Simmulation

Hindermann & Snell pp 53-61

- How do we generate a MRF X ?

For a Markov Chain:

- generate in order using transition probability

- Does not work for MRF

Solution:

- Iteratively generate points and hope for steady state

Outline of approach

- Generate a Markov chain of images
- At each time n

$$p_{n+1}(y^{(n+1)}) = \sum_{y^{(n)}} p(y^{(n+1)}|y^{(n)}) p_n(y^{(n)})$$

- Assuming steady state

$$\lim_{n\to\infty} p_n(y) = p(y)$$

- $p(y)$ must be the solution to

$$p(y) = \sum_{y'} p(y|y') p(y')$$

Approach in book → Metropolis method
In class → Gibbs sampler (Geman - Geman)

1) Order the pixels from 1 to N
$S(0), \cdots, S(N-1)$

2) $$X_r^{(K+1)} = \begin{cases} X_r^{(K)} & \text{if } r \neq S(K \bmod N) \\ W_r & \text{if } r = S(K \bmod N) \end{cases}$$

Generate W with distribution
$$p(X_s \mid X_i \; i \neq s)$$
For Ising $= \dfrac{1}{1 + \exp\{-2\beta(V(x_s, x_{\partial s}) - 2)\}}$

$X^{(0)} \to X^{(1)} \to X^{(2)} \to \cdots$

Notice:
1) $X^{(K)}$ is a Markov Chain in K
2) $X^{(K)}$ is nonhomogeneous.

Define $Y^{(m)} = X^{(mN)}$ $N = \#$ of pixels

Then

1) $Y^{(m)}$ is a Markov Chain
   - Any subsample MC is a Markov Chain

2) $Y^{(m)}$ is homogeneous
   - Each update replaces all pixels

$$p(y^{(m+1)} | y^{(m)})$$

3) $Y^{(m)}$ is irreducible.

That means that $\forall\, y^{(m+1)}$ and $y^{(m)}$

$$p(y^{(m+1)} | y^{(m)}) > 0$$

We can get between any two states

Lecture 17

4) Must show that

$$p(y) = \sum_{y'} p(y|y')\, p(y')$$

where

$$p(y) = \frac{1}{Z} \exp\left\{ -\sum_{c \in \mathcal{C}} V_c(y) \right\}$$

$$= g(y)$$

Gibbs distribution

• It is enough to show that

$$g(y) = \sum_{y'} p_s(y|y')\, g(y')$$

where $p_s(y|y')$ is the replacement of a single pixel.

$$p_s(y|y') = g_s(y_s | y'_{\partial s})\, \delta(y_i - y'_i \ \ i \neq s)$$

$$g(y') = g_s(y'_s | y'_{\partial s})\, g(y'_i \ \ i \neq s)$$

$$g(y) = \sum_{y'} p_s(y|y')\, g(y')$$

$$= \sum_{y'_s} \sum_{y'_i\ i \neq s} g_s(y_s|y'_{\partial s})\, \delta(y_i - y'_i\ i\neq s)\, g_s(y'_s|y'_{\partial s}) \cdot g(y'_i\ i \neq s)$$

$$= \sum_{y'_s} g_s(y_s|y_{\partial s})\, g_s(y'_s|y_{\partial s})\, g(y_i\ i\neq s)$$

$$= g_s(y_s|y_{\partial s})\, g(y_i\ i \neq s)$$

$$= g(y)$$

Theorem: 1) + 2) + 3) + 4) $\Rightarrow$ For any initial y:

$$\lim_{m\to\infty} p_m(y) = g(y)$$

- From result of 4)

$$\lim_{k\to\infty} p_k(x) = g(x)$$

Metropolis Sampler

It may be difficult to generate a RV with distribution

$$X_s \sim p(x_s | X_{\partial s})$$

In this case, the Gibbs sampler is not practical.

Alternatively, we may use a Metropolis Sampler

Objective: Generate a random (field) X with distribution

$$p(x) = \frac{1}{Z} e^{-u(x)}$$

Select a sampling function with two properties

1) $$q(x, x') = q(x', x)$$
which is the probability of selecting x' given an initial state x

so $$\sum_{x'} q(x, x') = 1$$

2) The MC generated by $q(x_n, x_{n+1})$ is irreducible i.e. all states may be reached

Metropolis Algorithm

1) Randomly generate a new state x' given the current state x by using the distribution

$$q(x, x')$$

2) If $u(x') \leq u(x)$

$$\text{set } x \leftarrow x'$$

If $u(x') > u(x)$

$$\text{compute } \alpha = \exp\{-(u(x') - u(x))\}$$

set $x \leftarrow x'$ with probability α

set $x \leftarrow x$ with probability $1-\alpha$

3) Repeat process starting at step 1)

Theorem: The metropolis algorithm converges to a ergodic distribution of

$$\lim_{n \to \infty} p_n(x) = \frac{1}{z} e^{-u(x)}$$

proof!

1) MC is irreducible

2) MC is homogeneous

3) need to show that

$$\sum_i \pi_i P_{ij} = \pi_i$$

where

$$\pi_i = \frac{1}{z} e^{-u(x=i)}$$

$$P_{ij} = \underbrace{p(X_{n+1}=j \mid X_n = i)}_{\text{Metropolis algorithm.}}$$

· To do this, we will need the concept of a reversible MC

Definition: A MC is reversible iff

$$\begin{cases} p(X_n = i, X_{n+1} = j) = p(X_n = j, X_{n+1} = i) \\ \text{and it is stationary} \end{cases}$$

$\Leftrightarrow \quad p(X_n=i)\,p(X_{n+1}=j \mid X_n=i)$
$\qquad = p(X_n=j)\,p(X_{n+1}=i \mid X_n=j)$

$$\Leftrightarrow \quad \underbrace{\pi_i P_{ij} = \pi_j P_{ji}}_{\text{known as detailed Balance equations}}$$

Reversible $\Leftrightarrow \begin{Bmatrix} \text{stationary} \\ + \\ \pi_i P_{ij} = \pi_j P_{ji} \end{Bmatrix}$

Notice that

$$\pi_i P_{ij} = \pi_j P_{ji}$$

$$\Rightarrow \sum_i \pi_i P_{ij} = \pi_j \sum_i P_{ji} = \pi_j$$

$$\underbrace{\sum_i \pi_i P_{ij} = \pi_j}_{\text{Full balance equations}}$$

Conclusion: If we find a solution to the detailed balance equations, then π_i is an ergodic distribution for the MC.

Consider two states X and X'

Without loss of generallity assume $U(X) \geq U(X')$

Then the detailed balance equations are:

$$\underbrace{\tfrac{1}{2} e^{-U(x)}}_{\pi_i} \underbrace{q(x,x') \cdot 1}_{P_{ij}} \overset{?}{=} \underbrace{\tfrac{1}{2} e^{-U(x')}}_{\pi_j} \underbrace{q(x',x)\, \alpha}_{P_{ji}}$$

where $\alpha = e^{-(U(x) - U(x'))}$

so
$$e^{-U(x)} \overset{?}{=} e^{-U(x')} e^{-(U(x) - U(x'))}$$

$$e^{-U(x)} = e^{-U(x)}$$

↑ yes! <u>QED</u>

Hastings - Metropolis Sampler

Can we use Metropolis algorithm when
$$q(x,x') \neq q(x',x) \;?$$

Yes, if we choose an a acceptance probability of

$$\alpha(x,x') = \min\left\{1,\ \frac{\pi(x')\,q(x',x)}{\pi(x)\ \ q(x,x')}\right\}$$

Notice that $\alpha(x,x')$ compensates for bias in $q(x,x')$

Special Cases:

1) $q(x',x) = q(x,x')$

$$\alpha(x,x') = \min\left\{1,\ \frac{\pi(x')}{\pi(x)}\right\}$$

$$= \begin{cases} 1 & \text{if} \quad \pi(x') \leq \pi(x) \\ \dfrac{\pi(x')}{\pi(x)} & \text{if} \quad \pi(x') > \pi(x) \end{cases}$$

for $\pi(x) = \frac{1}{2} e^{-u(x)}$

$$= \begin{cases} 1 & \text{if} \quad u(x') \leq u(x) \\ e^{-(u(x')-u(x))} & \text{if} \quad u(x') > u(x) \end{cases}$$

$\Rightarrow$ Metropolis sampler

2) $q(x, x') = p(x_s | x_{\partial s})\, \delta(x'_r - x_r \text{ for } r \neq s)$

$$\frac{q(x', x)}{q(x, x')} = \frac{p(x_s | x_{\partial s})}{p(x'_s | x_{\partial s})}$$

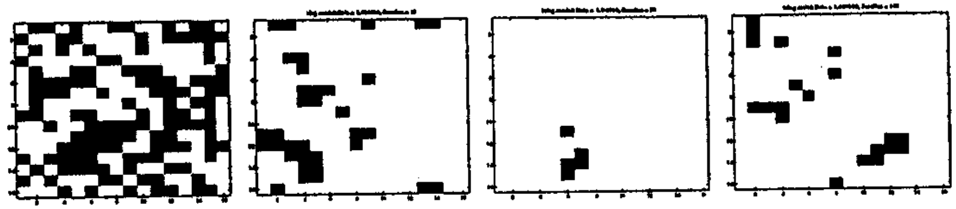
$$\frac{\pi(x')}{\pi(x)} = \frac{p(x'_s | x_{\partial s})}{p(x_s | x_{\partial s})}$$

$$\frac{\pi(x')\, q(x', x)}{\pi(x)\, q(x, x')} = 1 \Rightarrow \alpha(x, x') = 1$$

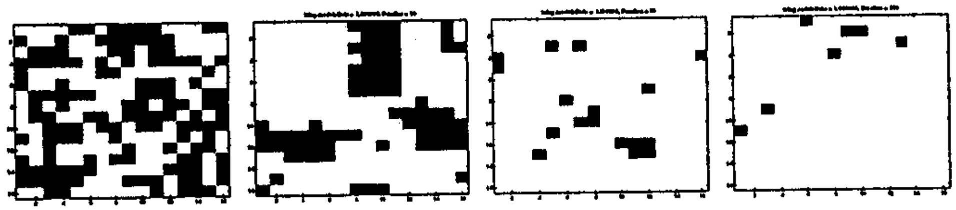
$\Rightarrow$ Gibbs sampler

Example Simulation for Ising Model($\beta = 1.0$)

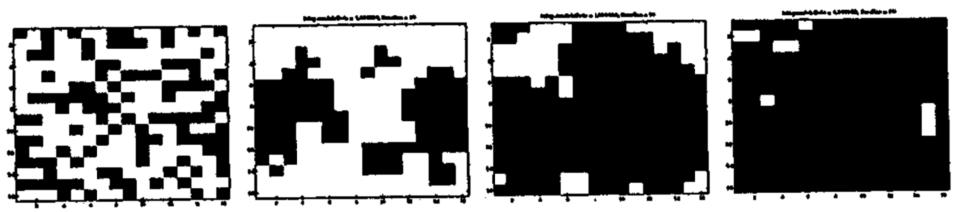
- Test 1



- Test 2



- Test 3



- Test 3