Markov chain Monte Carlo

Jo Eidsvik

Mathematical Sciences, NTNU

Markov chain Monte Carlo (MCMC)

- Usual Stochastic Processes Markov chains; a transition matrix P from which one can (sometimes) derive the (unique) stationary distribution π.
- MCMC, one would like to find, for a given distribution π, the transitions P which has π as limiting distribution.

This has become a very popular method for Monte Carlo sampling during the last $20 - 30$ years. (Hastings original paper was in 1970.)

Monte Carlo sampling

Suppose we want to approximate some expectation:

$$m = E[f(x)] = \sum_x f(x)\pi_x$$

We assume it is relatively easy to simulate from the π distribution.
Monte Carlo algorithm:

- Sample x^b, $b = 1, \ldots, B$ from distribution π.
- Approximate by averaging:

$$\hat{m} = \frac{1}{B} \sum_{b=1}^{B} f(x^b)$$

Under weak regularity conditions, $\hat{m} \to m$, when $B \to \infty$.

MCMC

MCMC uses dependent samples to do Monte Carlo approximations.
Idea: Construct a Markov chain that converges to π. Then take sample averages. The first (transient) part (called burn-in) is discarded in the sample averages because it is would be biased from the initial state.

Requirements of Markov chain

1. Markov chain must converge to right limiting density or probability mass function $\pi(\boldsymbol{x})$
2. Markov chain must stay in the right stationary density or probability mass function $\pi(\boldsymbol{x})$
3. Irreducible, Aperiodic chain
4. For asymptotic properties of integral approximation we require ergodicity and recurrence.

MCMC

There are two main algorithms.

- **Gibbs sampling.**
- **Metropolis-Hastings sampling.**

Gibbs sampler

Gibbs sampling is one MCMC method. Transition matrices (kernel) are composed of **full conditionals**.
Algorithm

- Initiate $\boldsymbol{x}^0$, $b = 0$.
- Iterate while $b < B$,
  1. Pick an element i of vector $\boldsymbol{x}$.
  2. Sample $x_i^{b+1} \sim \pi(x_i | \boldsymbol{x}_{-i}^b)$
  3. Set $x_j^{b+1} = x_j^b$ for all $j \neq i$.
  4. $b = b + 1$

Gibbs sampler for $N_2(0, \mathbf{\Sigma})$

For the bivariate Gaussian with mean 0, identity variance, and correlation ρ, the Gibbs sampler goes as follows:
Algorithm

- Initiate $\boldsymbol{x}^0 = (x_1^0, x_2^0)$, $b = 0$.
- Iterate while $b < B$,
  1. Sample $x_1^{b+1} \sim N(\rho x_2^b, 1 - \rho^2)$
  2. Sample $x_2^{b+1} \sim N(\rho x_1^{b+1}, 1 - \rho^2)$
  3. $b = b + 1$

Gibbs sampler

The Gibbs sampler has been extremely successful :

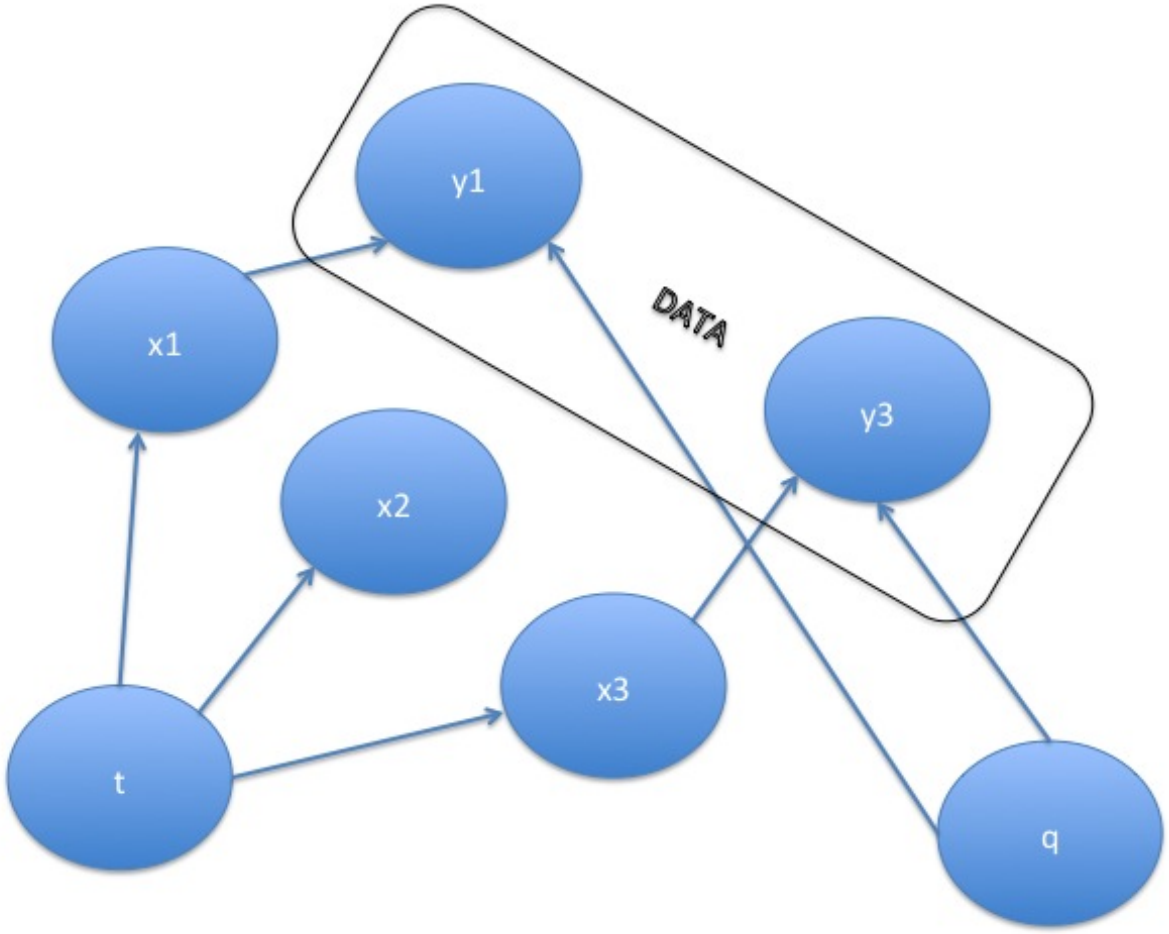
- In large graphical models
- When full conditionals are directly available
- When split-and-conquer is feasible by conjugate prior-posteriors.
- When model dependence / interaction is not too large, so that convergence and mixing is not too poor.

Gibbs sampler: Example

Graphical model:

- Success probability $p(t) = \text{Beta}(a, b)$
- Individual level variable $x_i \in \{0, 1\}$, $p(x_i = 1|t) = t$, $i = 1, \ldots, n$.
- Response variable $p(y_i|x_i, q) = N(x_i, q^{-1})$.
- Precision of measurement $p(q) = \text{Gamma}(\alpha, \beta)$.

Gibbs sampler: Example



Gibbs sampler: Example

This model is very common in 'mixed' models applied in medicine, geostatistics, finance, etc. The hierarchical model imposes dependence through common parent nodes and conditional independence. It is ideal for splitting the model building in different stages.

- $p(x_i|t, x_1, \ldots, x_{i-1}) = p(x_i|t)$.
- $p(q|t, x_1, \ldots, x_n) = p(q)$
- $p(y_i|t, q, x_1, \ldots, x_n, y_1, \ldots, y_{i-1}) = p(y_i|x_i, q)$.

Joint model $\prod_i [p(y_i|x_i, q)p(x_i|q)]\, p(q)p(t)$.

Gibbs sampler: Example

Goal is posterior $\pi(t, x_1, \ldots, x_n, q|y)$. We cannot explore this directly, but we can sample iteratively from all full conditionals. These also simplify because of conditional independence. We need

- $p(t|x_1, \ldots, x_n) = \text{Beta}(a + \sum_i x_i, b + n - \sum_i x_i)$.
- $p(q|x_1, \ldots, x_n, y_1, \ldots, y_n) = \text{Gamma}(\alpha + n/2, \beta + \frac{\sum_i (y_i - x_i)^2}{2})$
- $p(x_i|t, q, y_i) \propto p(x_i|t)p(y_i|x_i, q)$.
- $p(x_i = 1|t, q, y_i) = \frac{t \exp(-q\frac{(y_i-1)^2}{2})}{t \exp(-q\frac{(y_i-1)^2}{2}) + (1-t)\exp(-q\frac{y_i^2}{2})}$

(Gibbs sampling iterates between sampling from these.)

Metropolis-Hastings algorithm

(Assuming discrete state-space S, but works generally.)

- Initialize x^0.
- Iterate the following $b = 1, \ldots, B$:
  1. Sample $U \sim U(0, 1)$
  2. Propose a new potential state x^*, from proposal distribution $P(x^* = j | x^{b-1} = i) = Q_{ij}$, $j \in S$.
  3. Compute the acceptance probability: $\alpha_{ij} = \min\left(1, \frac{\pi_j}{\pi_i}\frac{Q_{ji}}{Q_{ij}}\right)$
  4. Set $x^b = x^* = j$ if $U < \alpha_{ij}$. Else set $x^b = i$.

We are free to choose Q_{ij}. This gives enormous flexibility! The art is to select this wisely! If we do so, Metropolis will *beat* Gibbs (Performance metrics later.)

Random walk MH for $N_2(0, \mathbf{\Sigma})$

For the bivariate Gaussian with mean 0, identity variance, and correlation ρ, the Random walk MH sampler goes as follows:
Algorithm

- Initiate $\boldsymbol{x}^0 = (x_1^0, x_2^0)$, $b = 0$.
- Iterate while $b < B$,
  1. Sample $\boldsymbol{x}^* \sim N(\boldsymbol{x}^b, \sigma^2 \boldsymbol{I})$
  2. Accept $\boldsymbol{x}^{b+1} = \boldsymbol{x}^*$ with probability $\alpha = \min\{1, \exp(-1/2 \boldsymbol{x}^{*t} \mathbf{\Sigma}^{-1} \boldsymbol{x}^* + 1/2 \boldsymbol{x}^{b,t} \mathbf{\Sigma}^{-1} \boldsymbol{x}^b)\}$
     Else set $\boldsymbol{x}^{b+1} = \boldsymbol{x}^b$.
  3. $b = b + 1$

The symmetric proposal density cancels in the acceptance rate.

Reversibility

Metropolis–Hastings gives reversible Markov chains.
A Markov chain is called time reversible if

$$\pi_i P_{ij} = \pi_j P_{ji} \quad \text{for all } i, j$$

The condition is often called *detailed balance*.

Reversibility - forward backward

- Time reversibility means that we can run chain backward and we do not see difference in transition dynamics.
- $P_{i,i_1} P_{i_1,i_2} \dots P_{i_n,j} = P_{j,i_n} \dots P_{i_2,i_1} P_{i_1,i}$ for any path of states.

$$P_{\text{back},ji} = P(X_n = i | \, X_{n+1} = j) = \frac{P(X_n = i, X_{n+1} = j)}{P(X_{n+1} = j)} = \frac{\pi_i}{\pi_j} P_{ij}$$

Time-reversibility means $P_{\text{back},ji} = P_{ji}$. This is detailed balance equation.

Reversibility for Metropolis–Hastings

For $i \neq j$:

$$\pi_i P_{ij} = \pi_i Q_{ij} \alpha_{ij} = \min\{\pi_i Q_{ij}, \pi_i Q_{ij} \frac{\pi_j Q_{ji}}{\pi_i Q_{ij}}\} = \min\{\pi_i Q_{ij}, \pi_j Q_{ji}\} = \pi_j P_{ji}$$

Reversibility for Metropolis–Hastings

For $i \neq j$:

$$\pi_i P_{ij} = \pi_i Q_{ij} \alpha_{ij} = \min\{\pi_i Q_{ij}, \pi_i Q_{ij} \frac{\pi_j Q_{ji}}{\pi_i Q_{ij}}\} = \min\{\pi_i Q_{ij}, \pi_j Q_{ji}\} = \pi_j P_{ji}$$

One could of course imagine non-reversible chains giving a unique limiting distribution. This has not been done much.

Choice of MH proposal distribution

This is very case specific.

- Want large changes.
- Want high acceptance probability.
- Want low computational time for proposal and evaluation.
- It is difficult to get all of the above.

Famous proposals for continuous target distributions

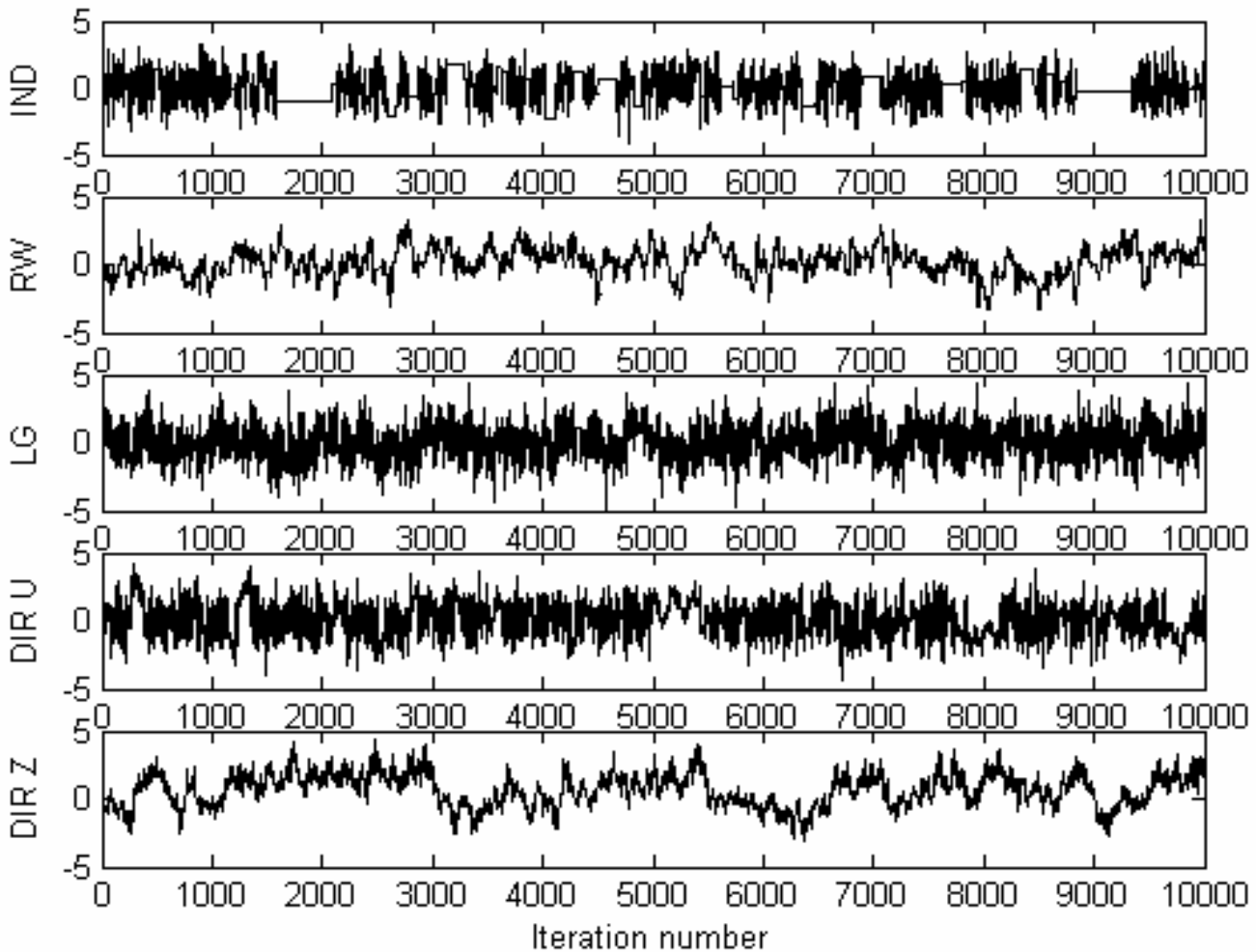
- Random walk: $q(\boldsymbol{x}|\boldsymbol{x}^b) = N(\boldsymbol{x}^b, \sigma^2 \boldsymbol{I})$. Only requires tuning of σ. Asymptotically optimal (under some assumptions) to have acceptance rate about 0.23.
- Langevin: $q(\boldsymbol{x}|\boldsymbol{x}^b) = N(\boldsymbol{x}^b + \frac{\sigma^2}{2}\nabla \log \pi(\boldsymbol{x}^b), \sigma^2 \boldsymbol{I})$. Only requires tuning of σ. Asymptotically optimal (under some assumptions) to have acceptance rate about 0.57.
- Independent proposal: $q(\boldsymbol{x}|\boldsymbol{x}^b) = N(\mu, \boldsymbol{\Sigma})$, where $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ can include derivatives of $\log \pi(\boldsymbol{x}^b)$, or some initial approximation.
- It is common to have a hybrid mix of the above, i.e. the proposal mechanism vary with iterations.

Mixing of the MCMC sampler

MCMC gives *dependent* samplers from the density or probability mass function π.
If the dependence (autocorrelation) is very large, the Markov chain is said to mix slowly, and it takes a very long time to explore the sample space adequately.

Trace plot

From top; Independent proposal, Random walk, Langevin



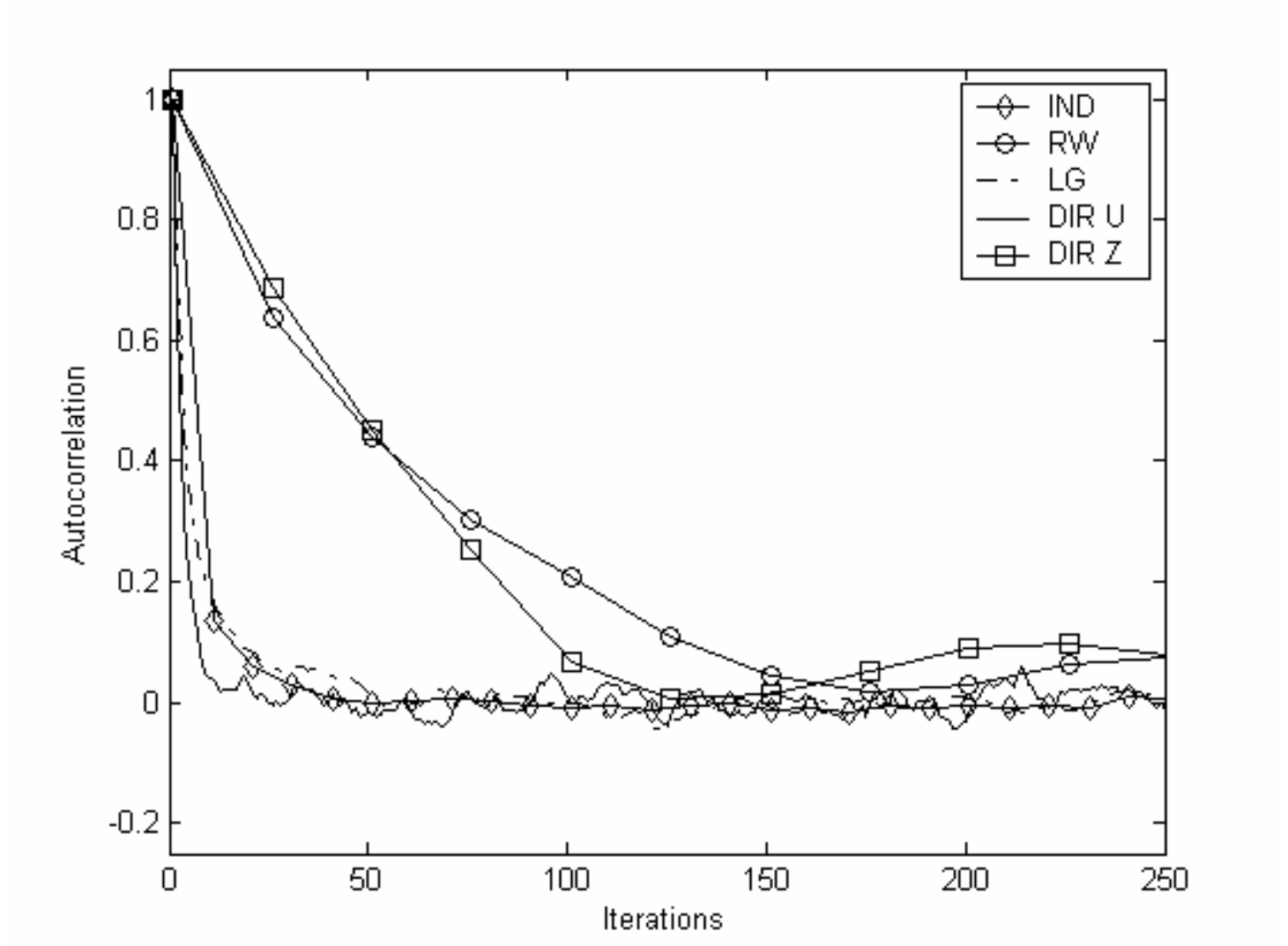
Autocorrlation in the MCMC sampler

Integrated autocorrelation:

$$\mathrm{IAC} = 1 + 2 \sum_{t=1}^{2T+1} \hat{\rho}_t, \quad \hat{\rho}_t = \hat{\mathrm{Corr}}(x_s, x_{s+t}),$$

$$T = \max(\tau; \hat{\rho}_{2t} + \hat{\rho}_{2t+1} > 0 \ \text{ for all } t \leq \tau)$$

Autocorrelation



Mixing vs evaluation of the MCMC sampler

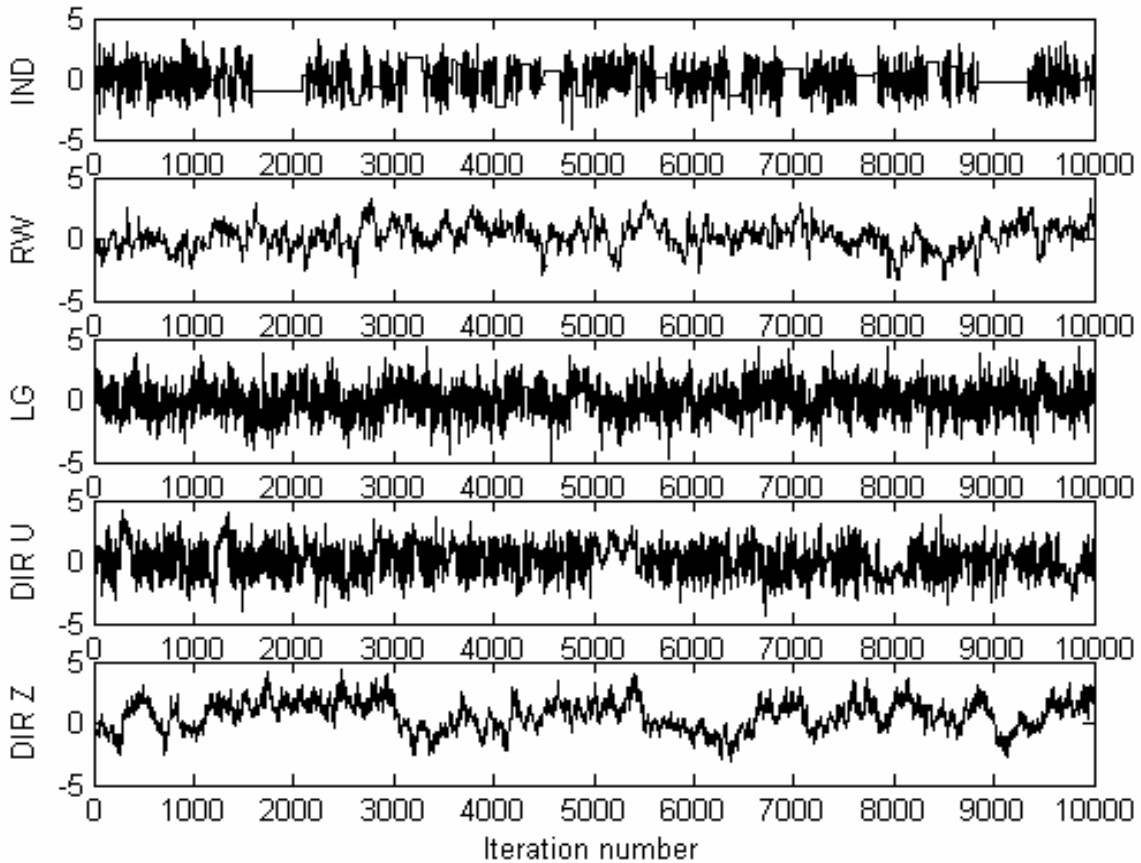
The number of evaluations (M) increases with the complexity of the target density per iteration (derivatives? or second derivatives?).
A measure of algorithm cpu time: M $\times$ IAC

Mixing and eigenvalues of the Markov chain

Theoretical results (Peskun, 1973, and others) show that mixing is governed by the second largest eigenvalue of the Markov chain (largest is 1).
It is difficult to assess these eigenvalues, but a rule of thumb is to push proposals away from the current state, while still maintaining reasonable acceptance rates.

Adaptive proposals

Independent proposal performs really well, when it does not get stuck!
RW moves slowly and surely, but not very far. Why not train proposal distribution from samples ?



Adaptive MH Gaussian density

From Haario et al. (2001).
When $b > b_0$,

$$q_b(\boldsymbol{x}^* | b x^b, \ldots, \boldsymbol{x}^0) = N(\boldsymbol{x}^b, \boldsymbol{C})$$

$$\boldsymbol{C} = \sigma^2 \text{cov}(\boldsymbol{x}^0, \ldots, \boldsymbol{x}^{b-1}) + \sigma^2 \epsilon \boldsymbol{I},$$

$$\text{cov}(\boldsymbol{x}^0, \ldots, \boldsymbol{x}^{b-1}) = \frac{1}{b-1} \sum_{c=0}^{b-1} (\boldsymbol{x}^c - \bar{\boldsymbol{x}})^t (\boldsymbol{x}^c - \bar{\boldsymbol{x}})$$

σ is a tuning parameter, that can depend on the dimension of $\boldsymbol{x}$.
ϵ ensures that there is too much adaptation.
Note : this is not a Markov chain! Asymptotically this still works (dependence is not too strong to make estimates have strange properties).