Part A Simulation and Statistical Programming HT15

Lecturer: Geoff Nicholls

University of Oxford

Lecture 14: MCMC, convergence; Implementing Bayesian inference using MCMC

Recall the Metropolis Hastings MCMC algorithm

MCMC targeting $p(x) = \tilde{p}(x)/Z_p$ using proposal $Y \sim q(y|x)$.

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

[1] Draw $y \sim q(\cdot|x)$ and $u \sim U[0,1]$.

[2] If

$$u \leq \alpha(y|x) \text{ where } \alpha(y|x) = \min\left\{1, \frac{\tilde{p}(y)q(x|y)}{\tilde{p}(x)q(y|x)}\right\}$$

set $X_{t+1} = y$, otherwise set $X_{t+1} = x$.

We initialise this with $X_0 = x_0, p(x_0) > 0$ and iterate for $t = 1, 2, 3, ...n$ to simulate the samples $X_0, X_1, X_2, ...X_n$ we need.

Recall the Ising model:

Denote by $\Omega = \{0,1\}^{n^2}$ the set of all binary images $X = (X_1, X_2, ..., X_{n^2})$, $X_i \in \{0,1\}$, where $i = 1, 2, ..., n^2$ is the cell index on the square lattice of image cells. Let $\#x$ give the number of disagreeing neighbors in the binary image X.

The *Ising model* is the following distribution over Ω:

$$\pi(x) = \exp(-\theta\#x)/Z.$$

Here θ is a *smoothing parameter* which is usually taken to be greater than zero and Z is a normalizing constant.

MCMC for the Ising Model

Recall the algorithm we wrote down earlier this week. Let $X^{(t)} = x$. $X^{(t+1)}$ is determined in the following way.

[1] Chose $i \sim U\{1, 2, ..., n^2\}$ and set $x' = x$ except $x'_i = 1 - x_i$.

[2] With probability $\alpha(x'|x)$ set $X^{(t+1)} = x'$ and otherwise set $X^{(t+1)} = x$.

Here $\alpha(x'|x)$ is

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

(refer R-file for implementation)

Remarks on implementation and monitoring MCMC

We work on a log scale if possible, to avoid overflow errors.

Worst

```
if (runif(1)<exp(-theta*hashXp)/exp(-theta*hashX) ) { ...
```

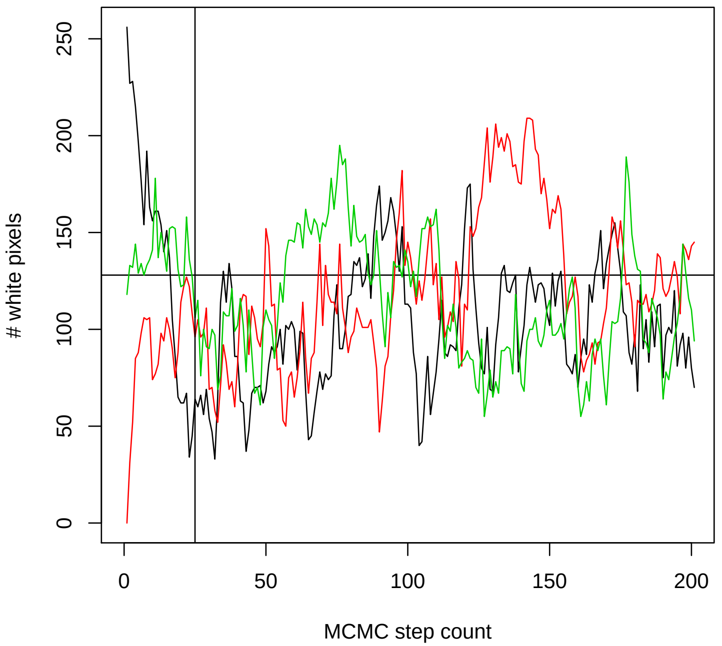
Better

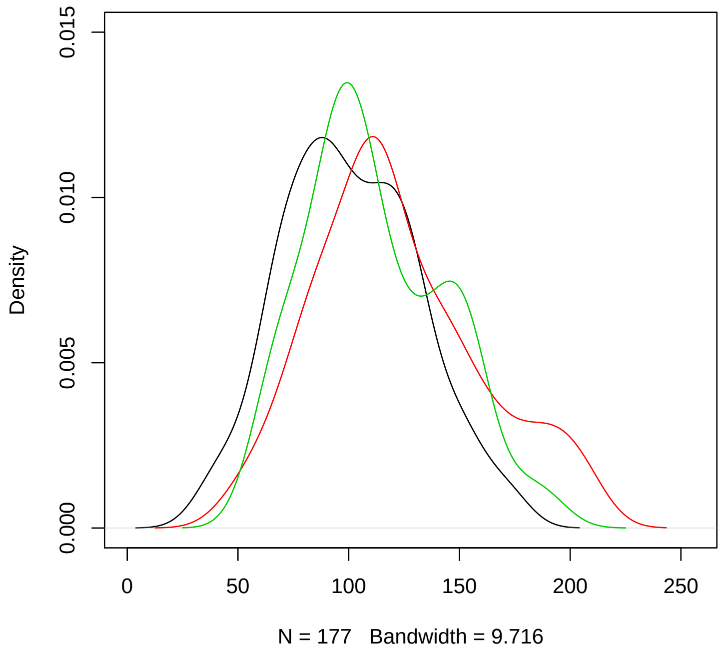
```
if (runif(1)<exp(theta*(hashX-hashXp)) ) { ...
```

Best

```
if (log(runif(1))<theta*(hashX-hashXp)) { ...
```

How do we monitor convergence for a multivariate problem like the Ising model? We monitor a few summary statistics - for example, $\#x$, or maybe $w(x) = \sum_i x_i$, the number of white pixels. We want to see this statistic converging to a stationary process. We repeat the run from different starting points and check we get essentially the same histogram of sampled values.





We usually sub-sample the output - this is just for practical reasons. The large densely sampled arrays don't add any interesting detail and are unwieldy to plot and compute on.

This code samples $w(x)$ every `SS` steps, and plots $w(x)$ and the current state if `show=TRUE`.

```
if (!(j%%SS)) {
  wp[j/SS+1]=sum(X)
  if (show) {
    par(mfrow=c(1,2));
    plot(wp,xlim=c(0,N/SS),ylim=c(0,n^2)); abline(h=n^2/2)
    image(X,col=gray(0:255/255),axes=F); box()
  }
}
```

Plotting is time consuming so if we just want to gather a sample of X and a $w(x)$-time series we switch it off.

Bayesian image recovery

Let X be an unknown true image. Suppose

$$X \sim \text{Ising}(\theta)$$

with θ known. The prior for X is $\pi(x) \propto \exp(-\theta \# x)$.

Suppose we observe X through a 'noisy channel'. At pixel $i = 1, 2, ..., n^2$ we observe

$$Y_i = X_i + \epsilon_i, \quad \text{with} \quad \epsilon_i \sim N(0, \sigma^2)$$

iid, and σ known. The likelihood for x_i is $L(x_i; y_i) = N(y_i; x_i, \sigma^2)$ so

$$L(x; y) \propto \prod_{i=1}^{n^2} \exp(-(x_i - y_i)^2 / 2\sigma^2).$$

If we observe $Y=y$ the probability that the unknown true image X equals x is

$$\begin{aligned}\pi(x|y) &\equiv \Pr(X=y|Y=y)\\ &\propto L(x;y)\pi(x)\\ &\propto \exp(-|x-y|^2/2\sigma^2)\exp(-\theta\#x)\end{aligned}$$

where $|x-y|^2=\sum_i(x_i-y_i)^2$.

We will simulate $X\sim\pi(x|y)$ and use the samples to estimate $E(X_i|Y=y)$ for each cell, $i=1,2,...,n^2$.

`(refer to R file for implementation)`

Modify our MCMC for $\pi(x)$ to target $\pi(x|y)$.

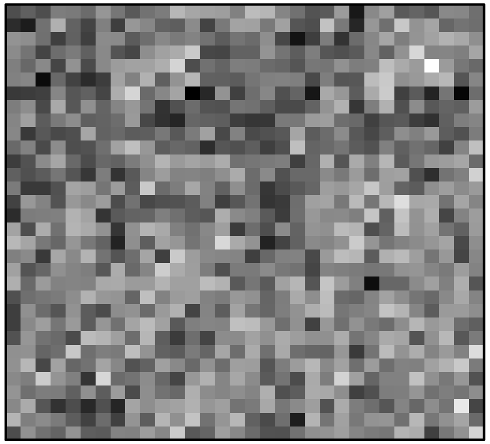
Essentially all we have to do is replace the acceptance probability in the algorithm targeting $\pi(x)$,

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

by the acceptance probability in the algorithm targeting $\pi(x|y)$,

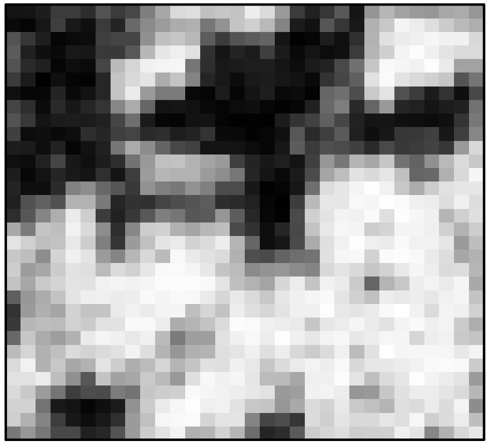
$$\begin{aligned}\alpha(x'|x) &= \min\left\{1,\ \frac{\pi(x'|y)q(x|x')}{\pi(x|y)q(x'|x)}\right\}\\ &= \min\Big\{1,\ \exp(-\theta(\#x' - \#x)\\ &\qquad\qquad \times \exp(-(|x' - y|^2 - |x - y|^2)/2\sigma^2)\Big\}\,.\end{aligned}$$

True 0/1



noisy sigma=1

mcmc sample X|Y



post mean