



STK4051/STK9051

Computational Statistics

Riccardo De Bin

`debin@math.uio.no`

Outline of the lecture

- Gibbs Sampling
 - Basic Gibbs Sampling
 - Properties of the Gibbs Sampler
 - Update Ordering
 - Blocking
 - Griddy-Gibbs Sampler
- Implementation
 - Ensuring Good Mixing and Convergence
 - Simple Graphical Diagnostics
 - Burn-in and Run Length
 - Choice of Proposal
 - Reparameterization

Gibbs Sampling: introduction

Gibbs Sampling:

- specifically adapted for multidimensional target distributions;
- goal: still to construct a Markov chain
 - ▶ whose stationary distribution – or some marginalization thereof – equals the target distribution, f ;
- sequentially samples from univariate conditional distributions,
 - ▶ often available in closed form.

Gibbs Sampling: Basic Gibbs sampler

Suppose one can sample from the univ. conditional distributions

$$X_i | X_1, \dots, X_{i-1}, X_{i+1}, \dots, X_p \sim f(x_i | x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_p)$$

A basic Gibbs sampler:

1. Select starting values $\mathbf{x}^{(0)}$ and set $t = 0$.
2. Generate, in turn,

$$X_1^{(t+1)} | \cdot \sim f(x_1 | x_2^{(t)}, \dots, x_p^{(t)})$$

$$X_2^{(t+1)} | \cdot \sim f(x_2 | x_1^{(t+1)}, x_3^{(t)}, \dots, x_p^{(t)})$$

$$\vdots$$

$$X_{p-1}^{(t+1)} | \cdot \sim f(x_{p-1} | x_1^{(t+1)}, x_2^{(t+1)}, \dots, x_{p-2}^{(t+1)}, x_p^{(t)})$$

$$X_p^{(t+1)} | \cdot \sim f(x_p | x_1^{(t+1)}, x_2^{(t+1)}, \dots, x_{p-1}^{(t+1)})$$

3. Increment t and go to step 2.

Gibbs Sampling: Fur Seal Pup Capture - Recapture Example

After centuries of severe population reductions due to commercial and subsistence hunting, the abundance of fur seals in New Zealand has been increasing in recent years.

- goal: estimate the number of pups in a fur seal colony;
- how: using a capture-recapture approach,
 - ▶ no single census attempt is likely accurate;
 - ▶ it is not necessary to try to capture most of the individuals;
 - ▶ the individuals captured in a census are released with a marker;
 - ▶ a capture of a marked individual during any subsequent census is termed a recapture;
 - ▶ population size can be estimated on the basis of the history of capture and recapture data;
- high recapture rates suggest a true population size not greatly exceeding the total number of unique individuals captured.

Gibbs Sampling: Fur Seal Pup Capture - Recapture Example

Let:

- N be the unknown population size to be estimated;
- I be the number of census attempts;
- $\mathbf{c} = (c_1, \dots, c_I)$ be the total number of captures (including recaptures) at each attempt;
- r be the total number of distinct animals ever captured.

Here we assume:

- the population is closed during the period of the sampling;
- i.e., deaths, births and migrations are inconsequential during this period.

Gibbs Sampling: Fur Seal Pup Capture - Recapture Example

In his example:

- we consider a model with separate, unknown capture probabilities for each census effort,

$$\alpha = (\alpha_1, \dots, \alpha_I).$$

- we assume all animals equally catchable on any one capture occasion, but capture probabilities may change over time;
- the likelihood for this model is

$$L(N, \alpha | \mathbf{c}, r) \propto \frac{N!}{(N-r)!} \prod_{i=1}^I \alpha_i^{c_i} (1 - \alpha_i)^{N-c_i}$$

- this model is sometimes called the $M(t)$ model.

Gibbs Sampling: Fur Seal Pup Capture - Recapture Example

Data:

- from a study on the Otago Peninsula of New Zealand;
- reasonable to assume the population of pups was closed;
- $I = 7$ census attempts during one season;
- $r = \sum_{i=1}^7 m_i = 84$ unique fur seals were observed;
- the number of pups captured and recaptured are

		Census attempt, i						
		1	2	3	4	5	6	7
Number captured	c_i	30	22	29	26	31	32	35
Number newly caught	m_i	30	8	17	7	9	8	5

Data from Givens & Hoeting (2013).

Gibbs Sampling: Fur Seal Pup Capture - Recapture Example

Estimation based on a hierarchical Bayesian framework:

- N and α are assumed to be a priori independent, with priors
 - ▶ N : noninformative Jeffreys prior $f(N) \propto 1/N$;
 - ▶ capture probabilities: $f(\alpha_i | \theta_1, \theta_2) = \text{Beta}(\theta_1, \theta_2)$ for $i = 1, \dots, 7$, assumed to be a priori exchangeable;
- fixing $\theta_1 = \theta_2 = \frac{1}{2}$ corresponds to the Jeffreys prior;
- the combination of a uniform prior for N and a Jeffreys prior for α_i is recommended when $I > 5$;
- this leads to a proper posterior distribution for the parameters when $I > 2$ and there is at least one recapture ($c_i - m_i > 1$).

Gibbs Sampling: Fur Seal Pup Capture - Recapture Example

A Gibbs sampler can then be constructed by simulating from the conditional posterior distributions

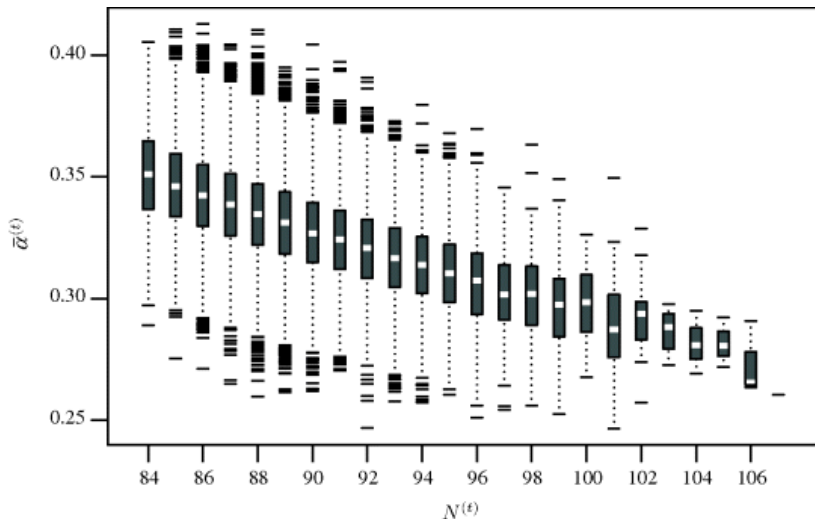
$$N - 84 | \cdot \sim \text{NegBin} \left(84, 1 - \prod_{i=1}^7 (1 - \alpha_i) \right)$$
$$\alpha_i | \cdot \sim \text{Beta} \left(c_i + \frac{1}{2}, N - c_i + \frac{1}{2} \right) \quad \text{for } i = 1, \dots, 7$$

where:

- $|\cdot$ denotes conditioning on the remaining parameters and data;
- NegBin denotes the negative binomial distribution.

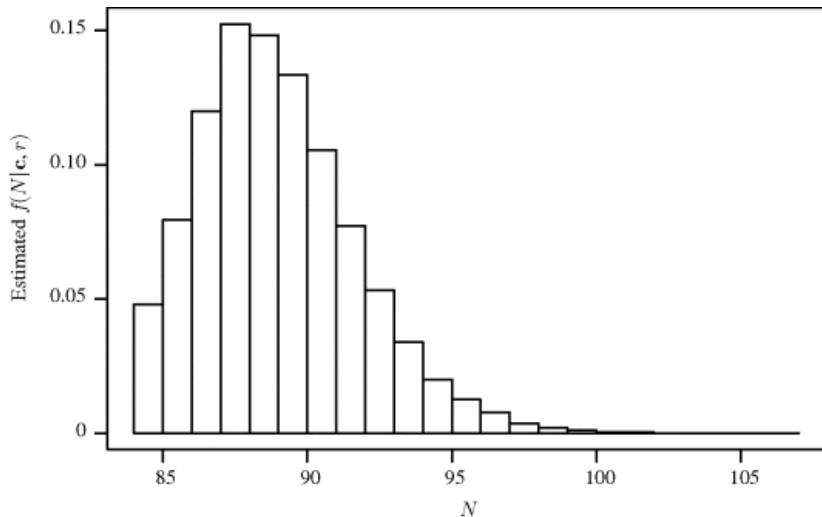
Results based on 100000 iterations (with the first 50000 iterations discarded for burn-in).

Gibbs Sampling: Fur Seal Pup Capture - Recapture Example



(Givens & Hoeting, 2013, figure 7.6)

Gibbs Sampling: Fur Seal Pup Capture - Recapture Example



(Givens & Hoeting, 2013, figure 7.7)

Gibbs Sampling: Notes and enhancements to the Gibbs sampler

Note:

- the completion of step 2 for all components is called **cycle**;
- the conditional distribution updates within each cycle can be reordered (**update ordering**);
- elements of $\mathbf{X}$ can be blocked (grouped) and updated jointly,
 - ▶ e.g., when $p = 4$, perhaps it is useful to consider

$$X_2^{(t+1)}, X_3^{(t+1)} \Big| \cdot \sim f \left(x_2, x_3 \mid x_1^{(t+1)}, x_4^{(t)} \right)$$

- ▶ in the book it is called **blocking**;
- the Gibbs sampling is a Metropolis-Hastings algorithm with time inhomogeneous proposal distributions and with Metropolis-Hasting ratio R always equal to 1.

Gibbs Sampling: Hybrid Gibbs sampling

A **hybrid MCMC** algorithm might proceed with the following sequence of updates with $p = 6$:

1. Update $X_1^{(t+1)} \mid \left(x_2^{(t)}, x_3^{(t)}, x_4^{(t)}, x_5^{(t)}, x_6^{(t)} \right)$ with a Gibbs step.
2. Update $\left(X_2^{(t+1)}, X_3^{(t+1)} \right) \mid \left(x_1^{(t+1)}, x_4^{(t)}, x_5^{(t)}, x_6^{(t)} \right)$ with a Metropolis step.
3. Update $X_4^{(t+1)} \mid \left(x_1^{(t+1)}, x_2^{(t+1)}, x_3^{(t+1)}, x_5^{(t)}, x_6^{(t)} \right)$ with a step from a random walk chain.
4. Update $\left(X_5^{(t+1)}, X_6^{(t+1)} \right) \mid \left(x_1^{(t+1)}, x_2^{(t+1)}, x_3^{(t+1)}, x_4^{(t+1)} \right)$ with a Gibbs step.

The Metropolis-Hastings steps within a Gibbs algorithm are typically useful when the univariate conditional density for one or more elements of $\mathbf{X}$ is not available in closed form.

Gibbs Sampling: Fur Seal Pup Capture - Recapture Example

In the previous example we set:

- a Beta prior distribution for the capture probabilities;
- a noninformative Jeffreys prior for N , so $f(N) \propto 1/N$.

Results are sensitive to the values selected for θ_1 and θ_2 :

- alternative setup to mitigate this sensitivity;
- $f(\theta_1, \theta_2) \propto \exp\{-(\theta_1 + \theta_2)/1000\}$;
- (θ_1, θ_2) are assumed to be a priori independent of the remaining parameters.

Gibbs Sampling: Fur Seal Pup Capture - Recapture Example

The Gibbs sampler can then be constructed by simulating from the conditional posterior distributions:

$$N - 84 | \cdot \sim \text{NegBin}\left(84, 1 - \prod_{i=1}^7 (1 - \alpha_i)\right)$$

$$\alpha_i | \cdot \sim \text{Beta}(c_i + \theta_1, N - c_i + \theta_2) \quad \text{for } i = 1, \dots, 7$$

$$\theta_1, \theta_2 | \cdot \sim k \left[\frac{\Gamma(\theta_1 + \theta_2)}{\Gamma(\theta_1)\Gamma(\theta_2)} \right]^7 \prod_{i=1}^7 \alpha_i^{\theta_1} (1 - \alpha_i)^{\theta_2} \exp\{-(\theta_1 + \theta_2)/1000\}$$

where k is an unknown constant.

Note that the last distribution is not easy to sample:

- use a Metropolis-Hastings step to do that;
- thus the Gibbs sampler becomes a hybrid Gibbs sampler;
- this happens when a prior distribution is used for θ_1 and θ_2 .

Gibbs Sampling: Griddy-Gibbs Sampler

Suppose that it is difficult to sample from the univariate conditional density for $X_k|x_{-k}$ for a particular k . To implement a griddy-Gibbs step,

- select some grid points $z_1, \dots, z_n$ over the support of $f(\cdot|x_{-k})$;
- let $w_j^{(t)} = f(z_j|x_{-k}^{(t)})$ for $j = 1, \dots, n$;
- using these weights and the corresponding grid, one can approximate the density function $f(\cdot|x_{-k})$;
- generate from this approximation;
- proceed with the remainder of the MCMC algorithm.

Gibbs Sampling: Griddy-Gibbs Sampler

Note:

- the approximation to the k -th univariate conditional can be refined as iterations proceed,
 - ▶ then the chain is not time homogeneous;
 - ▶ no guarantee that a griddy-Gibbs chain has a limiting stationary distribution equal to f ;
- to ensure time homogeneity, do not make any improvement to the approximating univariate distribution,
 - ▶ the limiting distribution of the chain is still not correct;
 - ▶ relies on an approximation to $f(\cdot|x_{-k})$, not on the true density;
 - ▶ revert to a hybrid Metropolis-within-Gibbs framework: the variable generated from the approximation to $f(\cdot|x_{-k})$ is viewed as a proposal, which is then randomly kept or discarded based on the Metropolis-Hastings ratio

Implementation: issues

All MCMC algorithms we saw have the correct limiting stationary distribution.

Questions we may ask ourselves:

- how quickly does the chain forget its starting value?
- how quickly does the chain fully explore the support of the target distribution?
- how far apart in a sequence do observations need to be before they can be considered to be approximately independent?
- has the chain has run sufficiently long so that:
 - ▶ it is reasonable to believe that the output adequately represents the target distribution?
 - ▶ the output can be used reliably for estimation?

Ensuring Good Mixing and Convergence: summary

Strategies and diagnostics:

1. simple graphical diagnostics;
2. burn-in and run length;
3. choice of proposal;
4. reparameterization;
5. effective sample size;
6. number of chains.

Ensuring Good Mixing and Convergence: Simple Graphical Diagnostics

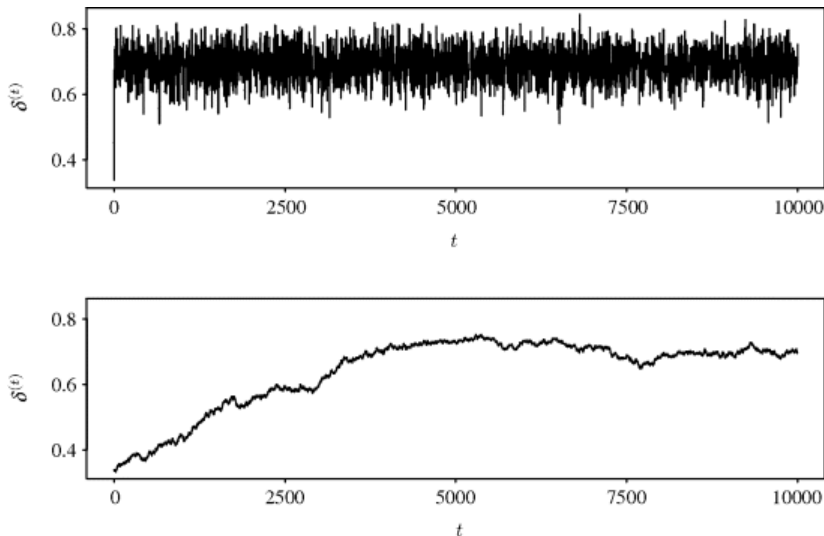
Sample paths:

- iteration number versus realizations of $\mathbf{X}^{(t)}$ for $t = 0, 1, \dots$
- also called trace or history plots;
- we want to see a very wiggly line that (perhaps) moves away from the starting value.

Autocorrelation function (ACF) plot:

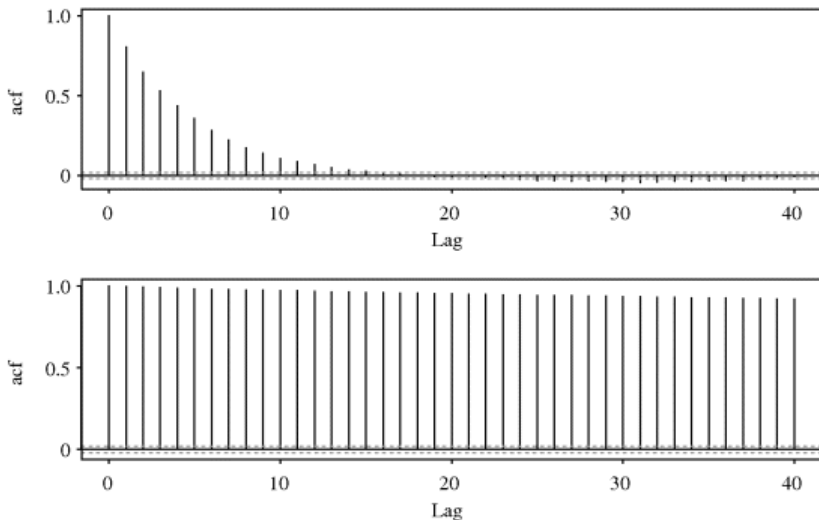
- the lag versus the correlation between iterates at that lag;
- slow decay in the acf suggests poor mixing;
- multi-parameter problems:
 - ▶ examine cross-correlations between related parameters;
 - ▶ high cross-correlations may indicate poor mixing of the chain.

Ensuring Good Mixing and Convergence: Simple Graphical Diagnostics



(Givens & Hoeting, 2013, figure 7.5)

Ensuring Good Mixing and Convergence: Simple Graphical Diagnostics



(Givens & Hoeting, 2013, figure 7.8)

Burn-in and run length: Burn-in

Burn-in:

- initial MCMC iterates,
 - do not have exactly the correct marginal distribution;
 - strong dependency on the initial point (or distribution);
- to reduce the severity of this problem, the first several thousand values from the chain are typically discarded.

Gelman and Rubin statistic, R :

- commonly used to determine burn-in period and run length;
- motivated by an analysis of variance (ANOVA);
- based on the comparison of the output of several chains;
- the burn-in period or MCMC run-length should be increased if
 - between-chain variance $\gg$ the within-chain variance.

Burn-in and run length: Gelman and Rubin statistic

Initial steps:

- run the MCMC algorithm J separate times;
- to produce equal-length chains ($J \geq 2$);
- with starting values all over the support of the target density.

Notation:

- L is the length of each chain w/o the D burn-in iterations;
- X is the variable of interest,
 - ▶ with $x_j^{(t)}$ be its value at the t th iteration of the j th chain;
- for the j th chain,
 - ▶ the D values $x_j^{(0)}, \dots, x_j^{(D-1)}$ are discarded;
 - ▶ the L values $x_j^{(D)}, \dots, x_j^{(D+L-1)}$ are retained.

Burn-in and run length: Gelman and Rubin statistic

Let $\bar{x}_j = \frac{1}{L} \sum_{t=D}^{D+L-1} x_j^{(t)}$ and $\bar{x}_{\cdot} = \frac{1}{J} \sum_{j=1}^J \bar{x}_j$ and define the between-chain variance as

$$B = \frac{L}{J-1} \sum_{j=1}^J (\bar{x}_j - \bar{x}_{\cdot})^2.$$

Now if we define the within-chain variance for the j th chain to be $s_j^2 = \frac{1}{L-1} \sum_{t=D}^{D+L-1} (x_j^{(t)} - \bar{x}_j)^2$, then

$$W = \frac{1}{J} \sum_{j=1}^J s_j^2$$

represents the mean of the J within-chain estimated variances.

Burn-in and run length: Gelman and Rubin statistic

With these pieces, we can define the Gelman and Rubin statistic as

$$R = \frac{\frac{L-1}{L}W + \frac{1}{L}B}{W}.$$

Note:

- if all the chains are stationary, then both the numerator and the denominator should estimate the marginal variance of X ;
- if there are notable differences between the chains, then the numerator will exceed the denominator;
- in theory, $\sqrt{R} \rightarrow 1$ as $L \rightarrow \infty$,
 - ▶ true for the adjusted estimator $\hat{R} = \frac{J+1}{J}R + \frac{L-1}{JL}$;
- practical advice: $\sqrt{R} < 1.2$ is acceptable (or $\sqrt{\hat{R}} < 1.1$).

Burn-in and run length: Gelman and Rubin statistic

Remarks:

- if the chosen burn-in period did not yield an acceptable result,
 - ▶ D should be increased;
 - ▶ L should be increased;
 - ▶ or preferably both D and L should be increased;
- if the $x_j^{(t)}$ iterates are transformed so that their distribution is approximately normal, better performance of this diagnostic;
- reparameterize the model and rerun the chain.

Possible issues:

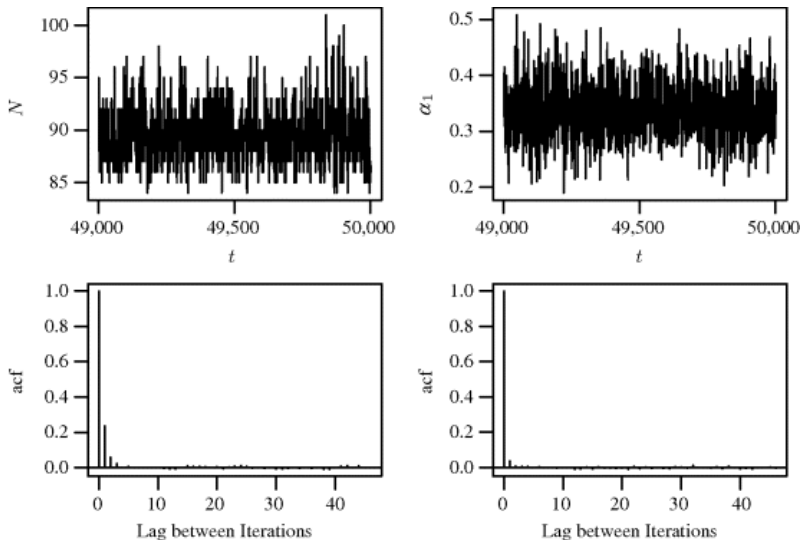
- hard to select suitable starting values,
 - ▶ especially tricky in the case of multimodal f ;
 - ▶ the procedure does not work if all of the chains become stuck in the same subregion or mode;
- possible misleading results when f is multidimensional.

Implementation: Fur Seal Pup Capture - Recapture Example

Diagnostics in the Fur Seal Pup Capture - Recapture Example:

- basic Gibbs sampler;
- $J = 5$ runs of 100000 iterations, each with a burn-in of 50000;
- for N :
 - ▶ the Gelman–Rubin statistic is equal to 0.999995;
 - ▶ the chain $N^{(t)}$ is roughly stationary;
- for $(\alpha_1, \dots, \alpha_7)$:
 - ▶ the Gelman–Rubin statistics are all close to one;
 - ▶ the capture probabilities exhibit little correlation between MCMC samples.

Implementation: Fur Seal Pup Capture - Recapture Example



(Givens & Hoeting, 2013, figure 7.9)

Ensuring Good Mixing and Convergence: Choice of Proposal

MCMC algorithms are affected by the proposal distribution:

- aim for a proposal distribution that is more diffuse than f ,
 - ▶ tail behaviour of g more important than its resemblance to f ;
 - ▶ faster convergence if f/g is bounded;
- the variance of the proposal distribution can be selected through an informal iterative process:
 - ▶ start a chain, and monitor the proportion of acceptances;
 - ▶ adjust the spread of the proposal distribution accordingly;
- for a Metropolis algorithm with normal target and proposal distributions, aim for an acceptance between 25 and 50%,
 - ▶ best choice for one-dimensional problems being about 44%;
 - ▶ about 23.4% for higher-dimensional problems.

Ensuring Good Mixing and Convergence: Reparameterization

To apply such rules, care must be taken:

- target and proposal distributions must be roughly normal;
- at least assure they have simple, unimodal distributions;
- if the target distribution is multimodal,
 - the chain may get stuck in one mode;
 - without adequately explore the rest of the parameter space;
- issue (for most MCMC methods): we need as much knowledge as possible about the unknown target distribution.

Advanced methods (e.g., Adaptive Markov chain Monte Carlo) tune the proposal distribution during the MCMC algorithm:

- in some implementations, do not require the user to stop, tune, and restart the algorithm on multiple occasions.

Ensuring Good Mixing and Convergence: Reparameterization

Model reparameterization can provide substantial improvements:

- high correlation between the elements of $\mathbf{X}$ can slow down the convergence;
- reparameterization is the primary strategy for reducing it;
- example with a Gibbs sampler:
 - ▶ f bivariate normal with very strong positive correlation;
 - ▶ both univariate conditionals will allow only small steps;
 - ▶ $\mathbf{Y} = (\mathbf{X}_1 + \mathbf{X}_2, \mathbf{X}_1 - \mathbf{X}_2)$ yields one univariate conditional on the axis of maximal variation in $\mathbf{X}$;
 - ▶ easier to move from one point to any other point.

Reparameterization approaches are typically model specific:

- center and scale the covariates to reduce correlations;
- use hierarchical centering: parameters in a hierarchical Bayesian model are moved down the hierarchy.

Implementation: Fur Seal Pup Capture - Recapture Example

Fur Seal Pup Capture - Recapture Ex. with hybrid Gibbs sampler:

- a Metropolis-Hastings step (with random walk) is required to sample (θ_1, θ_2) from the conditional posterior;
- the prior distribution restricts (θ_1, θ_2) to be larger than 0;
- this constraint can impede MCMC performance, particularly if there is high posterior density near the boundary;
- to improve performance, transform (θ_1, θ_2) to

$$\mathbf{U} = (U_1, U_2) = (\log \theta_1, \log \theta_2);$$

- proposal from $N(0, 0.085^2 I_2) \rightarrow$ acceptance rate $\approx 23\%$;
- this requires transformation of the conditional densities. . .

Implementation: Fur Seal Pup Capture - Recapture Example

Reparametrised conditional posterior densities:

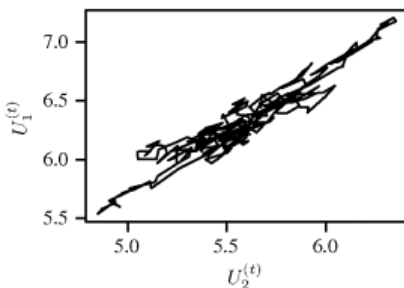
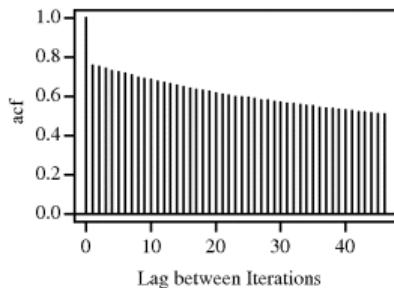
$$\begin{aligned} N - 84 | \cdot &\sim \text{NegBin} \left(84, 1 - \prod_{i=1}^7 (1 - \alpha_i) \right) \\ \alpha_i | \cdot &\sim \text{Beta}(c_i + \exp\{u_1\}, N - c_i + \exp\{u_2\}) \quad \text{for } i = 1, \dots, 7 \\ \theta_1, \theta_2 | \cdot &\sim k_u \exp\{u_1 + u_2\} \left[\frac{\Gamma(\exp\{u_1\} + \exp\{u_2\})}{\Gamma(\exp\{u_1\})\Gamma(\exp\{u_2\})} \right]^7 \\ &\quad \prod_{i=1}^7 \alpha_i^{\exp\{u_1\}} (1 - \alpha_i)^{\exp\{u_2\}} \exp \left\{ -\frac{\exp\{u_1\} + \exp\{u_2\}}{1000} \right\}. \end{aligned}$$

Implementation: Fur Seal Pup Capture - Recapture Example

Results based on a chain of 100000 iterations with the first 50000 iterations discarded for burn-in:

- the Gelman and Rubin statistic for N is equal to 1.00047;
- which suggests the $N^{(t)}$ chain is roughly stationary;
- however, the autocorrelation plots indicate high correlation between iterations,
 - ▶ correlation of 0.6 between MCMC samples 40 iterations apart;
 - ▶ for comparison, it was 0 for lag 2 in the simple Gibbs sampler;
- the bivariate sample path of $U_1^{(t)}$ vs $U_2^{(t)}$ indicates a high correlation between the parameters too;
- lack of convergence of the MCMC algorithm (or at least poor behaviour of the chain) for the hybrid algorithm.

Implementation: Fur Seal Pup Capture - Recapture Example



(Givens & Hoeting, 2013, figure 7.10)

References |

GIVENS, G. H. & HOETING, J. A. (2013). *Computational Statistics*. Wiley.