4F13: Machine Learning

http://learning.eng.cam.ac.uk/zoubin/ml06/

Department of Engineering, University of Cambridge
Michaelmas 2006

Lecture 9

Sampling and Markov Chain Monte Carlo (MCMC)

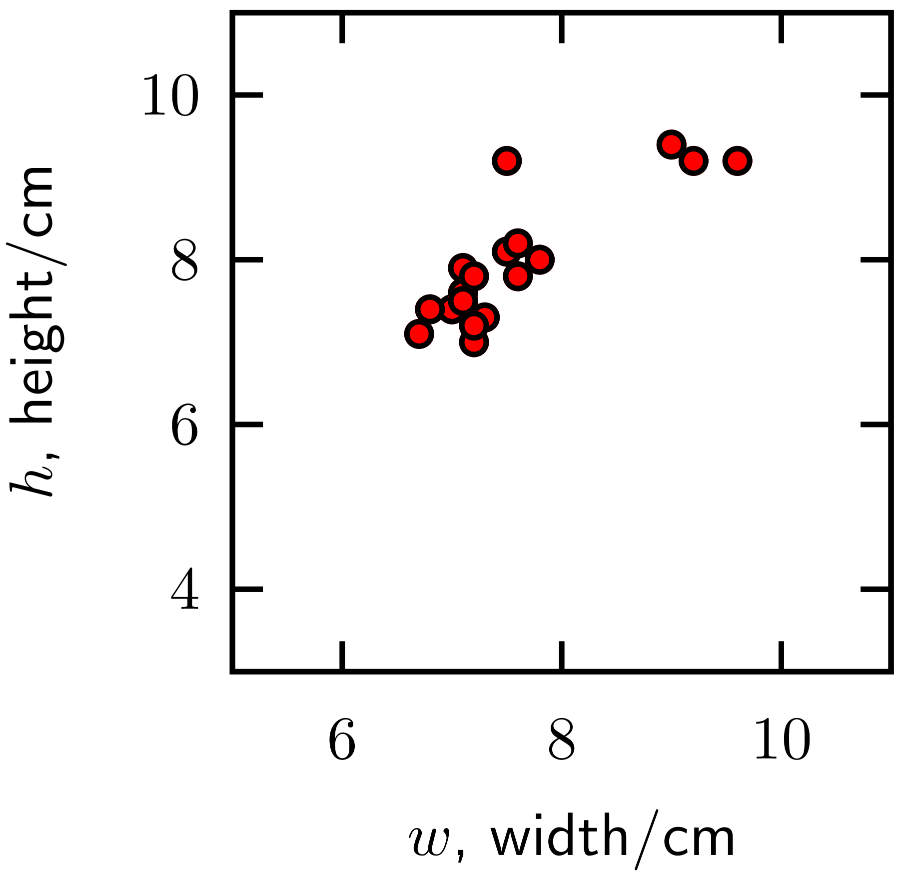
Iain Murray

i.murray+ta@gatsby.ucl.ac.uk

Last time

- **Monte Carlo, statistical sampling**
  - How to compute expectations by sampling
- **Rejection sampling**
  - How to sample fiddly distributions
    (for simulations, or if a method must use a certain distribution)
- **Importance sampling**
  - How to avoid sampling from fiddly distributions
    (like rejection, only works in low dimensions)

Importance sampling setup



$$p(h|w,\mathcal{D}) = \int p(h|w,\theta)p(\theta|\mathcal{D})\;\mathrm{d}\theta$$

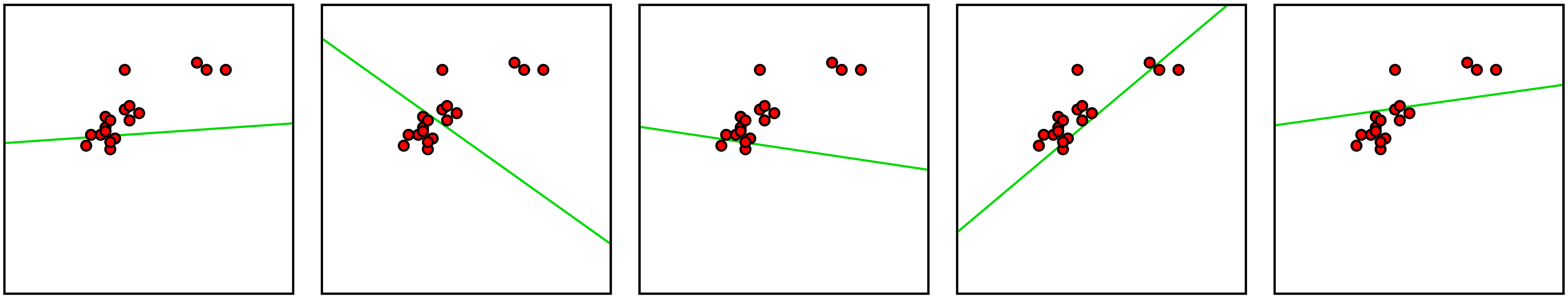
$$\approx \sum_s p(h|w,\theta^{(s)})\frac{w^{(s)}}{\sum_s' w^{(s')}}$$

$$w^{(s)} = \frac{P^*\big(\theta^{(s)}|\mathcal{D}\big)}{Q^*\big(\theta^{(s)}\big)}, \qquad \theta^{(s)} \sim Q$$

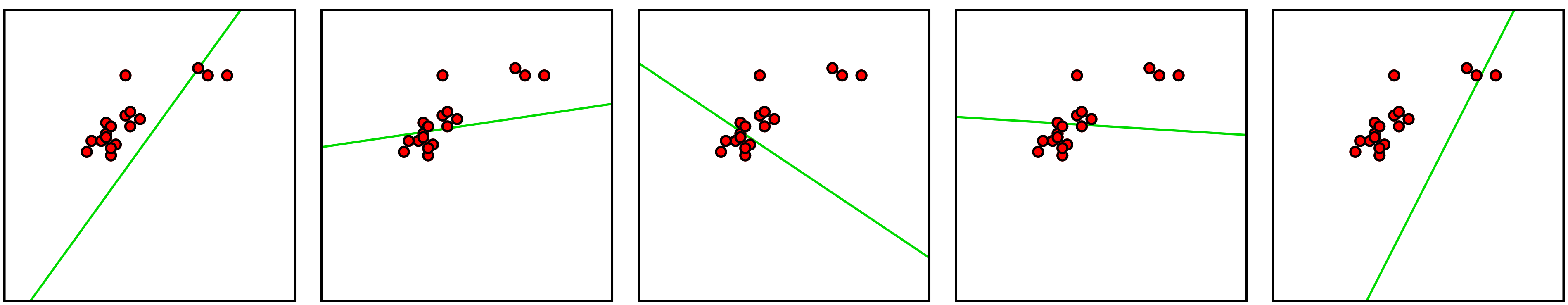
How to pick $Q(\theta)$?

$P(\theta|\mathcal{D}) \propto P^*(\theta|\mathcal{D}) = P(\mathcal{D}|\theta)P(\theta)$ — from Bayes' rule

Importance sampling weights



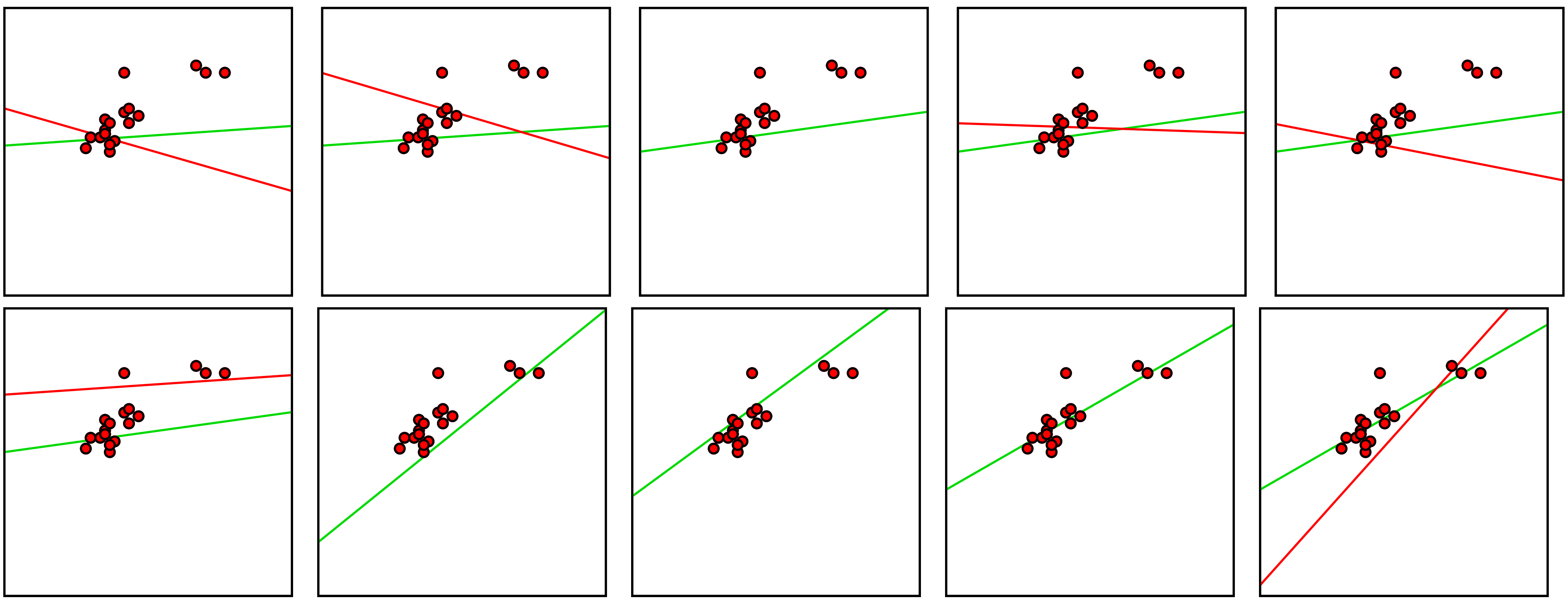
$w =$0.00548 $w =$1.59e-08 $w =$9.65e-06 $w =$0.371 $w =$0.103



$w =$1.01e-08 $w =$0.111 $w =$1.92e-09 $w =$0.0126 $w =$1.1e-51

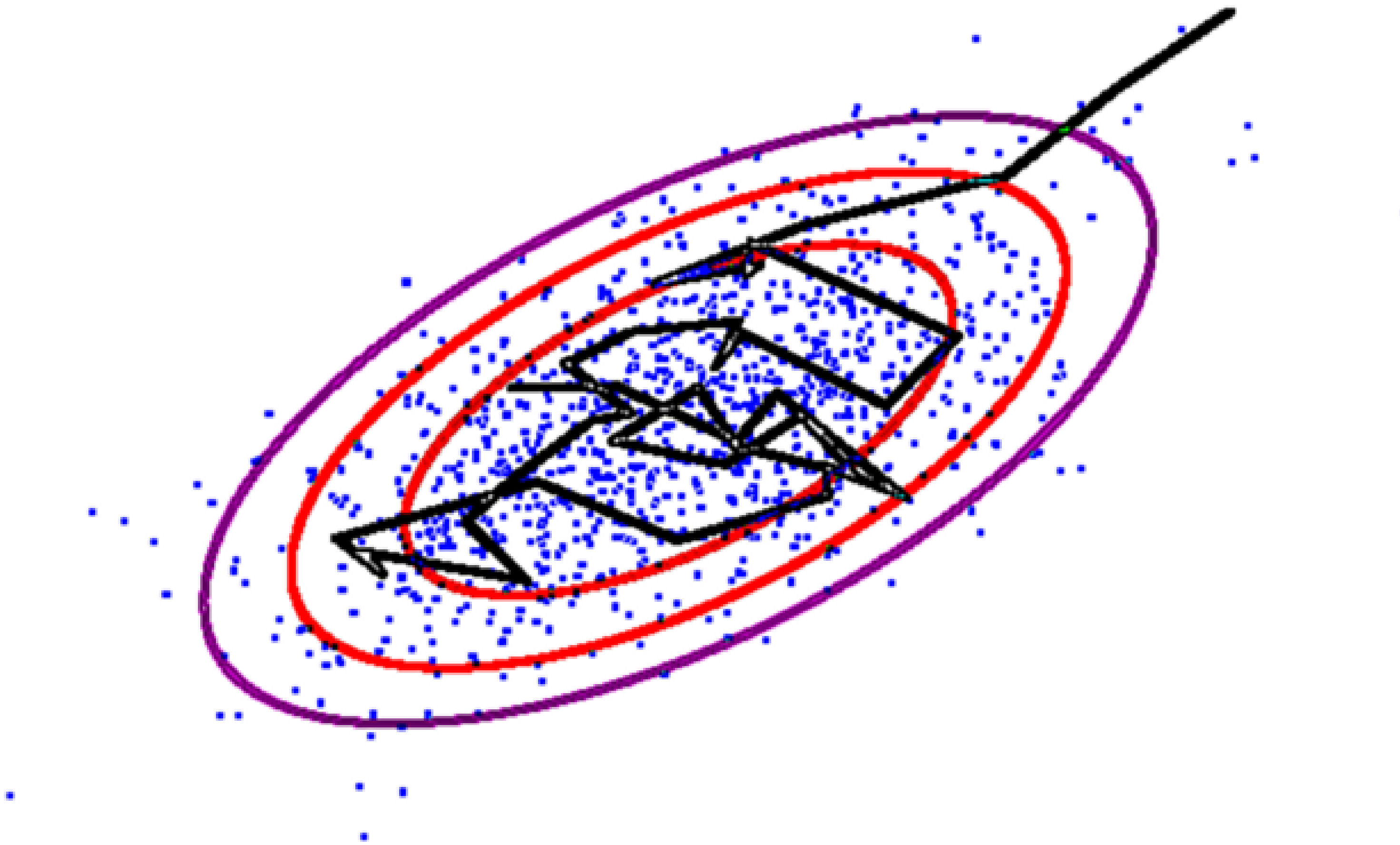
Metropolis–Hastings

- Propose a move from the current setting $Q(\theta'; \theta)$, e.g. $\mathcal{N}(\theta, \sigma^2)$
- Accept with probability $\min\left(1, \frac{P^*(\theta'|\mathcal{D})Q^*(\theta;\theta')}{Q^*(\theta';\theta)P^*(\theta|\mathcal{D})}\right)$
- Otherwise next setting is a copy of the previous parameters



Tending towards sampling from $p(\theta|\mathcal{D})$

In parameter space



Exploring a distribution by a random walk

Transition operators

$T(x' \leftarrow x) =$ **probability of moving from current state x to state x'**

(Discrete problems) probabilities can be stored in a matrix:

$$T = \begin{pmatrix} 2/3 & 1/2 & 1/2 \\ 1/6 & 0 & 1/2 \\ 1/6 & 1/2 & 0 \end{pmatrix} \qquad T_{ij} = T(x_i \leftarrow x_j)$$

T is an *operator* when applied to a probability vector (distribution)

$$\begin{pmatrix} 2/3 & 1/2 & 1/2 \\ 1/6 & 0 & 1/2 \\ 1/6 & 1/2 & 0 \end{pmatrix} \begin{pmatrix} 1/3 \\ 1/3 \\ 1/3 \end{pmatrix} = \begin{pmatrix} 5/9 \\ 2/9 \\ 2/9 \end{pmatrix}$$

Stationary distributions

$$P = \begin{pmatrix} 3/5 \\ 1/5 \\ 1/5 \end{pmatrix} \qquad TP = \begin{pmatrix} 2/3 & 1/2 & 1/2 \\ 1/6 & 0 & 1/2 \\ 1/6 & 1/2 & 0 \end{pmatrix} \begin{pmatrix} 3/5 \\ 1/5 \\ 1/5 \end{pmatrix} = \begin{pmatrix} 3/5 \\ 1/5 \\ 1/5 \end{pmatrix} = P$$

The probability of where you end up after many transitions is P...

$$\begin{pmatrix} 2/3 & 1/2 & 1/2 \\ 1/6 & 0 & 1/2 \\ 1/6 & 1/2 & 0 \end{pmatrix}^{100} \begin{pmatrix} 1/3 \\ 1/3 \\ 1/3 \end{pmatrix} = \begin{pmatrix} 3/5 \\ 1/5 \\ 1/5 \end{pmatrix} \quad \text{(to machine precision)}$$

... regardless of how you start

Markov chain Monte Carlo

Find a T such that

$$P(x') = \sum_x T(x' \leftarrow x) P(x)$$

P is a **stationary distribution** of T

Ensure $T^K(x' \leftarrow x) > 0$ for all $P(x') > 0$ so that:

- given sufficient time the starting location is forgotten
- the chain has a unique stationary distribution

Run a **Markov chain** (started arbitrarily)

$$x_1 \rightarrow x_2 \rightarrow x_3 \rightarrow \ldots \text{ where } x_t \sim T(x_t \leftarrow x_{t-1})$$

After a "burn-in" period every state is (approximately) drawn from P

Using these samples is **Markov chain Monte Carlo (MCMC)**

How do we find a T?

Detailed balance

Detailed balance means $\rightarrow x \rightarrow x'$ and $\rightarrow x' \rightarrow x$ are equally probable:

$$T(x' \leftarrow x) P(x) = T(x \leftarrow x') P(x')$$

"Like Bayes' rule", but don't write $T(x'|x)$; use $T(x'; x)$ or $T(x' \leftarrow x)$

Summing both sides over x:

$$\sum_x T(x' \leftarrow x) P(x) = P(x') \cancelto{1}{\sum_x T(x \leftarrow x')}$$

detailed balance implies a stationary condition

Enforcing detailed balance is easy: it only involves isolated pairs

Metropolis–Hastings

Transition operator

- Propose a move from the current state $Q(x';x)$, e.g. $\mathcal{N}(x, \sigma^2)$
- Accept with probability $\min\left(1,\ \frac{P(x')Q(x;x')}{P(x)Q(x';x)}\right)$
- Otherwise next state in chain is a copy of current state

Notes

- Can use P^* and Q^*; normalizers cancel in acceptance ratio
- Satisfies detailed balance (shown below)
- Q must be chosen to fulfill the other technical requirements

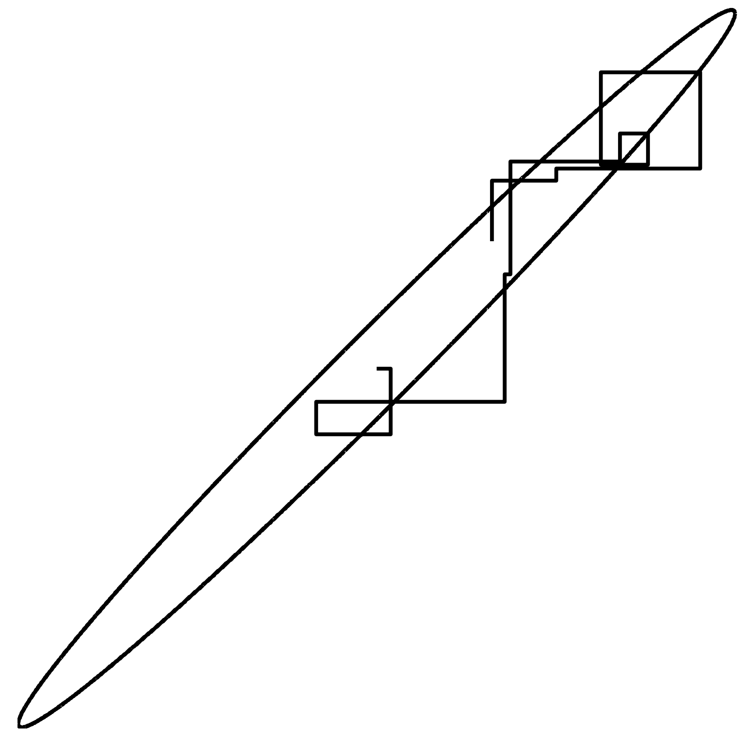
---

$$
\begin{aligned}
P(x) \cdot T(x' \leftarrow x) = P(x) \cdot Q(x';x) \min\left(1,\ \frac{P(x')Q(x;x')}{P(x)Q(x';x)}\right) &= \min\Big(P(x)Q(x';x),\ P(x')Q(x;x')\Big) \\
&= P(x') \cdot Q(x;x') \min\left(1,\ \frac{P(x)Q(x';x)}{P(x')Q(x;x')}\right) = P(x') \cdot T(x \leftarrow x')
\end{aligned}
$$

Gibbs sampling

A method with no rejections:

- Initialize $\mathbf{x}$ to some value
- For each variable in turn successively resample $P(x_i|\mathbf{x}_{j\neq i})$



Exercise: prove (when) Gibbs sampling is valid. Key points:

The Metropolis–Hastings accept prob. is 1 for 'proposal' $P(x_i|\mathbf{x}_{j\neq i})$

If two operators maintain a stationary distribution, applying both will still maintain the stationary distribution.

Routine Gibbs sampling

Gibbs sampling benefits from few free choices and convenient features of conditional distributions:

- Conditionals with a few discrete settings can be explicitly normalized:

$$P(x_i|\mathbf{x}_{j\neq i}) \propto P(x_i, \mathbf{x}_{j\neq i})$$
$$= \frac{P(x_i, \mathbf{x}_{j\neq i})}{\sum_{x_i'} P(x_i', \mathbf{x}_{j\neq i})} \leftarrow \text{this sum is small and easy}$$

- Continuous conditionals often turn out to be standard distributions.
- Otherwise rejection sampling is an option
  (although a simpler Metropolis scheme may be preferable)

`WinBUGS` and `OpenBUGS` sample graphical models using these tricks

Sampling summary

- Probabilistic modelling requires the computation of many sums and integrals
- Sampling requires insomnia or fast computers, but is highly competitive on the most complex problems
- Monte Carlo does not explicitly depend on dimension, although the global methods work only in low dimensions
- Markov chain Monte Carlo (MCMC) uses simple, local computations $\Rightarrow$ "easy" to implement.

Methods:

- Direct, rejection and importance sampling
- MCMC: Metropolis–Hastings, Gibbs sampling, . . .

Zoubin's next lecture is on alternative, deterministic algorithms