MCMC Diagnostics and Models - Detailed Hand Calculations

Part 1: Bimodal Distribution Example

1.1 Purpose and Problem

Purpose: Test MCMC convergence diagnostics on a challenging distribution with multiple modes.

Problem:

- Many real-world distributions have multiple peaks (e.g., mixture models, multimodal posteriors)
- MCMC can get stuck in one mode and fail to explore the full distribution
- Need to diagnose whether chains have converged and explored all modes

Model:

```
p(x) = 0.5 × N(x|-3, 4) + 0.5 × N(x|1, 2)
```

This represents:

- 50% of data comes from N(-3, 4) (mean=-3, variance=4)
- 50% of data comes from N(1, 2) (mean=1, variance=2)

1.2 Prior and Likelihood

For sampling from this distribution directly:

- **No explicit prior** (we're sampling from the target distribution itself)
- **Target density**: $p(x) = 0.5 \times N(x|-3, 4) + 0.5 \times N(x|1, 2)$

1.3 Hand Calculation Example

Step 1: Compute theoretical moments

```
E[X] = 0.5 × E[X₁] + 0.5 × E[X₂]
     = 0.5 × (-3) + 0.5 × 1
     = -1.5 + 0.5
     = -1

E[X²] = 0.5 × E[X₁²] + 0.5 × E[X₂²]
      = 0.5 × (Var[X₁] + E[X₁]²) + 0.5 × (Var[X₂] + E[X₂]²)
      = 0.5 × (4 + 9) + 0.5 × (2 + 1)
      = 0.5 × 13 + 0.5 × 3
      = 6.5 + 1.5
      = 8

Var[X] = E[X²] - (E[X])²
       = 8 - (-1)²
       = 8 - 1
       = 7
```

Step 2: Metropolis sampling (small example)

Initial: $x_0 = 0$ Proposal variance: $\tau = 0.5$

Iteration 1:

```
1. Propose: x' = x₀ + τ × ε = 0 + 0.5 × 0.8 = 0.4
2. Compute acceptance ratio:
   α = p(x')/p(x₀)

   p(0) = 0.5 × N(0|-3,4) + 0.5 × N(0|1,2)
        = 0.5 × 0.0001 + 0.5 × 0.2197
        = 0.1099

   p(0.4) = 0.5 × N(0.4|-3,4) + 0.5 × N(0.4|1,2)
          = 0.5 × 0.0003 + 0.5 × 0.2689
          = 0.1346

   α = 0.1346/0.1099 = 1.225 > 1

3. Accept: x₁ = 0.4
```

Iteration 2:

```
1. Propose: x' = 0.4 + 0.5 × (-1.2) = -0.2
2. Compute acceptance ratio:
   p(-0.2) = 0.5 × 0.00005 + 0.5 × 0.1942 = 0.0971
   α = 0.0971/0.1346 = 0.722

3. Generate u ~ U(0,1) = 0.3
   Since 0.3 < 0.722, accept: x₂ = -0.2
```

Step 3: Multiple chains example

Chain 1: [0.4, -0.2, -0.5, 0.8, 1.2] Chain 2: [-3.1, -2.8, -3.2, -2.9, -3.0]

Notice Chain 2 is stuck in the left mode!

Step 4: Compute R̂-hat

```
M = 2 chains, S = 5 samples

Chain means:
θ̄₁ = (0.4 - 0.2 - 0.5 + 0.8 + 1.2)/5 = 0.34
θ̄₂ = (-3.1 - 2.8 - 3.2 - 2.9 - 3.0)/5 = -3.0
θ̄ = (0.34 + (-3.0))/2 = -1.33

Within-chain variances:
s₁² = [(0.4-0.34)² + (-0.2-0.34)² + ...]/4 = 0.533
s₂² = [(-3.1-(-3.0))² + ...]/4 = 0.025
W = (0.533 + 0.025)/2 = 0.279

Between-chain variance:
B = 5/(2-1) × [(0.34-(-1.33))² + (-3.0-(-1.33))²]
  = 5 × [2.79 + 2.79]
  = 5 × 5.58
  = 27.9

R̂² = (S-1)/S + (1/S)(B/W)
    = 4/5 + (1/5)(27.9/0.279)
    = 0.8 + 0.2 × 100
    = 20.8

R̂ = √20.8 = 4.56
```

This high $\hat{R}$ indicates poor convergence!

Part 2: Change Point Detection Model

2.1 Purpose and Problem

Purpose: Detect when a significant change occurred in a time series of count data.

Problem:

- Accident rates may change due to safety regulations
- Need to identify when the change occurred
- Must estimate rates before and after the change

Example: Coal mining accidents from 1851-1962

- Early period: higher accident rate
- Later period: lower rate (due to safety improvements)
- When did the change occur?

2.2 Model Definition

Data: $x = [x_1, x_2, ..., x_n]$ (accident counts per year)

Parameters:

- c: change point (year when rate changed)
- λ_1: accident rate before change point
- λ_2: accident rate after change point

Likelihood:

```
xᵢ ~ Poisson(λ₁) if i ≤ c
xᵢ ~ Poisson(λ₂) if i > c
```

Priors:

```
c ~ Uniform(1, N)
λ₁ ~ Gamma(α, β)
λ₂ ~ Gamma(α, β)
```

2.3 Small Hand Calculation Example

Data: x = [3, 2, 1, 5, 6, 7] (N = 6 years) **Hyperparameters**: $\alpha = 1$, $\beta = 1$

2.4 Gibbs Sampling Steps

Initial values: $c^{(0)} = 3$, $\lambda_1^{(0)} = 2$, $\lambda_2^{(0)} = 5$

Iteration 1:

Step 1: Sample λ_1 | x, c, λ_2

Posterior for λ_1:

```
p(λ₁ | x, c, λ₂) = Gamma(α + Σᵢ₌₁ᶜ xᵢ, β + c)

With c = 3:
Sum of data before change point: 3 + 2 + 1 = 6
α' = 1 + 6 = 7
β' = 1 + 3 = 4

λ₁⁽¹⁾ ~ Gamma(7, 4)
```

Let's say we sample $\lambda_1^{(1)} = 1.8$

Step 2: Sample λ_2 | x, c, λ_1

```
p(λ₂ | x, c, λ₁) = Gamma(α + Σᵢ₌ᶜ₊₁ᴺ xᵢ, β + N - c)

With c = 3:
Sum of data after change point: 5 + 6 + 7 = 18
α' = 1 + 18 = 19
β' = 1 + (6-3) = 4

λ₂⁽¹⁾ ~ Gamma(19, 4)
```

Let's say we sample $\lambda_2^{(1)} = 4.7$

Step 3: Sample c | x, λ_1, λ_2

For each possible c, compute:

```
log p(c=k | x, λ₁, λ₂) ∝ Σᵢ₌₁ᵏ xᵢ log(λ₁) - k×λ₁ + Σᵢ₌ₖ₊₁ᴺ xᵢ log(λ₂) - (N-k)×λ₂
```

With $\lambda_1 = 1.8$, $\lambda_2 = 4.7$:

c = 1:

```
log p ∝ 3×log(1.8) - 1×1.8 + (2+1+5+6+7)×log(4.7) - 5×4.7
      = 3×0.588 - 1.8 + 21×1.548 - 23.5
      = 1.764 - 1.8 + 32.508 - 23.5
      = 8.972
```

c = 2:

```
log p ∝ (3+2)×log(1.8) - 2×1.8 + (1+5+6+7)×log(4.7) - 4×4.7
      = 5×0.588 - 3.6 + 19×1.548 - 18.8
      = 2.94 - 3.6 + 29.412 - 18.8
      = 9.952
```

c = 3:

```
log p ∝ (3+2+1)×log(1.8) - 3×1.8 + (5+6+7)×log(4.7) - 3×4.7
      = 6×0.588 - 5.4 + 18×1.548 - 14.1
      = 3.528 - 5.4 + 27.864 - 14.1
      = 11.892
```

c = 4:

```
log p ∝ (3+2+1+5)×log(1.8) - 4×1.8 + (6+7)×log(4.7) - 2×4.7
      = 11×0.588 - 7.2 + 13×1.548 - 9.4
      = 6.468 - 7.2 + 20.124 - 9.4
      = 9.992
```

c = 5:

```
log p ∝ (3+2+1+5+6)×log(1.8) - 5×1.8 + 7×log(4.7) - 1×4.7
      = 17×0.588 - 9.0 + 7×1.548 - 4.7
      = 9.996 - 9.0 + 10.836 - 4.7
      = 7.132
```

Normalize to get probabilities:

```
log p = [8.972, 9.952, 11.892, 9.992, 7.132]
p ∝ exp(log p - max(log p))
  = exp([8.972-11.892, 9.952-11.892, 0, 9.992-11.892, 7.132-11.892])
  = exp([-2.92, -1.94, 0, -1.90, -4.76])
  = [0.054, 0.144, 1.000, 0.149, 0.009]

Normalized:
p = [0.040, 0.106, 0.736, 0.110, 0.007]
```

Sample $c^{(1)}$ according to these probabilities. Most likely $c^{(1)} = 3$.

2.5 Computing Posterior Mean

After running the Gibbs sampler for 1000 iterations (after warmup):

```
Samples: {c⁽ᵢ⁾, λ₁⁽ᵢ⁾, λ₂⁽ᵢ⁾} for i = 1, ..., 1000

Posterior means:
E[c|x] ≈ (1/1000) Σᵢ c⁽ᵢ⁾ = 3.2
E[λ₁|x] ≈ (1/1000) Σᵢ λ₁⁽ᵢ⁾ = 1.9
E[λ₂|x] ≈ (1/1000) Σᵢ λ₂⁽ᵢ⁾ = 6.1
```

2.6 Posterior Predictive Distribution

To predict accident count for year 7:

For each posterior sample $(c^{(i)}, \lambda_1^{(i)}, \lambda_2^{(i)})$:

```
if 7 ≤ c⁽ᵢ⁾:
    x₇⁽ᵢ⁾ ~ Poisson(λ₁⁽ᵢ⁾)
else:
    x₇⁽ᵢ⁾ ~ Poisson(λ₂⁽ᵢ⁾)
```

Example samples:

```
Sample 1: c=3, λ₁=1.8, λ₂=4.7
Since 7 > 3: x₇ ~ Poisson(4.7) → x₇ = 5

Sample 2: c=4, λ₁=2.1, λ₂=5.2
Since 7 > 4: x₇ ~ Poisson(5.2) → x₇ = 4

...
```

Posterior predictive mean:

```
E[x₇|x] ≈ (1/1000) Σᵢ x₇⁽ⁱ⁾ = 5.8
```

2.7 Making Decisions

Question: Has the accident rate decreased?

Compute: $P(\lambda_1 > \lambda_2 \mid x)$

```
Count samples where λ₁⁽ⁱ⁾ > λ₂⁽ⁱ⁾
P(λ₁ > λ₂ | x) ≈ 0.02
```

This indicates strong evidence that $\lambda_2 > \lambda_1$ (accident rate increased).

Part 3: MCMC Diagnostics Summary

3.1 When to Use $\hat{R}$

- Run $M \geq 4$ chains with different starting points
- Compute $\hat{R}$ for each parameter
- $\hat{R} \approx 1$: Good convergence
- $\hat{R} > 1.1$: Poor convergence, need more iterations

3.2 When to Use ESS

- After confirming convergence with $\hat{R}$
- ESS tells you the "effective" number of independent samples
- Use for computing Monte Carlo Standard Error (MCSE)
- $MCSE = SD/\sqrt{ESS}$

3.3 Practical Workflow

1. Run multiple chains with overdispersed starting points
2. Discard warmup period (typically 50%)
3. Compute $\hat{R}$ for each parameter
4. If $\hat{R} < 1.1$, compute ESS
5. Use merged chains for inference
6. Report uncertainty using MCSE

This detailed walkthrough shows how MCMC methods solve real problems and how to diagnose their convergence properly.