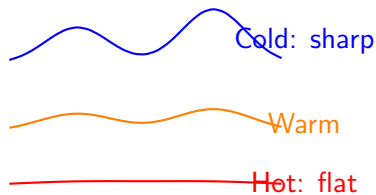
Parallel Tempering MCMC

The Problem and Solution

The Problem:

- Standard MCMC gets stuck in local modes
- Can't explore separated peaks
- Dilemma: small steps (stuck) vs. large steps (rejected)



The Solution: Temperature Ladder

- Run N chains targeting π^{γ_n}
- $0 < \gamma_1 < \cdots < \gamma_N = 1$
- Hot ($\gamma \approx 0$): Explores freely
- Cold ($\gamma = 1$): Exploits peaks

Key Insight Different temperatures see the same distribution differently - hot chains explore, cold chains exploit

The Temperature Mechanism

Key Idea: Tempered Distributions

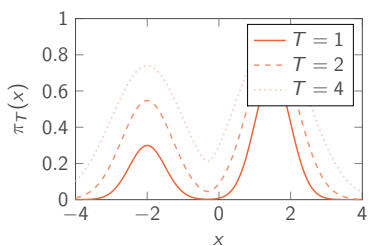
Define a family of distributions indexed by inverse temperature $0 < \gamma_1 < \gamma_2 < \ldots < \gamma_N = 1$:

$$\pi_{\gamma_n}(x) \propto \pi(x)^{\gamma_n}$$

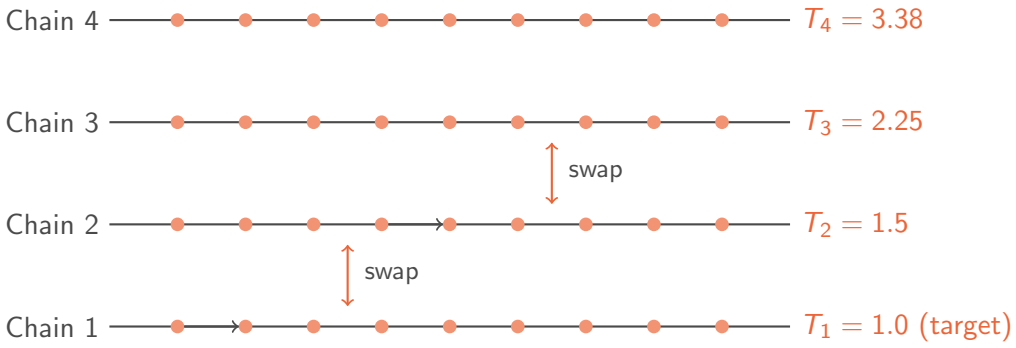
where $n = 1, \ldots, N$ and $\pi(x)$ is our target distribution.

Properties:

- $\gamma_N = 1$: Original target distribution
- $\gamma_N \approx 0$: Uniform, explores broadly



Parallel Chains Architecture



Two Types of Moves

1. **Within-chain updates**: Standard MCMC at each temperature
2. **Between-chain swaps**: Exchange states between adjacent temperatures

Parallel Tempering Algorithm Prerquisites

- **Target Distribution**: $\pi(x)$
- **Proposal Distribution**: For each tempered chain $q(x'|x)$ - could potentially depend on temperature
- **Initialization**: $x_n^{(0)}$ for $n = 1, \ldots, N$
- **Standard MCMC Step**: Any MCMC kernel (e.g., RWM, MALA)
- **Number of Chains**: N
- **Number of Samples per Chain**: T
- **Temperature Schedule**: $\{\gamma_n\}_{n=1}^N$ with $\gamma_N = 1$

Parallel Tempering Algorithm

Algorithm 1 Parallel Tempering MCMC

for $t = 1$ **to** T **do**
  for all $n \in \{1, \dots, N\}$ **in parallel do**
    Sample $x_n^{(t)}$ using a standard MCMC step targeting π^{γ_n}
  end for
  $k \sim \text{Uniform}\{1, \dots, N-1\}$
  $\alpha_{\text{swap}} = \min\left\{1, \left(\frac{\pi(x_{k+1}^{(t)})}{\pi(x_k^{(t)})}\right)^{\gamma_k - \gamma_{k+1}}\right\}$
  Swap $(x_k^{(t)}, x_{k+1}^{(t)})$ with probability α_{swap}
end for
return $\{x_N^{(t)}\}_{t=1}^{T}$

Swap Acceptance Ratio Derivation

Propose swap: Exchange states $x_{k_1} \leftrightarrow x_{k_2}$ between chains k_1 and k_2

$$
\begin{aligned}
\alpha_{\text{swap}} = \frac{\text{Prob of proposed state}}{\text{Prob of current state}} &= \min\left\{1, \frac{\pi^{\gamma_{k_1}}(x_{k_2}) \cdot \pi^{\gamma_{k_2}}(x_{k_1})}{\pi^{\gamma_{k_1}}(x_{k_1}) \cdot \pi^{\gamma_{k_2}}(x_{k_2})}\right\} \\
&= \min\left\{1, \frac{[\pi(x_{k_2})]^{\gamma_{k_1}}}{[\pi(x_{k_2})]^{\gamma_{k_2}}} \cdot \frac{[\pi(x_{k_1})]^{\gamma_{k_2}}}{[\pi(x_{k_1})]^{\gamma_{k_1}}}\right\} \\
&= \min\left\{1, [\pi(x_{k_2})]^{\gamma_{k_1}-\gamma_{k_2}} \cdot [\pi(x_{k_1})]^{\gamma_{k_2}-\gamma_{k_1}}\right\} \\
&= \min\left\{1, \frac{[\pi(x_{k_2})]^{\gamma_{k_1}-\gamma_{k_2}}}{[\pi(x_{k_1})]^{\gamma_{k_1}-\gamma_{k_2}}}\right\} = \min\left\{1, \left(\frac{\pi(x_{k_2})}{\pi(x_{k_1})}\right)^{\gamma_{k_1}-\gamma_{k_2}}\right\}
\end{aligned}
$$

Parallel Tempering: Swap Move Acceptance

Metropolis-Hastings Derivation

Joint target: $\pi^{\gamma_1} \otimes \pi^{\gamma_2} \otimes \cdots \otimes \pi^{\gamma_N}$ where γ_i (inverse temperature)
MH acceptance ratio:

$$\alpha = \frac{\text{Probability of proposed state}}{\text{Probability of current state}} = \frac{\pi^{\gamma_{k_1}}(x_{k_2})\pi^{\gamma_{k_2}}(x_{k_1})}{\pi^{\gamma_{k_1}}(x_{k_1})\pi^{\gamma_{k_2}}(x_{k_2})}$$

Accept with probability: $\min(1, \alpha)$
Swapping state of two chains doesn't change the joint target distribution. This ensures detailed balance w.r.t. the joint distribution!

Parallel Tempering MCMC: Normalization Requirements

Target Distribution

NOT required to be normalized

Tempered Distributions

Also NOT normalized

Why It Works: Acceptance Ratios

Within-chain moves:

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

Normalizing constants cancel!

Between-chain swaps:

$$\alpha = \min\left(1, \frac{\pi_i(\mathbf{x}_j)\pi_j(\mathbf{x}_i)}{\pi_i(\mathbf{x}_i)\pi_j(\mathbf{x}_j)}\right)$$

Constants cancel again!

Why Swapping Works

The Relay Race Mechanism:

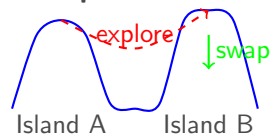
1. Hot chain randomly discovers new mode
2. Swap propagates discovery downward
3. Cold chain thoroughly explores it
4. Information flows both ways

Swap Acceptance (adjacent chains):

$$\alpha = \min\left\{1, \left(\frac{\pi(x_{k+1})}{\pi(x_k)}\right)^{\gamma_k - \gamma_{k+1}}\right\}$$

- Favors moving high-prob states to cold
- Favors moving low-prob states to hot
- Adjacent swaps $\rightarrow$ high acceptance

Example: Two Islands



Without PT: Stuck on Island A forever
With PT: Both islands sampled!

Detailed Balance and Ergodicity

Proposition (Detailed Balance)

The parallel tempering algorithm satisfies detailed balance with respect to the joint distribution:

$$\pi(x_1, \ldots, x_K) = \prod_{i=1}^{K} \frac{1}{Z_i} \pi(x_i)^{1/T_i}$$

Proof Sketch:

1. Within-chain moves: Standard MCMC detailed balance
2. Swap moves: Show $\pi(\mathbf{x})P(\mathbf{x} \to \mathbf{x}') = \pi(\mathbf{x}')P(\mathbf{x}' \to \mathbf{x})$
3. Symmetry of proposal + Metropolis ratio ensures balance