impose $p(n_1, b, r) \propto 1$ (an "improper" prior, *ie* a prior that is not finitely normalizable and hence not a probability distribution).

The posterior density P we want is

$$P(n_1, b, r|y) \propto P(y|n_1, b, r)p(n_1, b, r) \tag{1}$$

$$\propto \prod_{i=1}^{T} P(y_i|n_i(n_1, b, r)) \times 1 \tag{2}$$

$$\propto \prod_{i=1}^{T} \frac{n_i^{y_i} \exp(-n_i)}{y_i!}. \tag{3}$$

where $T = 6$.

Step two, we will write some computer programs to evaluate $P(y|n_1, b, r)$. We will start by writing a computer program to evaluate $n_t, t = 2, 3 \ldots T$ given n_1, b and r. The program takes as input T, n_1, b and r and outputs a vector $[n_1, n_2 \ldots n_T]$.

```
function n=popln(n1,r,b,T)
% function n=popln(n1,r,b,T)

n(1)=n1;
for t=2:T
   n(t)=r*n(t-1)/(1+(b*n(t-1))^4);
end
```

For example Figure 2 was generated using

```
n=popln(500,30,0.01,30); plot(n);
```

Completing step two, here is a computer program to evaluate the log of the likelihood, $\log(\Pr\{Y = y|n_1, r, b\})$. Note `gammaln(n + 1)` is $\log(n!)$ in MatLab.

```
function [llkd,n]=LogLkd(n1,r,b,y)
%function [llkd,n]=LogLkd(n1,r,b,y)
```

```
T=length(y);
n=popln(n1,r,b,T);
llkd = sum(log(n).*(y)-n-gammaln(y+1));
```

OK, step three design and implement an MCMC algorithm which samples the posterior probability density for n_1, b and r given y. The aim here is to design a Markov chain with equilibrium $P(n_1, b, r|y)$. It is usually a good idea to at least try the simplest thing: random-walk MCMC using fixed jump sizes. This update is often inefficient *per update*, but it may be possible to evaluate the update very rapidly, so the chain mixes very rapidly *per CPU second*. Also, it is easy to implement the algorithm correctly (fingers crossed).

Fix step sizes w_n, w_b, w_r all greater than zero. Let $X_t = (n_1, b, r)$. X_{t+1} is determined in the following way. For each parameter p draw a candidate $p' \sim U(p - w_p, p + w_p)$. Now we have the candidate state (n_1', b', r'). With probability

$$\alpha = 1 \wedge \frac{P(n_1', b', r'|y)}{P(n_1, b, r|y)}$$

set $X_{t+1} = (n_1', b', r')$ else set $X_{t+1} = (n_1, b, r)$.

Here's some code.

```
%fit n_{t+1}=rn_t/(1+(b*n_t)^4) for n_1, r and b to t=[1,2...6]
%y=[1225 825 4752 774 3547 1549]. Prior n_1,r,b: Uniform[0,inf)^3
%Observation model: y_t~Poisson(n_t).

%data
y=[1225 825 4752 774 3547 1549];

%X_0=(n1,r,b)
r=1; b=0.001; n1=y(1);
llo=LogLkd(n1,r,b,y);

%MCMC jump parameters
a=[10,0.1,0.0001];

%MCMC sample size parameters
N=1000000;
M=1000;

%store every M'th sample in S
S=zeros(4,N/M+1);
S(:,1)=[n1;r;b;llo];
```

```
for k=1:N
    %generate candidate state
    p=[n1,r,b]+a.*(2.*rand(1,3)-1);
    n1p=p(1); rp=p(2); bp=p(3);

    %accept/reject (n1,b,r >0)
    if (n1p>0 & rp>0 & bp>0)
        llp=LogLkd(n1p,rp,bp,y);
        alpha=min(1,exp(llp-llo));
        if rand<alpha
            n1=n1p; b=bp; r=rp; llo=llp;
        end
    end

    %every M updates draw a sample and print current values
    if ~mod(k,M)
        [llt,n]=LogLkd(n1,r,b,y);
        disp(sprintf('%0.5g ',k,n1,r,b,llt,s/M));
        S(:,k/M+1)=[n1;r;b;llt];
    end
end
```

And step four, we should convince ourselves the MCMC is debugged and converges. The MCMC should converge to the same equilibrium distribution regardless of the start state, and thereafter behave as a stationary process. The burn-in (trending section) visible in MCMC output should be a small fraction of the total run length. The integrated autocorrelation time should be a small fraction of the total run length, so that the effective sample size is something reasonable (where "reasonable" means roughly "enough samples to represent the range of parameter values supported by the posterior distribution").

In fact we see that the simple random walk update is not up to the job (without at least some further tuning). See Figure 3. The MCMC converges to different modes (actually the outcome is very sensitive to the first few random numbers, it is started between two modes). In order to fix this I added some moves designed for this specific problem. The new code section replaces

```
    %generate candidate state
    p=[n1,r,b]+a.*(2.*rand(1,3)-1);
    n1p=p(1); rp=p(2); bp=p(3);
```

with some code we will return to later when we discuss reversible jump MCMC Green (1995),

```
rho=2;
.
.
.
if d<0.3
    p=[n1,r,b]+a.*(2.*rand(1,3)-1);
    n1p=p(1); rp=p(2); bp=p(3);
    hr=1;
elseif d<0.6
    beta=rho^(2*rand-1);
    n1p=n1*beta^3; rp=r*beta; bp=beta*b;
    hr=beta^5;
else
    beta=rho^(2*rand-1);
    n1p=n1*beta; rp=r*beta; bp=beta*b;
    hr=beta^3;
end
.
.
.
alpha=min(1,hr*exp(llp-llo));
```

[Come back to this after the discussion of scaling. Consider the third move. The draw $u \sim U(0,1)$, followed by $\beta = \rho^{2u-1}$ simulates $\beta \sim q(\beta)$ with $q(\beta) = \beta^{-1}\mathbb{I}_{1/\rho<\beta<\rho}$. This balances the scale factor β so that it is equally likely to shrink $\beta < 1$ or expand $\beta > 1$. Now the Jacobian for the transform $n_1' = n_1\beta$, $r' = r\beta$, $b' = b\beta$ and $\beta' = 1/\beta$ is β, and $q(\beta')/q(\beta) = \beta^2$ so the acceptance probability is $1 \wedge \pi(\theta'|y)/\pi(\theta|y) \times \beta^3$. I leave the second move as an exercise. I chose these moves (and $\rho = 2$) so that MCMC could move in a single scale move between the relevant sticking values.]

The graph shows fairly dramatic convergence; see Figure 4. I have computed the autocorrelation functions (ACF) and integrated autocorrelation times (IACT, τ_f). These are shown in Figure 5 (with τ_f the IACT for each statistic in turn, and M the cut off in the summation over the ACF used to compute the IACT). The effective sample sizes were $20, 30, 170$ and 120 for n_1, r, b, and $\log(P(y|n_1, r, b))$ respectively. It would probably be a good idea to repeat the analysis with a run length a factor of ten larger, if we want any real confidence. By the standards of Sokal (1989) and Geyer (1992) we have some mild problems.

For each parameter, we can now make a histogram of the marginal posterior distribution by simply making histograms of these output traces. The histograms are given in Figure 6. In fact it looks like the observation model, or the difference equation itself are poor models for the data. See Figure 7. The iteration does

not fit the data well for parameter values favored by the posterior distribution. Perhaps the parameters are functions of time? Perhaps we should allow for some "model-noise"

$$n_{t+1} = \frac{rn_t}{1 + b^4 n_t^4} + \epsilon_t \qquad \epsilon_t \sim \mathbf{N}(0, \sigma^2).$$

and estimate σ as an additional parameter. Maybe the Poisson error model for the counts is too crude. We should revise the model and go through the fitting process again.

Something else to worry about. Our interpretation of the posterior (and our MCMC) assumes the posterior is a probability distribution. When the prior is not normalisable we should check that the posterior is normalisable. For the prior $p(n_1, b, r) = 1$ above, the posterior is normalisable if $\int_\Omega P(y|n_1, b, r) dr\, db\, dn_1 < \infty$.

This unfortunately is not true. The map $n_{t+1} = f(n_t)$ has a fixed point at $n = b^{-1}(r-1)^{1/4}$ which is stable for $1 < r < 2$ and all b. It may be shown that, for all b sufficiently large, $|n_t - n| \leq |n_1 - n|$ for each $t = 1, 2, 3 \ldots$, so that, for each n_1 and $1 < r < 2$ there is $\epsilon > 0$ so that

$$\lim_{b \to \infty} P(y|n_1, b, r) > \epsilon.$$

It follows that $\int_\Omega P(y|n_1, b, r) dr\, db\, dn_1$ does not exist.

So why did the above MCMC/Bayesian inference work? The issue of normalisability is not always of practical importance (but can be). We could imagine imposing upper limits on n_1, r and b, at huge values. The posterior is now normalisable. Our MCMC output is unchanged, since it never visited such large n_1, b and r so the presence or absence of upper bounds was never tested by the MCMC process. In this problem the non-normalisability is caused by a long tail of (physically infeasible) low probability states.

I have treated n_t as a real. This is convenient. One might set `n = round(n)`. This will be important if the population size visits states close to zero.

In the next two examples I illustrate a simple technique for handling badly behaved target densities. The idea is to find a deterministic transformation of the state parameterized by a vector u and generate candidates by randomizing u. The trick is to get the right acceptance probability to ensure the equilibrium of the MCMC is the desired target density.

Scaling example I: Singularities. In this example the target density is $\pi(x) \propto 1/\sqrt{x}$ in $(0,1)$. This density has a singularity at $x = 0$ and a finite integral on $(0,1)$. Try writing a random-walk MCMC $x' \sim U(x-\delta, x+\delta)$ for this problem. Once the MCMC enters the region $0 < x \ll \delta$ it will have difficulty escaping. The problem is that $\pi(x+\delta)/\pi(x)$ is very small. If you choose δ very small that problem is to some extent resolved, but then the walk will move very slowly in the region $\delta \ll x < 1$.

We will use the reversible jump formalism to calculate the acceptance probability for a log-scale random-walk update. The idea is to scale in and out of the singularity, that is, scale the distance to the singularity. We draw $\beta \sim U(1/\rho, \rho)$ and set $x' = \beta x$. Notice that the map $\phi(x,\beta)$ is inverted by $\beta' \in (1/\rho, \rho)$. This is necessary for the MCMC to be reversible. In terms of these variables, the general formula for the acceptance probability (satisfying detailed balance) would be

$$\alpha(x,x') = 1 \wedge \frac{\pi(x')q(\beta')}{\pi(x)q(\beta)} \left| \frac{\partial(x',\beta')}{\partial(x,\beta)} \right|$$

where q is the uniform density on $(\rho, 1/\rho)$. Here

$$\alpha(x,x') = 1 \wedge \frac{1}{\sqrt{\beta}} \times \frac{1}{\beta}.$$

The scaling update gives irreducibility without the need for further updates. Here is some MatLab code to simulate the update.

```
%MCMC updates and subsample interval
N=10000;
SS=10;

%X_0, initialize MCMC
x=rand;
Xout=zeros(N/SS+1,1);
Xout(1)=x;

%scale chosen from [1/rho,rho]
rho=2;
```

```
%MCMC iteration
for k=1:N
    %RW log scale beta ~ U(1/rho,rho)
    beta=1/rho+(rho-1/rho)*rand;
    xp=beta*x;
    if xp<=1
        alpha=min(1,(1/beta)^(3/2));
        if rand<=alpha
            x=xp;
        end
    end
    if ~mod(k,SS)
        Xout(k/SS+1)=x;
    end
end
```

Scaling example II: (Near) symmetries. In this example the target density is $\pi(x,y) = \exp(-\mu xy)g(x,y)$ in $(0,1)^2$ with $g(\beta x, y/\beta)$ some function insensitive to β. We will suppose $g(x,y) = 1$ in the code example.

We will use a mixture of MCMC updates to simulate this density. We start with a random walk update to ensure irreducibility, then add other updates to improve mixing along the ridge direction of the target probability density $\pi(x,y)$. We omit the details of the random walk update.

We use the reversible jump formalism to construct the ridge update. In this example the transformation $(x',y') = (\beta x, y/\beta)$ will do as an update operator. If $\beta \sim q(\beta)$ the acceptance probability for this update is

$$\alpha((x,y),(x',y')) = 1 \wedge \frac{\pi(x',y')q(\beta')}{\pi(x,y)q(\beta)} \left| \frac{\partial(x',y',\beta')}{\partial(x,y,\beta)} \right|$$

which is

$$\alpha((x,y),(x',y')) = 1 \wedge \frac{g(x',y')q(\beta')}{g(x,y)q(\beta)} \beta^{-2}.$$

The simplest choice for $q(\beta)$ is uniform in an interval designed to ensure reversibility, namely $q(\beta) = 1/(\rho - 1/\rho)$ for $\rho > 1$ a constant ($\rho = 2$ below) and $\beta \in (1/\rho, \rho)$. This update is biased towards generating large x and small y. There is a factor β^{-2} in α to remove this bias. We might do better to use $q(\beta) = \beta^{-1}$ for $\beta \in (1/\rho, \rho)$. This balances the x', y'-proposals so that $\alpha = 1 \wedge g(x',y')/g(x',y')$. When $g(x,y) = 1$ as in our code-example we find $\alpha = 1$ exactly so the update is always accepted.

```
%MCMC for exp(-mu x_1 x_2)g(x_1,x_2) with g=1
%mu=0 gives uniform on (0,1)^2 for debugging test
mu=0;
%MCMC updates, subsample interval, and storage
N=100000; SS=10; Xout=zeros(N/SS+1,2);
%X_0, initialize MCMC
x=rand(1,2); Xout(1,:)=x;
%RW jump size
w=0.1;
%scale chosen from [1/rho,rho]
rho=2;

for k=1:N
    d=rand;
    if d<1/3
        %basic RW
        xp=x+w*(2*rand(1,2)-1);
        hr=1;
    elseif d<2/3
        %RW log scale beta ~ U(1/rho,rho)
        beta=1/rho+(rho-1/rho)*rand;
        xp=[beta,1/beta].*x;
        hr=1/beta^2;
    else
        %RW log scale beta ~ 1/beta
        beta=rho^(2*rand-1);
        xp=[beta,1/beta].*x;
        hr=1;
    end
    if all(xp>=0 & xp<=1)
        logr=log(hr)+mu*(prod(x)-prod(xp));
        if log(rand) <= logr
            x=xp;
        end
    end
    if ~mod(k,SS), Xout(k/SS+1,:)=x; end
end
```

The code above is written in the way I write code for more complex examples, so I have ignored some simplifications in the interest of regular code structure. Also, it is often best to work with the log of the acceptance ratio for numerical stability. I tested it quickly by running it for $\mu = 0$ and checking (by eye) that the scatter of points (x, y) was uniform in $(0, 1)^2$. See Figure 8.

Radiocarbon dating example

The data comprises $K = 8$ conventional radiocarbon age determinations $y = (y_1, \ldots, y_K)$ and standard errors $\tilde{\sigma} = (\tilde{\sigma}_1, \ldots, \tilde{\sigma}_K)$ taken from Anderson et al. (1996) (all dated charcoal from the SM/C:Dune terrestrial series). In the format $(y_n, \tilde{\sigma}_n)$, the data are $(580, 47)$, $(630, 82)$, $(600, 50)$, $(537, 44)$, $(600, 50)$,$(670, 47)$, $(624, 58)$, $(560, 45)$.Let θ_n be a trial value for the unknown true date (measured in calendar years AD) of the n'th dated specimen. The observation model for y_n is, independently for each n,

$$y_n|\theta_n \sim N(\mu(\theta_n), \hat{\sigma}(\theta_n)^2 + \tilde{\sigma}_n^2), \tag{4}$$

where μ and $\hat{\sigma}^2$ are standard, empirically determined radiocarbon calibration functions, and we are treating the $\tilde{\sigma}_n$ as covariates. These functions, published in Stuiver et al. (1998), and illustrated in Figure 9, are available from

$$\texttt{http://depts.washington.edu/qil/}$$

in decadal tabulation, which we spline to arrive at functions μ and $\hat{\sigma}^2$ piecewise constant by year. Let $\ell(y_n|\theta_n)$ denote the normal density function as given in Eq. (4).

The questions of archaeological interest are typically "when was the site occupied, when was it abandoned and for how long was it occupied in all"? The age difference between the oldest and youngest date is a crude lower bound on the occupation span. We will introduce two parameters, χ_1 and χ_2 to represent the dates for occupation and abandonment of the site and try to estimate their values. In a sense the θ_i are nuisance parameters, sometimes called auxiliary parameters, which we introduce in order to get a simple probability distribution for the parameter of real interest, namely χ_2 and χ_1.

In the prior model the date parameters $\theta_1, \ldots, \theta_K$ are independently and uniformly distributed between two parameters $\chi = (\chi_1, \chi_2)$ which are themselves known to lie in a (time) interval $[A, B] \subset \mathbb{R}$ of length $R = B - A$, with $\chi_2 > \chi_1$, but are otherwise unknown. Letting $\theta = (\theta_1, \ldots, \theta_K)$ and $x = (\chi_1, \chi_2, \theta)$, the state space of the target distribution is

$$\Omega = \{x : A \le \chi_1 < \chi_2 \le B,\, \theta \in [\chi_1, \chi_2]^K\}.$$

See Litton and Buck (1996) for further background. The unnormalized prior densities of interest are

$$\pi_1(x) = 1$$

and

$$\pi_2(x) = \frac{1}{R - (\chi_2 - \chi_1)} \times \frac{1}{(\chi_2 - \chi_1)^K}.$$

The density π_1 has the undesirable feature that the marginal prior density for the span statistic $\chi_2 - \chi_1$ is strongly weighted towards large span. If we reparameterize

using $\chi^{\pm} = \chi_2 \pm \chi_1$ we find

$$\int_{2A+\chi^-}^{2B-\chi^-} \int_{[\chi_1,\chi_2]^K} \pi_1 \, d^K\theta d\chi^+ \propto (R-\chi^-)(\chi^-)^K.$$

The span is a key statistic in real dating applications, since it is an estimate of the length of time the site was occupied. If two scientists are arguing over the span, one favoring a span of one hundred years and the other two hundred, and they analyze the data using π_1, the analysis is biased towards the wider span by a factor 2^K, that is 2^8 here. The density π_2 has uniform marginal span on $[0, R]$. It is actually the density for a simple physical model of the deposition process, namely a model in which the K dated objects were deposited in the ground by a Poisson process acting at a constant rate between times χ_1 and χ_2.

Under a generic prior $\pi(x)$, the unnormalized posterior density $\pi(x|y)$ of x is

$$\pi(x|y) = \pi(x) \times \prod_{m=1}^{K} \ell(y_m|\theta_m)$$

with $dx = d\chi_1 d\chi_2 d\theta_1, \ldots, d\theta_K$, the restriction of Lebesgue measure to Ω.

MCMC. We use several updates to make out MCMC for x. We give the updates needed to simulate $\pi(x|y)$ when the prior is π_2. Suppose the current state is $X_t = x$. We have:

- **(1):** a Metropolis adjusted random walk update acting on a randomly chosen parameter; this gives ergodicity in principle, but is insufficient in practice;
- **(2):** a Metropolis adjusted random walk update of fixed scale Δ applied to all dates: a number $\delta \sim U(-\Delta, \Delta)$ is generated; the proposal is $x_i' = x_i + \delta$ for each $i = 1, 2, \ldots, K+2$;
- **(3):** a Metropolis-Hastings adjusted centered random scaling: we fix $\rho > 1$ and draw a random number $\beta \sim U(1/\rho, \rho)$; the proposal is

$$x_i' = \beta(x_i - \bar{x}) + \bar{x} \qquad \bar{x} = \frac{1}{K+2} \sum_{j=1}^{K+2} x_j$$

for each $i = 1, 2, \ldots, K+2$, and Hastings' ratio is

$$\begin{aligned} \alpha(x, x') &= 1 \wedge \frac{\pi(x'|y) q(\beta')}{\pi(x|y) q(\beta)} \left| \frac{\partial(x', \beta')}{\partial(x, \beta)} \right| \\ &= 1 \wedge \frac{1}{\beta} \times \frac{R - (\chi_2 - \chi_1)}{R - (\chi_2' - \chi_1')} \times \prod_{m=1}^{K} \left(\frac{\ell(y_m|\theta_m')}{\ell(y_m|\theta_m)} \right). \end{aligned}$$

At each update in the MCMC, with probability 1/3 each, one of these three moves is chosen. Move (3) is particularly important for analysis of the prior π_2, which has a singularity at $\chi^- = 0$.

In move (3), the Jacobian for the change of variables from (x, β) to (x', β') (with $\beta' = 1/\beta$ the β-value for the reverse update) is β^{K-1}. The $x \to x'$ transformation scales $K+1$ directions in Ω by β and leaves one (corresponding to the invariant $\bar{x} = \bar{x}'$) unscaled, so the $\partial x'/\partial x$ block has eigenvalues 1 (eigenvector $\underline{1}$ the $K+2\times 1$ vector of ones) and β (repeated $K+1$ times, eigenvectors $\underline{1} - (K+2)\underline{e_i}$). The $\partial\beta'/\partial\beta$ block contributes a factor $1/\beta^2$ giving β^{K-1} for the determinant. The ratio $\pi_2(x')/\pi_2(x)$ which appears in $\pi(x'|y)/\pi(x|y))$ contributes a factor $1/\beta^K$ giving $1/\beta$ in all.

The scaling move (3) is centered on the singularity, in the same way that the scaling move $x' = \beta x$ in example I, for $1/\sqrt{x}$ is centered at $x = 0$. Both moves generate the new state by scaling the distance from the state to the singularity by β.

Results. How well do the three updates work for $\pi(x|y)$ with prior π_2, the 'sticky', unbounded prior? The graph in Figure 10 shows the MCMC output for three choices of the MCMC updates: using move (1) alone, the single variable random-walk update; using both moves (1) and (2), so adding the update that jointly translates all variables; and using moves (1), (2) and (3), so adding the random scaling. We see that the random scaling is really needed to get the MCMC to draw representative samples from $\pi(x|y)$ in any reasonable number of updates.

How important is the choice of prior? It would be natural to compare the posterior distribution of the span $\chi_2 - \chi_1$ computed under the two priors, π_1 and π_2. It is clear we will get quite different answers, because χ_2 and χ_1 themselves are undated, so the only information we have about them comes from the prior and the dated variables θ, which bound χ-variables, $B > \chi_2 > \max(\theta)$ and $\min(\theta) > \chi_1 > A$. Under π_1 the χ-variables will have a flat distribution between the dated variables and the bounds, A and B. The graph in Figure 11 shows the posterior distribution of the statistic $\max(\theta) - \min(\theta)$ under the two priors. We look at $\max(\theta) - \min(\theta)$ because it gives a lower bound on the span $\chi_2 - \chi_1$ but is less sensitive to the choice of prior than the span itself. The two distributions are quite different. Under π_1 we would assert that spans shorter than 60 years can be ruled out. However the posterior distribution from π_2 taken with the data allows spans all the way down to zero. The human activity dated by the radiocarbon dates could have been no more than an overnight campsite.

Bacteria (synthetic data)

A population of bacteria reproduce and diffuse through their nutrient medium, which covers the interval $0 \leq x \leq L$. The population density $u(x,t)$ evolves according to a Fisher model,

$$\frac{\partial u}{\partial t} = ru(K-u) + D\frac{\partial^2 u}{\partial x^2}$$

for r K and D real valued parameters of the growth and diffusion process. The region outside $[0,L]$ is lethal, so that the boundary conditions are

$$u(0,t) = 0, \quad u(L,t) = 0.$$

The initial density of bacteria is known fairly precisely,

$$u(x,0) = \begin{cases} 1 & 0.75L < x < 0.8L \\ 0 & x \text{ otherwise.} \end{cases}$$

After a time T the growth and diffusion are stopped and the population density is measured at M points $x_i, i = 1, 2...M$, along the interval. The measured density values do not of course exactly coincide with the true density values at the measured points but have iid Gaussian noise with std s. The observation model is

$$y(x_i) = u(x_i, T) + \epsilon_i, \qquad \epsilon_i \sim N(0, s^2).$$

The setup is represented in Figure 12 and Figure 13.

To start with we must explain how the data $\{y_i, x_i\}, i = 1, 2 \ldots M$ could be used to estimate the parameters r, K and D. What practical biological considerations might inform our choice of prior for r, K and D? We need to write down prior densities for these parameters, and the joint posterior density for r, K and D.

The posterior density is

$$\pi(r, K, D|y) \propto L(y|r, K, D)p(r, K, D)$$

In order to keep things simple I have assumed additive Gaussian noise. The likelihood is

$$\begin{aligned} L(y|r,K,D) &= \prod_{i=1}^{M} L(y_i|r,K,D) \\ &= \prod_{i=1}^{M} L(y_i|u(x_i,T;r,K,D)) \\ &\propto \exp(-\sum_{i=1}^{M}(u(x_i,T;r,K,D) - y_i)^2/2s^2) \end{aligned}$$

In order to keep things simple I will take a uniform prior on r K and D in $[0,\infty)^3$. If I was really interested in r K and D I would use a more informative prior. I

dont recommend such flat priors in general (look at the archeology example). But using a diffuse (improper) prior can be handy, at least in the initial part of your study of the data. First of all, it is often harder to get MCMC to converge on diffuse priors than more concentrated priors, so if you get your MCMC working on the flat prior then when you substitute your more informative prior later on your MCMC should work well. Second the state of knowledge represented by a flat prior may be of independent interest - it is one of the priors you may want to study when you look at the sensitivity of the data to different choices of prior. However, whenever you use an improper (unormalisable) prior you must check the posterior is proper...

OK, lets fix some "unknown true" r, K and D and synthesise some data. We can then design and implement an MCMC algorithm to sample the posterior density for r, K and D given the synthetic data. Just for the sake of argument I used $L = 10$, $D = 1.5$, $r = 1$, $K = 2$, $s = 0.03$, $T = 2$ and a numerical analysis setup with 25 x-nodes at 75 time steps. I am just creating an example here, so I chose the numbers so things all work out fairly well. I gathered 23 density measurements at x-nodes 2 to 24 (nodes 1 and 25 have exactly zero population density by the boundary conditions).

I found that the first thing I tried, a simple random walk algorithm updating all the parameters at once, worked fine. After a bit of tuning to find a sensible value for the random-walk jump sizes, the algorithm converges fairly rapidly. Some results are shown in Figure 14. The code is given below.

```
%Given noisy measurements uDat of Z, the solution of the Fisher equation
%Z_t=rZ(K-Z)+DZ_xx with Dirichelet b.conds, taken  at a time t=Tmax and
%at x values x(xDat) between x=0 and x=L, estimate the growth rate r,
%carrying capacity K and diffusivity D of the 1-dim medium.

clear all;

global tdim q1 q2 v;

%Biological parameters
L=10;
D=1.5;
r=1;
K=2;

%data paprameters
s=0.03;
Tmax=2;
```

```
%numerical analysis setup
xdim=25;
tdim=75;
x = linspace(0,L,xdim);
t = linspace(0,Tmax,tdim);
dt=t(2)-t(1);
dx=x(2)-x(1);
q1=dt;
q2=dt/dx^2;
Z=zeros(tdim,xdim);
v=2:(xdim-1);
r1=0.75;
r2=0.8;
Z(1,round(r1*xdim):round(r2*xdim))=1;


%data observation locations
xDat=2:(xdim-1);
TDat=tdim;
nxDat=length(xDat);
nTDat=length(TDat);

%synthesise data
Z=MyFESolver(K,D,r,Z);
u = Z(TDat,xDat);
uDat = u + s*randn(nTDat,nxDat);


DatLLkd=sum(sum( -(u-uDat).^2 ))/(2*s^2);

XD=1; Xr=1; XK=1;
Z=MyFESolver(XK,XD,Xr,Z);
u = Z(TDat,xDat);
oLLkd=sum(sum( -(u-uDat).^2 ))/(2*s^2);

N=10000; m=100; DRAW=1;
X=zeros(3,N); X(:,1)=[XD;Xr;XK];
LL=zeros(1,N); LL(1)=oLLkd;

%random walk step size
w=0.1;
```

```
for n=2:N
    XDp=XD+w*(2*rand-1); Xrp=Xr+w*(2*rand-1); XKp=XK+w*(2*rand-1);
    if ( (XDp>0) & (Xrp>0) & (XKp>0) )
       Z=MyFESolver(XKp,XDp,Xrp,Z);
       u = Z(TDat,xDat);
       nLLkd=sum(sum( -(u-uDat).^2 ))/(2*s^2);
       AlphaRatio=exp(nLLkd-oLLkd);
       if ( (AlphaRatio>=1) | (rand<AlphaRatio) )
           XD=XDp; Xr=Xrp; XK=XKp;
           oLLkd=nLLkd;
           CZ=Z;
       end
   end
   X(:,n)=[XD;Xr;XK];
   LL(n)=oLLkd;
end

function Z=MyFESolver(K,D,r,Z0)
%Forward difference approximates the solution of the Fisher equation

global tdim q1 q2 v;

Z=Z0;
for tin=2:tdim
    tip=tin-1;
    Z(tin,v)=Z(tip,v)+r.*q1.*Z(tip,v).*(K-Z(tip,v))...
             + D.*q2.*(Z(tip,v+1)-2.*Z(tip,v)+Z(tip,v-1));
end
```

In fact (and we saw this before for the Blowfly data) the uniform prior on $r \geq 0, K \geq 0, D \geq 0$ leads to a posterior which is not finitely normalisable (improper). It is fairly clear that $u(x, T; r, K, D)$ will tend to a finite value when we take the limit $D \to \infty$. The inference is therefore meaningless for the uniform prior. Nevertheless the MCMC seems to settle down to a sensible distribution, recovering known r, K and D from synthetic data. The posterior distribution has a tail of infinite total probability mass but miniscule probability density at each state. The MCMC must eventually wander off to infinite D, but we dont see that as our run wasnt long enough. Again, physical considerations could be used to eliminate these states. We feel we know enough about the shape of the posterior density to know what we would see if we waited long enough.

The Binary Markov Random Field or Ising Model

Consider the space $\Omega = \{(x_1, x_2, ..., x_{N^2}) : x_m \in \{0, 1\}\}$. Suppose $x_m = 0$ marks a black pixel and $x_m = 1$ a white one so that Ω is the space of all black and white images (Figure 15) with N^2 pixels. We will write down a probability distribution for images which imposes a penalty on the number of pairs of pixels which are next-door-neighbors on the image lattice. Denote by $\#x$ the number of such 'disagreeing' pixel pairs in image x. Suppose that the probability of $x \in \Omega$ is given by

$$p(x) = \frac{1}{\mathcal{Z}_\theta} \exp(-2\theta \#x)$$

for some $\theta > 0$. The constant $\mathcal{Z}_\theta$ (which is still a function of θ) normalizes the probability distribution. The probability distribution above favours images in which pixels clump together in groups of equal color.

We wish to generate samples $X \sim p$ from this distribution. Each sample X is an array of N^2 binary values $x = (x_1, x_2, ..., x_{N^2})$. The normalising constant $\mathcal{Z}_\theta$ is not in general available (though see L. Onsager, Phys Rev, vol65, pp117, 1944) and classical sampling methods will not work. The distribution $p(x)$ is called a binary Markov random field, or "Ising" model. We will construct a reversible Markov chain on Ω with equilibrium distribution $p(x)$. We use the Metropolis Hastings prescription:

Let $X_t = x$. X_{t+1} is determined in the following way.

(1) Generate a candidate state x' from x using some process we can choose. Here is a simple method: given $x = (x_1, x_2, ..., x_{N^2})$, pick a pixel n at random from $1, 2, ..., N^2$. Set $x' = (x_1, x_2, ..., 1 - x_n, ..., x_{N^2})$. Notice that
   (a) we can get to any state in Ω by a sequence of such updates,
   (b) if x' and x differ by more than one pixel, $q(x, x') = q(x', x) = 0$,
   (c) if x' and x differ by exactly one pixel, $q(x, x') = q(x', x) = 1/N^2$,

   so our generation scheme is suitable for MH MCMC.

(2) Work out the Metropolis-Hastings acceptance probability

$$\begin{aligned} \alpha(x, x') &= \min\left\{1, \frac{p(x')q(x', x)}{p(x)q(x, x')}\right\} \\ &= \min\{1, \exp(-2\theta(\#x' - \#x))\} \end{aligned}$$

   Notice that both the q's and the normalization constants $\mathcal{Z}_\theta$ cancel. Since x and x' are the same except at x_n we write $\#x' - \#x = \#x'_n - \#x_n$ where $\#x_n$ is the number of disagreeing neighbors around pixel n. Set $X_{t+1} = x'$ with probability

$$\alpha(x, x') = \min\{1, \exp(-2\theta(\#x'_n - \#x_n))\}.$$

   Otherwise set $X_{t+1} = x$, *ie* no change.

Notice that

(1) If $\#x'_n \leq \#x_n$ i.e., the change from x to x' gives a smoother image with fewer disagreeing neighbors, then $\alpha = 1$ and the proposed change is accepted with probability 1.
(2) If $\#x'_n > \#x_n$, the change leads to a more irregular image with fewer agreeing neighbors. Then $\alpha = \exp(-2\theta(\#x'_n - \#x_n))$ and the proposed change is accepted with probability < 1.

Algorithm for generating a realization of the Markov chain
A Matlab script generating realizations from a random binary Markov field is

```
%Sample Ising model
%Lattice
M = 128; N = 128;
nbrs=GetNbrs(M,N);

%smoothing parameter
theta = 0.42;

%initialise MCMC X_0=x to agree with g where data is present
x = round(rand(M,N));
figure(3);hf=imagesc(x);colormap(gray);axis square;drawnow;
s=0;
SS=10000;

while 1,
   pixel = ceil(rand*M*N);
   hashfn=(x(nbrs{pixel})~=x(pixel));
   disagree = sum(hashfn);
   agree = sum(~hashfn);
   if log(rand) < 2*theta*(disagree-agree)
      x(pixel) = 1-x(pixel);
   end
   s = s + 1;
   if rem(s,SS)==0,
      set(hf,'CData',255*x); drawnow;
   end
end
```

```
function nbrs=GetNbrs(M,N)

if nargin==1, N=M; end
nbrs=cell(1,M*N);
for k=1:M*N
   [i,j]=ind2sub([M,N],k);
   subnbr=repmat([i;j],1,4)+[0 0 -1 1;1 -1 0 0];
   II=find(subnbr(1,:)>M | subnbr(1,:)<1);
   JJ=find(subnbr(2,:)>N | subnbr(2,:)<1);
   subnbr(:,union(II,JJ))=[];
   nbrs{k}=sub2ind([M,N],subnbr(1,:),subnbr(2,:));
end
```

The only non-straightforward part is handling pixels on the boundaries which have fewer than four neighbors. Increasing the value θ leads to images dominated by one color, as the average number of disagreeing edges decreases with increasing θ. Some examples are given in Figure 16.

Reconstruction of lost image data (synthetic data)

Let X denote a binary $N \times N$ image. Suppose X is observed, but a fraction p_{lost} of the pixels are lost. Let y denote the observed image (*ie*, the data). For $i = 1, \ldots N^2$ we have $y_i = X_i$ with probability p_{lost} and otherwise $y_i = \emptyset$. Let $B = \{i : y_i \neq \emptyset, i = 1, \ldots, N^2\}$ denote the set of observed pixels. Consider the problem of reconstructing X under an Ising image model prior with given smoothing parameter θ. Let x be a trial image, and consider the probability that $X = x$. The state space for x is

$$\Omega = \{x : \ x_i \in \{0, 1\}, i = 1, \ldots, N^2; \ x_i = y_i, i \in B\}$$

Let $\pi(x|y) = \Pr\{X = x|y\}$ so that for likelihood $L(y|x)$ and prior $p(x)$

$$\pi(x|y) \propto L(y|x)p(x).$$

Now $p(x)$ is given above, it is the Ising probability distribution. The likelihood for $x \in \Omega$ is

$$L(y|x) = p_{\text{lost}}^{N^2-|B|}(1 - p_{\text{lost}})^{|B|}$$

To simulate this posterior, we should condition the Ising model prior on $x_i = y_i$ at $i \in B$. This is just the Ising model on the reduced state space Ω, the space of binary images satisfying the constraint. All we need do to the above Ising-model algorithm is reject proposals that try to change x_i at $i \in B$ to values other than y_i. Here is the code.

```
%MCMC Bayesian image restoration, Binary NxM image X_i = 0,1
%Data y, y_i=X_i wp (1-p) else y_i=-1 (lost pixel), Ising model prior

%Read in the true image
X=imread('blob.bmp');
sz=size(X); M = sz(1); N = sz(2);
nbrs=GetNbrs(M,N);

%synthesise noisy image, fraction p data lost
p=0.9;
lost=find(rand(M,N)<p);
y=double(X); y(lost)=-1;

%prior smoothing parameter
theta = 1;

%initialise MCMC X_0=x to ayree with y where data is present
x = y;
x(y==-1)=round(rand(1,length(x(y==-1))));
figure(3);hf=imagesc(x);colormap(gray);axis square;drawnow;
SS=10000;
iter = 0;

while 1,
   pixel = ceil(rand*M*N);
   if y(pixel)==-1
      hashfn=(x(nbrs{pixel})~=x(pixel));
      disagree = sum(hashfn);
      agree = sum(~hashfn);
      if log(rand) < 2*theta*(disagree - agree)
         x(pixel) = 1-x(pixel);
      end
   end
   iter = iter + 1;
   if rem(iter,SS)==0,
      set(hf,'CData',255*x); drawnow;
   end
end
```

The initial part of the code synthesizes data. Sample results are shown in Figure 17 and Figure 18.

Notice that estimation of the smoothing parameter θ itself along with the missing image data $x_i, i \notin B$ is a very difficult problem. The posterior is

$$\pi(x, \theta|y) \propto L(y|x)p_1(x|\theta)p_2(\theta).$$

Now $L(y|x)$ is unchanged, and $p_2(\theta)$ is a simple function of a scalar argument (representing our prior belief about the value of the smoothing parameter). But $p_1(x|\theta) = \exp(-2\theta\#x)/\mathcal{Z}_\theta$ and the θ-dependence of $\mathcal{Z}_\theta$ is now fatal, since the ratio $\pi(x', \theta'|y)/\pi(x, \theta|y)$ depends on the intractable ratio $\mathcal{Z}_\theta/\mathcal{Z}_{\theta'}$. The problem of estimating the parameters of a Markov random field from realizations of the MRF is a hard problem, which MCMC by itself does not directly treat. Some progress has been made. The bridge sampling methods of Meng and Wong (1996) can treat some problems, though they are very computationally intensive for even moderate lattice sizes.

Appendix: Warmup

Mean and Variance of a Gaussian. K samples $y = (y_1 \dots y_K)$ are drawn from a normal distribution $N(\mu, \sigma^2)$. The mean μ and variance σ are unknown. Write down priors for μ and σ representing a state of knowledge something like ignorance. Write down the posterior probability distribution for μ. Generate synthetic data $y = (y_1 \dots y_K)$ given that the unknown true parameter values were $\mu = 1$ and $\sigma = 2$. Design and implement an MCMC sampler for the posterior distribution of the mean, μ and standard deviation σ.

The prior density $p(\mu, \sigma) = 1/\sigma$ defined for $\mu \in \Re$ and $\sigma > 0$ is an improper (*ie* unnormalizable) density representing ignorance of the value of μ and the scale of σ (so $p(\mu, \sigma)d\mu d\sigma$ is invariant under the transformation $\mu \to \mu + a$ and $\sigma \to a\sigma$ for $a > 0$ any real positive constant). The posterior probability density $h(\mu, \sigma|y)$ is then

$$h(\mu, \sigma|y) \propto \sigma^{-K-1} \exp\left(-\sum_{i=1}^{K} \frac{(y_i - \mu)^2)}{2\sigma^2}\right).$$

Because the priors are improper, we should really check that the posterior is finitely integrable over $\mu \in \Re$ and $\sigma > 0$.

MCMC program in MatLab for synthetic data.

```
%a) Synthesise K iid data y=[y(1),...y(K)]
K=30;

%Unknown true parameter values
mu=1; sigma=2;
% Data
y=mu+sigma.*randn(1,K);
```

```
%b) Now MCMC N states
N=10000;
%define start state
m=0; s=1;
%store the realization of the Markov chain in X
X=zeros(2,N);
X(:,1)=[m;s];

%random walk update, x'~U(x-w,x+w)
w=1;

%MCMC loop
for n=2:N
  sp = s + w*(2*rand-1);
  if sp>0
     mp = m + w*(2*rand-1);
     ratio=(s/sp)^(K+1)*exp(-sum((mp-y).^2)/(2*sp^2)+sum((m-y).^2)/(2*s^2));
     if rand < ratio
        m=mp; s=sp;
     end
  end
  X(:,n)=[m;s];
end
```

References

Andersen, K., Brooks, S., and Højbjerre, M. 2003. Bayesian model discrimination for inverse problems. Technical Report R-2003-06 Department of Mathematical Sciences, Aalborg University.

Anderson, A. J., Smith, I. W. G., and Higham, T. F. G. 1996. Radiocarbon chronology. In Anderson, A. J., Allingham, B., and Smith, I. W. G., editors, *Shag River Mouth: the archaeology of an early Southern Maori village* volume 27 pages 60–69. ANH Publications, ANU Canberra.

Geman, S. and Geman, D. 1984. Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 6, 721–741.

Geyer, C. J. 1992. Practical Markov chain Monte Carlo (with discussion). *Statist. Sci.* 7, 473–511.

Green, P. J. 1995. Reversible jump Markov chain Monte Carlo computation and Bayesian model determination. *Biometrika* 82, 711–732.

Hastings, W. 1970. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika* 57, 97–109.

Judd, K. 2003. Chaotic-time-series reconstruction by the bayesian paradigm: Right results by wrong methods. *PHYSICAL REVIEW E* 67, 026212 1–6.

Kaipio, J., Kolehmainen, V., Somersalo, E., and Vauhkonen, M. 2000. Statistical inversion methods in electrical impedence tomography. *Inverse Problems* http://venda.uku.fi/∼kaipio/.

Litton, C. D. and Buck, C. E. 1996. An archaeological example: radiocarbon dating. In W. Gilks, S. R. and Spiegelhalter, D., editors, *Markov Chain Monte Carlo in Practice* pages 465–480. Chapman and Hall London.

Meng, X.-L. and Wong, W. 1996. Simulating ratios of normalising constants via a simple identity: a theoretical exploration. *Statistica Sinica* 6, 831–860.

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., and Teller, E. 1953. Equation of state calculations by fast computing machines. *Journal of Chemical Physics* 21, 1087–1092.

Meyer, R. and Christensen, N. 2000. *Phys. Rev. E* 62, 3535.

Sokal, A. 1989. Monte Carlo methods in Statistical Mechanics. In *Cours de Troisième Cycle de la Physique en Suisse Romande, Lausanne.* Updated in 1996 for the Cargèse Summer School on "Functional Integration: Basics and Applications.", `http://dimacs.rutgers.edu/∼dbwilson/exact.html/cargese.ps.gz`.

Stuiver, M., Reimer, P. J., Bard, E., Beck, J. W., Burr, G. S., Hughen, K. A., Kromer, B., McCormac, F. G., d. Plicht, J., and Spurk, M. 1998. Intcal98 Radiocarbon Age Calibration, 24,000-0 cal BP. *Radiocarbon* 40, 1041–1083.

WWW.MATH.AUCKLAND.AC.NZ/∼NICHOLLS

E-mail address: `nicholls@math.auckland.ac.nz`

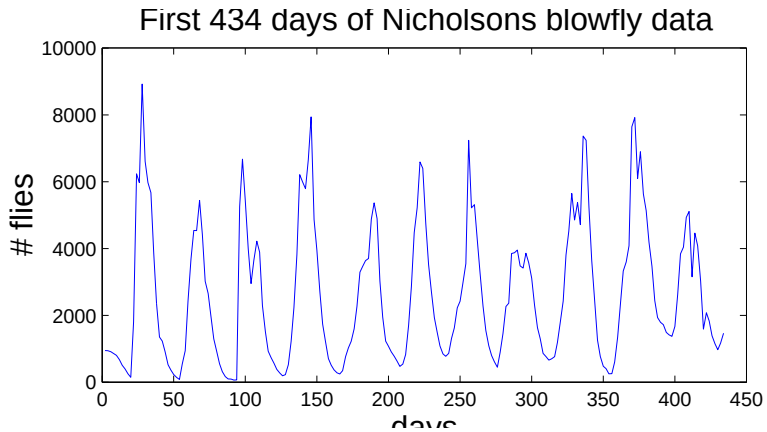

FIGURE 1. Fluctuating size of the population of flies in Nicholson's experiment. The x-axis gives time in days.

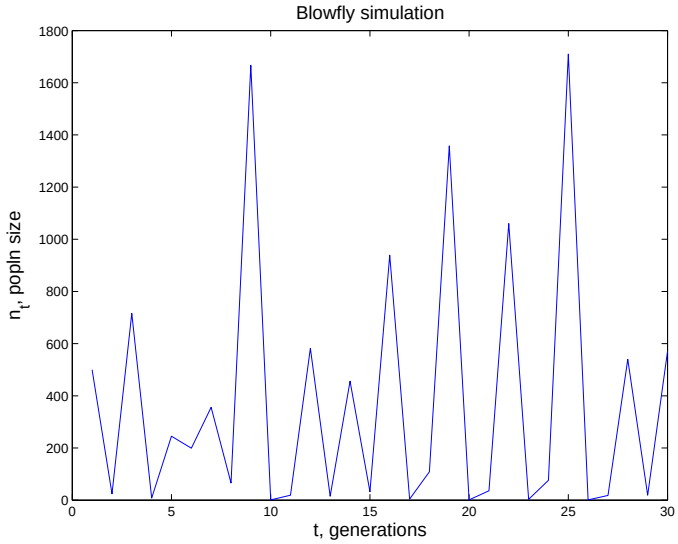

FIGURE 2. Simulation of $n_{t+1} = rn_t/(1 + b^4 n_t^4)$, $r = 30, b = 0.01$ and $n_1 = 500$.

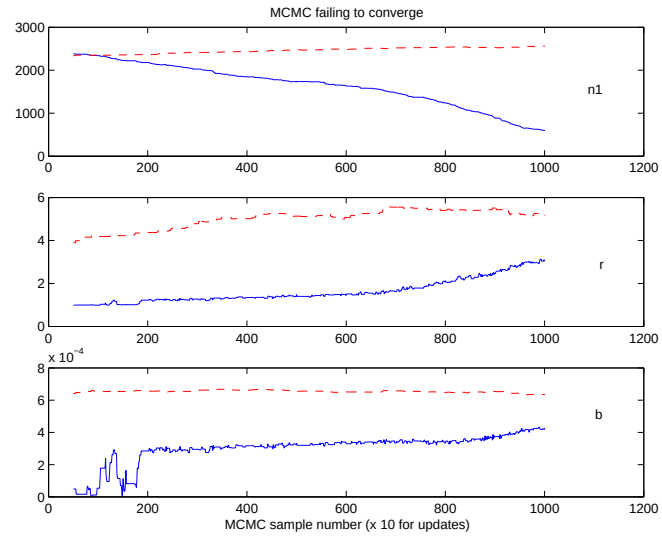

FIGURE 3. MCMC simulation of the posterior for the blowfly data. MCMC failing to converge - or mixing very slowly between two modes - two runs from the same start state are superposed! Much longer runs still fail to converge.

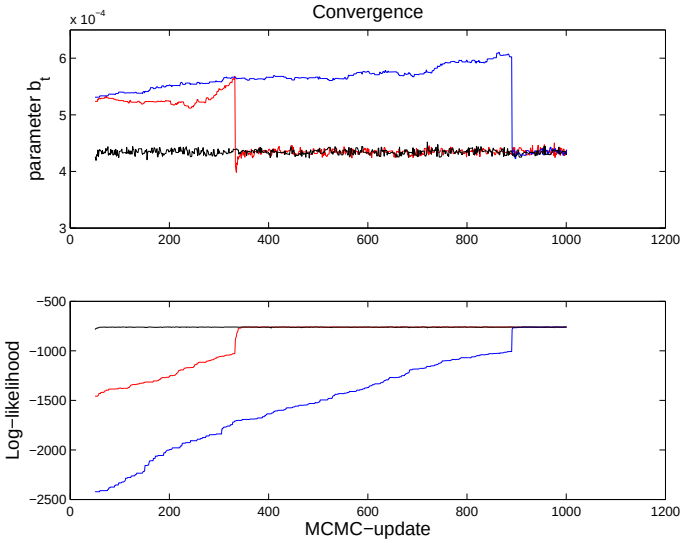

FIGURE 4. MCMC simulation of the posterior for the blowfly data. The x-axis is MCMC sample, multiply by 100 for MCMC updates. The MCMC is converging - just - three runs from three start states chosen deliberately to cause problems.

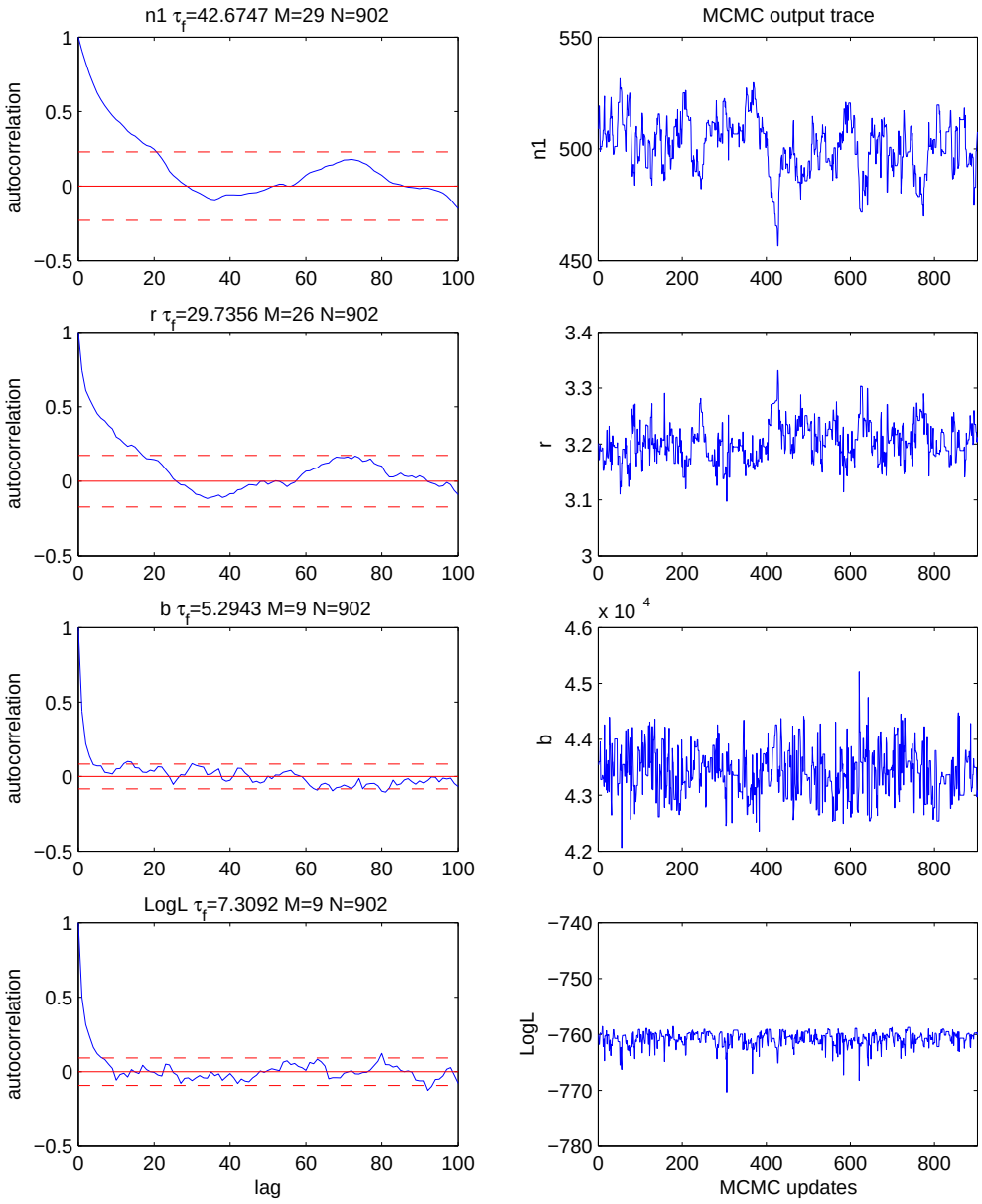

FIGURE 5. MCMC simulation of the posterior for the blowfly data. ACF's and MCMC simulation for the run used to compute the histograms in Figure 4. All x-axes are measured in samples, with 100 updates to a sample.

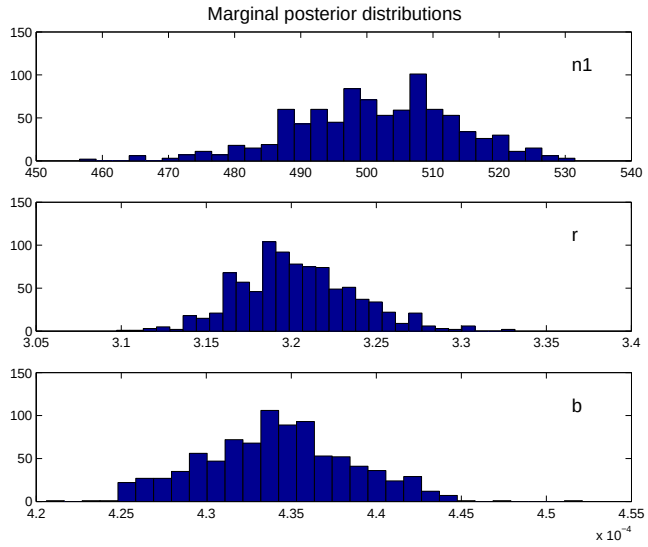

FIGURE 6. Results from MCMC simulation of the posterior for the blowfly data. Marginal posterior distributions for n_1, r and b.

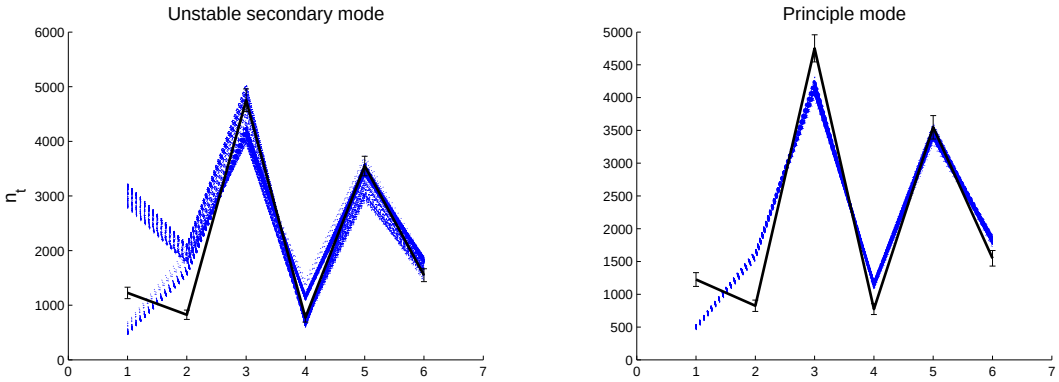

FIGURE 7. The iteration $n_{t+1} = rn_t/(1 + b^4 n_t^4)$ plotted for the values of n_1, r and b generated by the MCMC. Left, the converging chain including the meta-stable mode present in the burn-in. Right, samples from the converged chain. The meta-stable mode fits the later portion of the data well, but fails badly on the first few points. The parameters favored in MCMC equilibrium do not give a good fit either. See the model criticsm in the text.

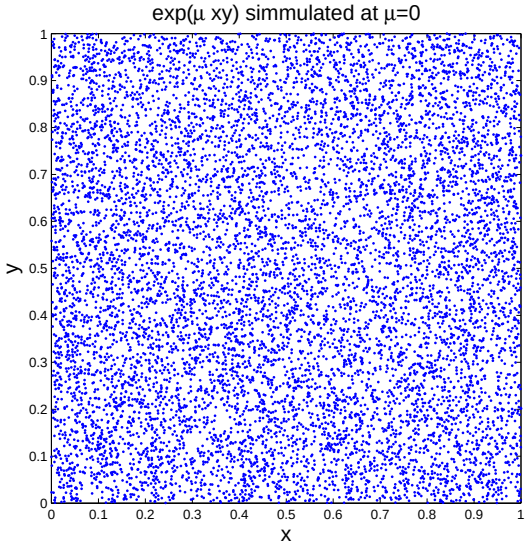

FIGURE 8. Results from MCMC simulation of $\exp(\mu xy)g(x,y)$ for $\mu = 0$ and $g = 1$, a simple case to test the scaling move. The distribution of points (x, y) is uniform in $(0, 1)^2$. The MCMC illustrated the use of a transformation to mix along a near symmetry of the target density.

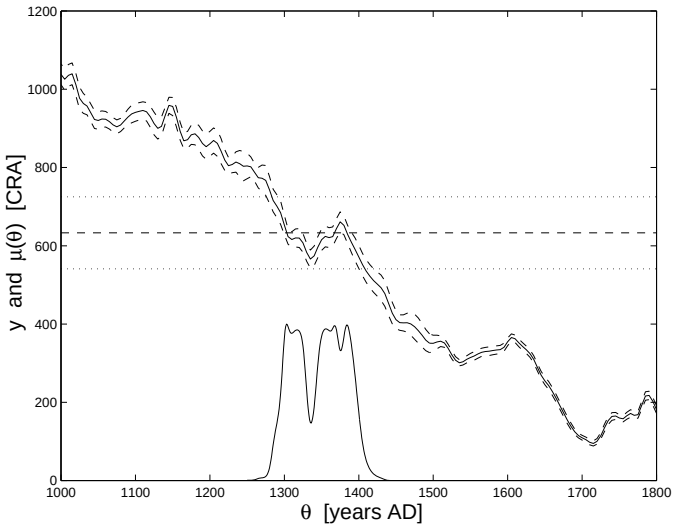

FIGURE 9. Radiocarbon dating example. Calibration function $\mu(\theta)$ (solid) with $2\hat{\sigma}(\theta)$ limits (dashed) for calibration of a terrestrial specimen, from Stuiver et al. (1998). Horizontal lines indicate a conventional radiocarbon age (CRA) determination, y_n, (dashed) with associated $2\tilde{\sigma}_n$ limits (dotted). The likelihood $\ell(y_n|\theta_n)$ (solid) is drawn below the calibration curve.

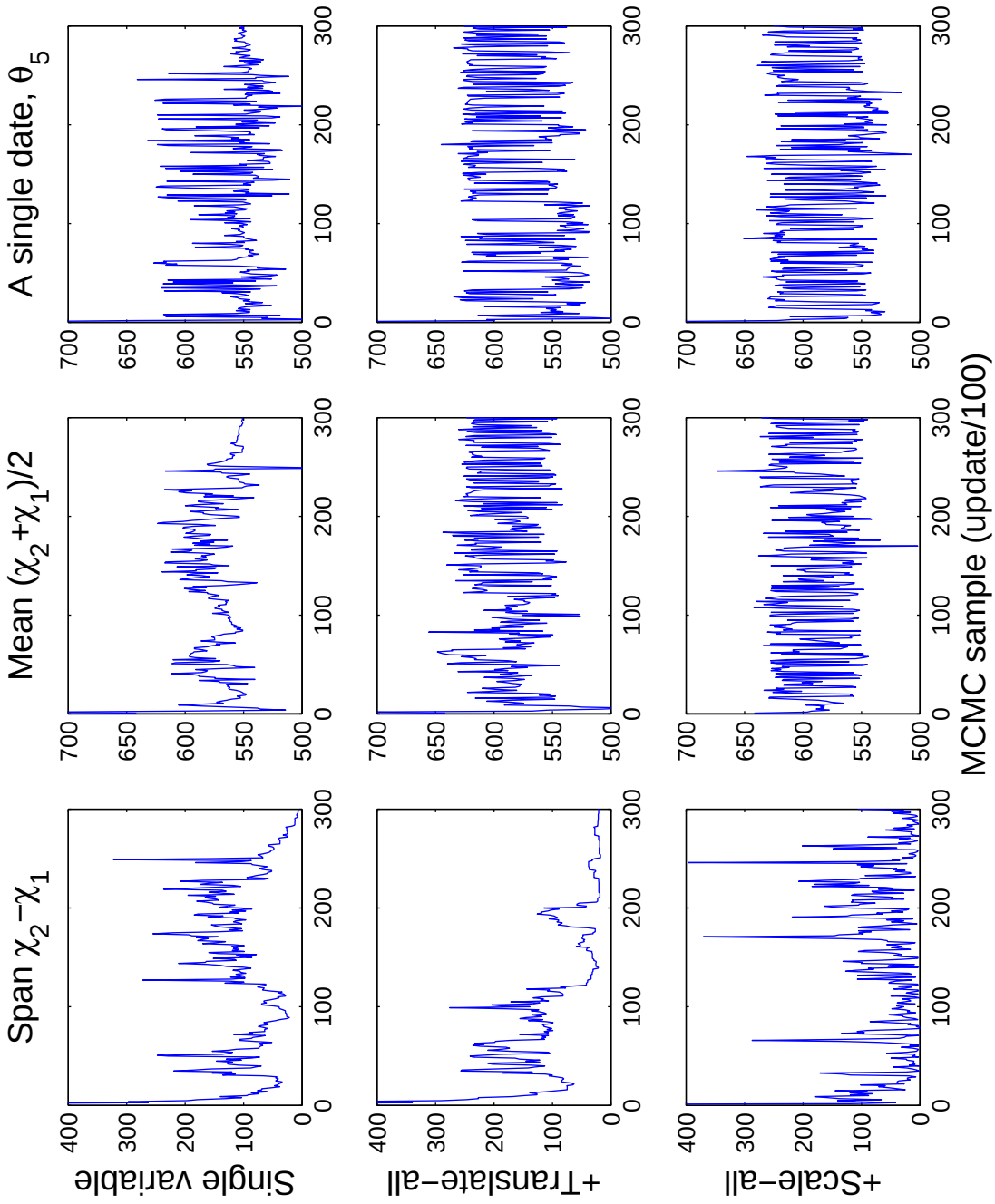

FIGURE 10. MCMC output for three statistics $(\chi_2-\chi_1$, $(\chi_2+\chi_1)/2$ and one of the date parameters, θ_5. Notice that move (2) of the text, the translation-update helps with the $(\chi_2+\chi_1)/2$-mixing while the scaling move, move (3) of the text, makes a dramatic difference to the quality of the $\chi_2 - \chi_1$-mixing. The posterior density is unbounded at $\chi_2 - \chi_1 = 0$.

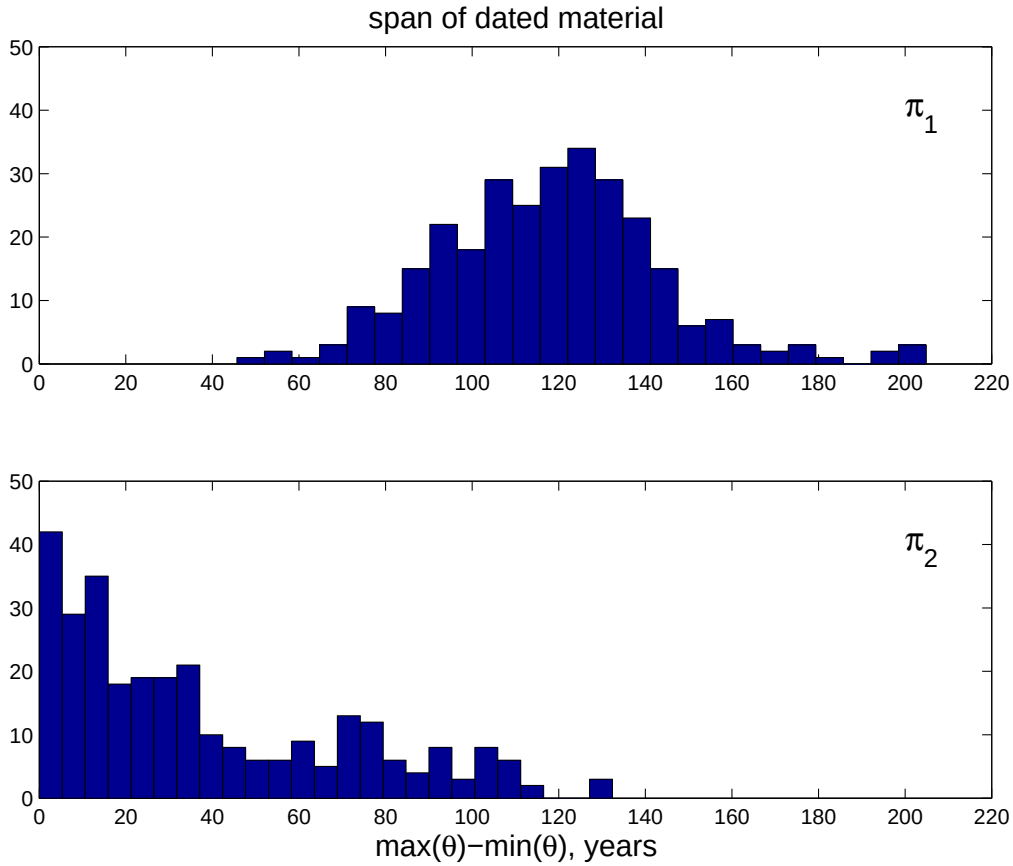

FIGURE 11. Posterior distributions for $\max(\theta) - \min(\theta)$. This quantity determines a lower bound on the span $\chi_2 - \chi_1$. The two histograms show the posterior distribution under the constant prior, π_1 and the uniform span prior π_2. The two priors lead to quite different interpretations of the data. The first prior rules out any short occupation time for the dated site.

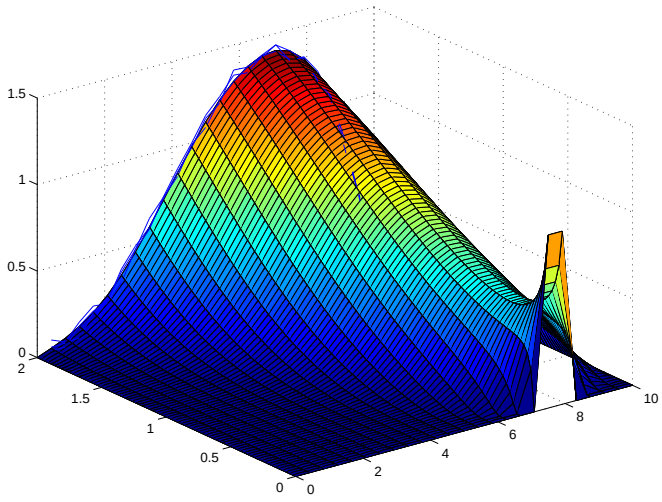

FIGURE 12. The time evolution of the population density (time runs into the page). The interval length is $L = 10$, diffusivity $D = 1.5$, per-capita reproduction $r = 1$ and the carrying capacity $K = 2$.

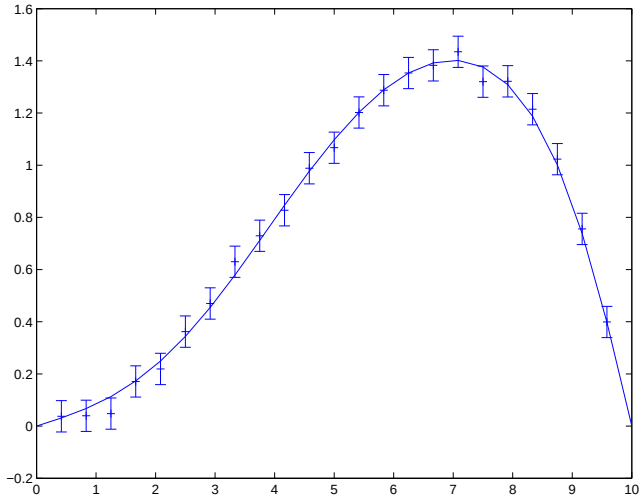

FIGURE 13. The data (x-axis is x, spatial position, y axis is u, population density), is measured in the final time slice. The observation noise is $s = 0.03$, and the data is gathered at time $T = 2$.

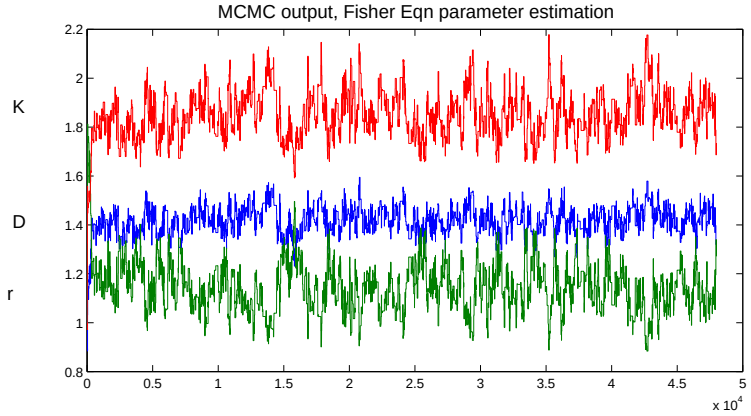

FIGURE 14. MCMC output for the posterior of the parameters of the Fisher equation.

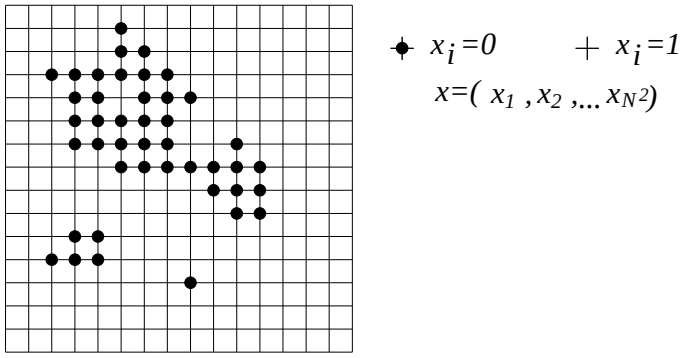

FIGURE 15. The Ising model on a square lattice with free boundaries.

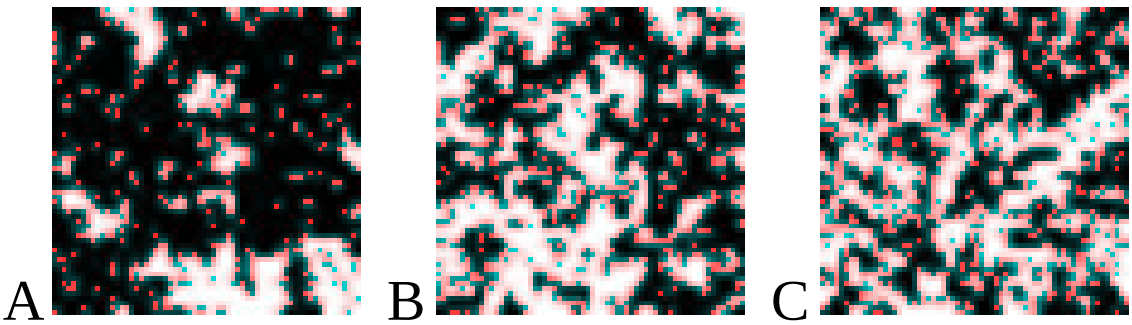

FIGURE 16. Samples $x \sim \exp(-2\theta \# x)/\mathcal{Z}_\theta$ with $N = 64$ and (A) $\theta = 0.45$ (B) $\theta = 0.4$ (C) $\theta = 0.35$.

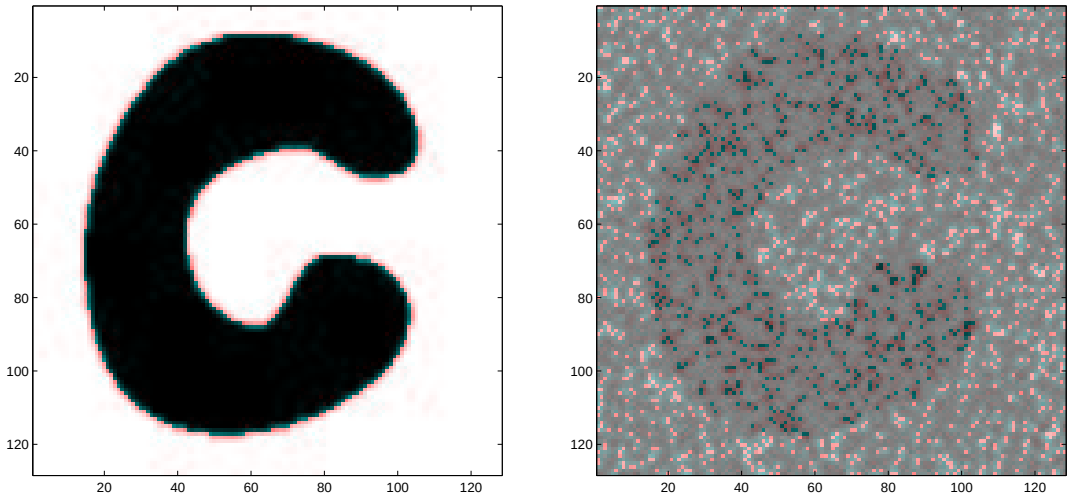

FIGURE 17. The true image for the pixel loss example (Left), and the pixels where data was recorded (Right) for $p = 0.9$ thinning.

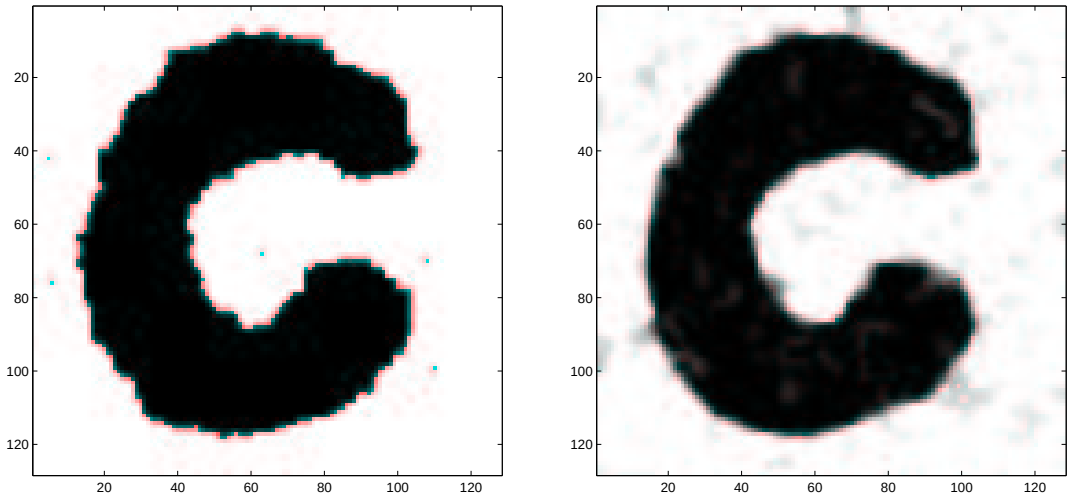

FIGURE 18. A sample from the posterior (Left) with smoothing parameter $\theta = 1$, and an estimate of the posterior mean (Right).