Ising Model Simulation

Gibbs Sampling, Metropolis-Hastings, and Convergence Analysis

AUTHOR
Your Name

PUBLISHED
December 18, 2024

Introduction

This report explores the 2D Ising model simulation using two algorithms: - Gibbs Sampling - Metropolis-Hastings

Key aspects: 1. Simulating spin configurations under different temperatures. 2. Examining the effects of boundary conditions (Periodic and Fixed). 3. Monitoring convergence using Monte Carlo Standard Errors (MCSE).

Setup

Functions for Simulation

Initialize Lattice

```r
initialize_lattice <- function(L) {
  matrix(sample(c(-1, 1), L * L, replace = TRUE), L, L)
}
```

Neighbor Computation

```r
compute_neighbors_sum <- function(lattice, i, j) {
  L <- nrow(lattice)
  lattice[(i %% L) + 1, j] +
    lattice[(i - 2) %% L + 1, j] +
    lattice[i, (j %% L) + 1] +
    lattice[i, (j - 2) %% L + 1]
}
```

Gibbs Sampling Update

```r
gibbs_step <- function(lattice, beta) {
  L <- nrow(lattice)
  for (i in 1:L) {
    for (j in 1:L) {
      neighbors_sum <- compute_neighbors_sum(lattice, i, j)
      prob <- 1 / (1 + exp(-2 * beta * neighbors_sum))
      lattice[i, j] <- ifelse(runif(1) < prob, 1, -1)
    }
  }
  return(lattice)
}
```

Metropolis-Hastings Update

```
metropolis_step <- function(lattice, beta) {
  L <- nrow(lattice)
  for (k in 1:(L * L)) {
    i <- sample(1:L, 1)
    j <- sample(1:L, 1)
    spin <- lattice[i, j]
    neighbors_sum <- compute_neighbors_sum(lattice, i, j)
    delta_E <- 2 * spin * neighbors_sum
    if (delta_E <= 0 || runif(1) < exp(-beta * delta_E)) {
      lattice[i, j] <- -spin
    }
  }
  return(lattice)
}
```

Boundary Conditions

Fixed Boundary Lattice Initialization

```
initialize_fixed_boundary <- function(L, boundary_spin = 1) {
  lattice <- matrix(boundary_spin, L, L)
  lattice[2:(L - 1), 2:(L - 1)] <- matrix(sample(c(-1, 1), (L - 2) * (L - 2), replace = TRUE),
  return(lattice)
}
```

Boundary Condition Simulation

```
simulate_boundary_conditions <- function(L, steps, T, condition = "Periodic") {
  beta <- 1 / T
  if (condition == "Fixed") {
    lattice <- initialize_fixed_boundary(L)
  } else {
    lattice <- initialize_lattice(L)
  }

  magnetizations <- numeric(steps)
  for (t in 1:steps) {
    lattice <- if (condition == "Fixed") metropolis_step(lattice, beta) else gibbs_step(lattic
    magnetizations[t] <- mean(lattice)
  }
  list(lattice = lattice, magnetizations = magnetizations)
}
```

Monte Carlo Convergence Monitoring

```
monitor_convergence <- function(magnetizations) {
  n <- length(magnetizations)
  means <- cumsum(magnetizations) / (1:n)
  mcse <- sqrt(cumsum((magnetizations - means)^2) / (1:n))
  data.frame(Iteration = 1:n, Mean = means, MCSE = mcse)
}
```

Simulation and Results

Parameters

```
L <- 32
steps <- 100
temperatures <- c(1.5, 2.5, 3.0, 5.0)
```

Simulating with Gibbs Sampling

```
results_gibbs <- list()
for (T in temperatures) {
  beta <- 1 / T
  lattice <- initialize_lattice(L)
  magnetizations <- numeric(steps)
  for (step in 1:steps) {
    lattice <- gibbs_step(lattice, beta)
    magnetizations[step] <- mean(lattice)
  }
  results_gibbs[[as.character(T)]] <- list(lattice = lattice, magnetizations = magnetizations)
}
```

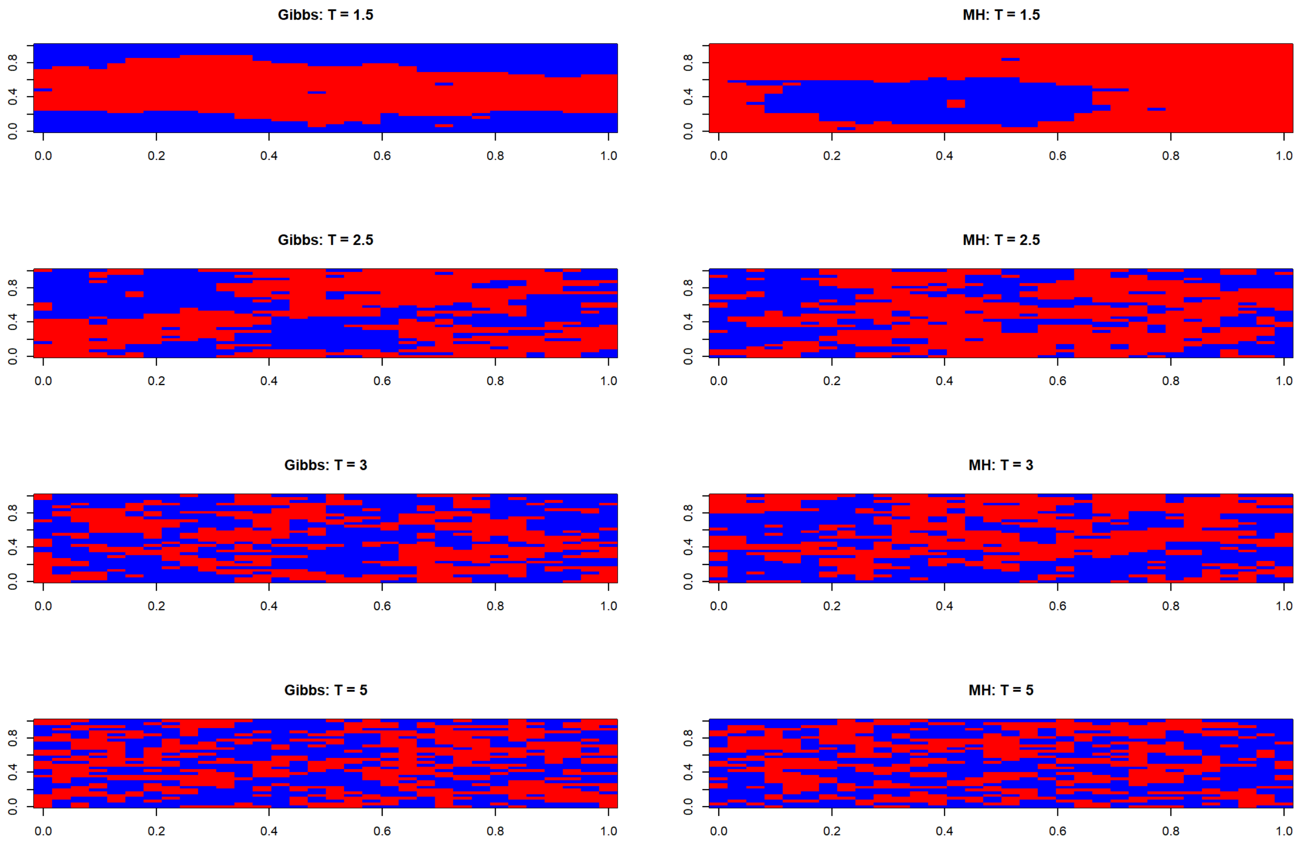
Simulating with Metropolis-Hastings

```
results_mh <- list()
for (T in temperatures) {
  beta <- 1 / T
  lattice <- initialize_lattice(L)
  magnetizations <- numeric(steps)
  for (step in 1:steps) {
    lattice <- metropolis_step(lattice, beta)
    magnetizations[step] <- mean(lattice)
  }
  results_mh[[as.character(T)]] <- list(lattice = lattice, magnetizations = magnetizations)
}
```

Visualizing Results

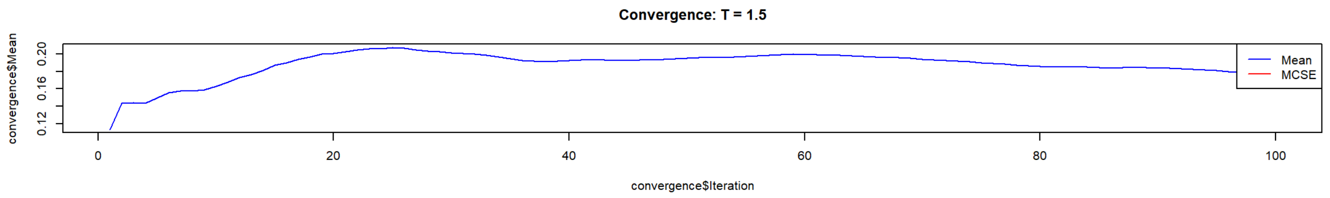
Lattice Configurations

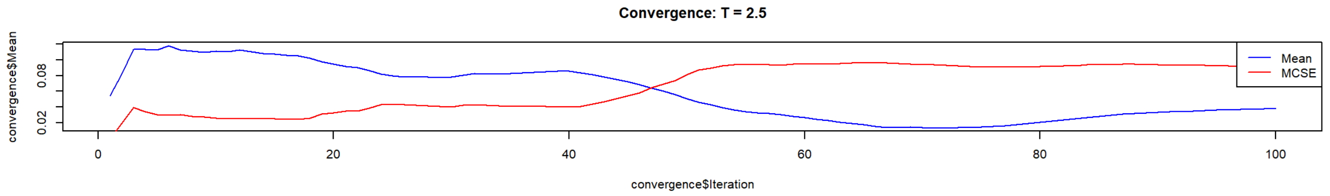
```
par(mfrow = c(length(temperatures), 2))
for (T in temperatures) {
  lattice_gibbs <- results_gibbs[[as.character(T)]]$lattice
  lattice_mh <- results_mh[[as.character(T)]]$lattice
  image(t(apply(lattice_gibbs, 2, rev)), col = c("blue", "red"), main = paste("Gibbs: T =", T)
  image(t(apply(lattice_mh, 2, rev)), col = c("blue", "red"), main = paste("MH: T =", T))
}
```

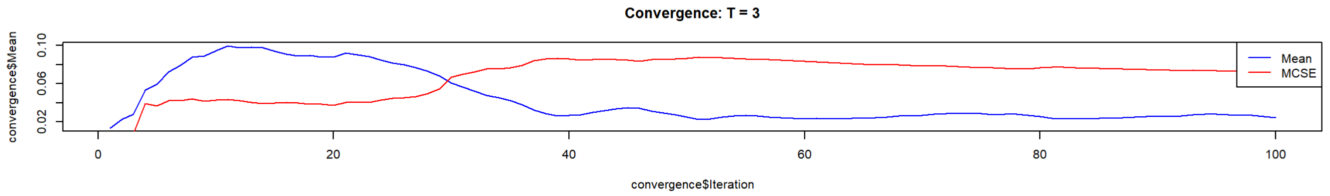


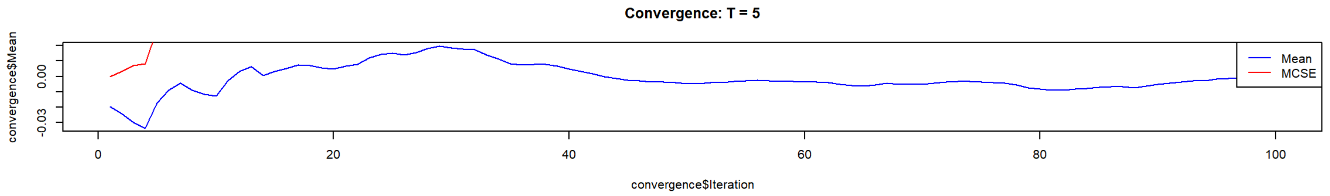
Convergence Monitoring

```
par(mfrow = c(length(temperatures), 1))
for (T in temperatures) {
  magnetizations <- results_gibbs[[as.character(T)]]$magnetizations
  convergence <- monitor_convergence(magnetizations)
  plot(convergence$Iteration, convergence$Mean, type = "l", col = "blue", main = paste("Conver
  lines(convergence$Iteration, convergence$MCSE, col = "red")
  legend("topright", legend = c("Mean", "MCSE"), col = c("blue", "red"), lty = 1)
}
```









Observations

1. At low T, spins align into ordered clusters.
2. At high T, spins become more random, reflecting thermal agitation.
3. Gibbs Sampling and Metropolis-Hastings yield similar results but converge differently.