PML 3. Markov chain Monte Carlo

Probabilistic Machine Learning Reading Group

Miguel Santos

December 3, 2025

Institute of Mathematical Sciences (ICMAT-CSIC)

Contents

Contents

The Bayesian Inference Problem

- We model data $\mathcal{D}$ and latent variables x through a joint $p(x, \mathcal{D}) = p(\mathcal{D} \mid x)p(x)$.
- The goal is to infer the posterior:

$$p(x \mid \mathcal{D}) = \frac{p(\mathcal{D} \mid x)p(x)}{p(\mathcal{D})}.$$

- Examples:
  - **Regression:** predict outcomes with uncertainty intervals.
  - **Clustering:** infer mixture components and their probabilities.
  - **Neural networks:** estimate uncertainty in model parameters.

Why Exact Inference is Hard

- The evidence

$$p(x) = \int p(x, \mathcal{D})\, dz$$

  is rarely tractable.
- No analytical solution for $p(\mathcal{D} \mid x)$

Previous chapter: VI

- Introduce **tractable distribution** $q(z)$ to **approximate** the true posterior.
- Turn inference into **optimization**: $\arg\max_{q \in \mathcal{Q}} \mathcal{L}(q)$
- $\mathcal{L}$ measures how similar p and q are.

MCMC vs Variational Inference

VI

- **Approximation based.**
- Accuracy depends on the selected family.
- Fast.
- Scales with SGD.

MCMC

- **Sample based.**
- Asymptotically exact.
- Slow for high dimension.
- Hard to scale. (Not impossible!!!)

Contents

Basis of MCMC

- **Monte Carlo:** random sampling, usually used to estimate expectations of the form

$$\mathbb{E}_\pi[f(X)] \approx \frac{1}{N}\sum_{i=1}^{N} f(X_i),$$

where $X_i \sim \pi$ i.i.d.

Example: estimating π

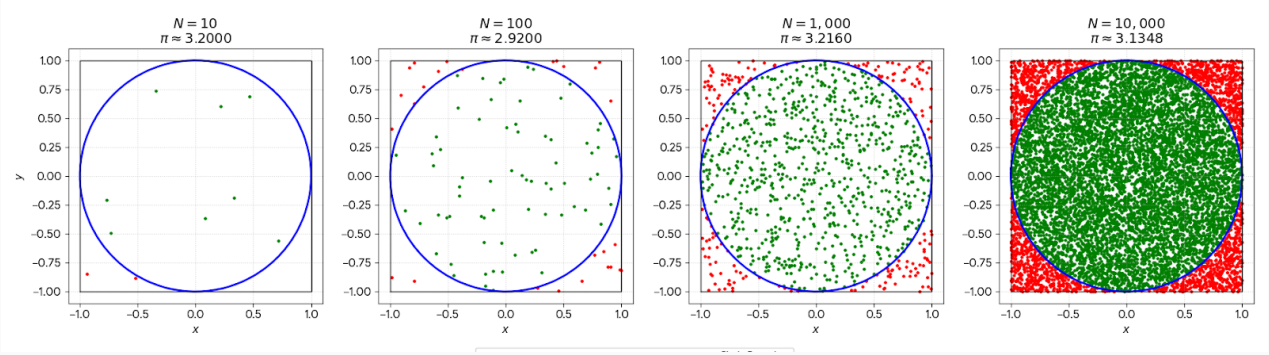


Figure 1: Monte Carlo simulations for Estimating π

Basis of MCMC

- **Markov Chain:**
  - Sequence $(X_0, X_1, X_2, \dots)$ with

$$\mathbb{P}(X_{t+1} \in A \mid X_t, X_{t-1}, \dots) = \mathbb{P}(X_{t+1} \in A \mid X_t).$$

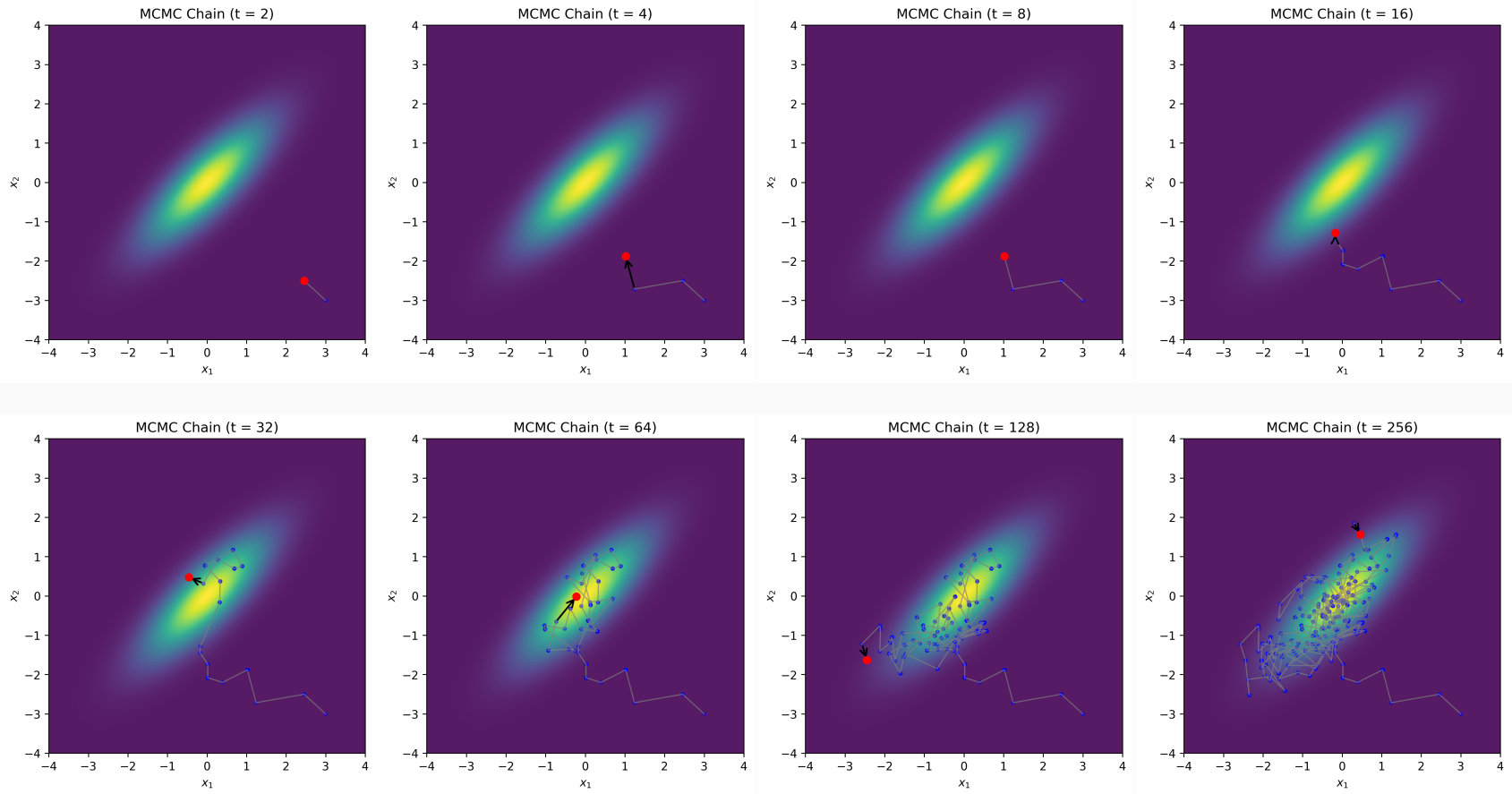
    .
  - **Stationary distribution:** the distribution of X_t does not change over time.

Basis of MCMC

Markov Chain Monte Carlo (MCMC):

- **Objective:** build a sample $\{x_0, x_1, \ldots, x_N\}$ of $p(x)$
- **Random sampling** sequentially $x_0 \to x_1 \to x_2 \to \ldots$.
- $p(x)$ is an **stationary distribution** of the built Markov Chain.
- "The time spend in each state x^* is proportional to the objective distribution $p(x^*)$".

Basis of MCMC



[Code S1]

Contents

Contents

Elements

- Proposal distribution / transition kernel q:

$$x_n \to x_{n+1}, \qquad x_{n+1} \sim q(x_{n+1} \mid x_n)$$

Example: $x_{n+1} = x_n + \varepsilon$, with $\varepsilon \sim \mathcal{N}(0, \sigma^2 I_d)$, so that $q(x_{n+1} \mid x_n) = \mathcal{N}(x_n, \sigma^2 I_d)$.

- Acceptance probability:

$$A = \min \left\{ 1, \underbrace{\frac{p(x_{n+1})}{p(x_n)}}_{\text{target density ratio}} \cdot \underbrace{\frac{q(x_n \mid x_{n+1})}{q(x_{n+1} \mid x_n)}}_{\text{proposal correction}} \right\}$$

[GIF]

Kernel examples

Examples of proposal q

- Random walk proposal: $x' = x + \varepsilon$, with $\varepsilon \sim \mathcal{N}(0, \sigma^2 I_d)$, so that $q(x' \mid x) = \mathcal{N}(x'; x, \sigma^2 I_d)$.
- Independent proposal (importance–sampling style): $q(x' \mid x) = q(x')$.
- Mixture proposal: $q(x' \mid x) = \sum_k w_k \, q_k(x' \mid x)$, with $w_k \geq 0$ and $\sum_k w_k = 1$.
- Data-driven proposal: $q(x' \mid x, \mathcal{D})$, where $\mathcal{D}$ denotes the data.
- Adaptive MCMC: $q_t(x' \mid x) = \mathcal{N}(x'; x, \tau(t) I_d)$ with $\tau(t) = \tau_0 \left(1 + \frac{1}{t+1}\right)$, where t is the iteration index.

Proposal distributions for MH

Conditions on the proposal q

- *Support containment:*

$$\operatorname{supp}(p) \subseteq \operatorname{supp}\left(q(\cdot \mid x)\right) \quad \forall x,$$

  i.e. any point where $p(x) > 0$ must be reachable with $q(x' \mid x) > 0$.

- **Robust behaviour**: q should not be too concentrated to allow for exploration or too much expanded for convergence.

[Code S2]

Random Walk MH: Initialization

Where does the MCMC chain start?

- **Burn-in**, run several samples at the beginning that are not store as objective sample for approximation.
- For gradient based methods, do not start in modes as $\nabla \log p(x) = 0$.
- To reduce dependence on the initial state, run **several chains in parallel** from different starting points.

Contents

Gibbs sampler: basic idea

For multivariate distributions.

Idea: update one coordinate at a time.

Example in 3D Target density: $p(x_1, x_2, x_3)$, from current sample point, (x_1, x_2, x_3)

$$
\begin{aligned}
x_1^{(t+1)} &\sim p(x_1 \mid x_2^{(t)}, x_3^{(t)}) \\
x_2^{(t+1)} &\sim p(x_2 \mid x_1^{(t+1)}, x_3^{(t)}) \\
x_3^{(t+1)} &\sim p(x_3 \mid x_1^{(t+1)}, x_2^{(t+1)}).
\end{aligned}
$$

In general. For d-dimensional $x = (x_1, \ldots, x_d)$, one Gibbs sweep is

$$
x_i^{(t+1)} \sim p(x_i \mid x_1^{(t+1)}, \ldots, x_{i-1}^{(t+1)}, x_{i+1}^{(t)}, \ldots, x_d^{(t)}), \quad i = 1, \ldots, d,
$$

[Code S3]

Gibbs Sampling: Comments

- When **full conditional distributions** are not available in closed form, Gibbs can be **combined with Metropolis–Hastings**.

$$x_i^{(t+1)} \sim p\big(x_i \mid x_1^{(t+1)}, \ldots, x_{i-1}^{(t+1)}, x_{i+1}^{(t)}, \ldots, x_d^{(t)}\big), \quad i = 1, \ldots, d,$$

Gibbs Sampling: Comments

- **Blocked Gibbs**: update **grouped variables**.

  In a GMM, update the means latent variables at once.

- **Collapsed Gibbs**: analytically integrate out some parameters so that fewer variables are sampled.

  In a GMM we can integrate out the component means and sample only the latent labels $z = 1, 2, \ldots, K$. In each step we also recompute the target statistics.

Contents

Auxiliary variable MCMC: basic ideas

Core Idea: Simplification via Expansion

- Introduce an **auxiliary variable** $\mathbf{u}$, defining a new joint distribution $p(\mathbf{x}, \mathbf{u})$ such that:
  1. In the end we compute $p(x) = \sum_u p(\mathbf{x}, \mathbf{u})$.
  2. The joint distribution is known.
  3. The conditional distributions $p(\mathbf{x}|\mathbf{u})$ and $p(\mathbf{u}|\mathbf{x})$ are **easy to sample** from.

Example: Slice sampling

- **Auxiliary Variable $\mathbf{u}$**: The auxiliary variable $\mathbf{u}$ is sampled from a uniform distribution.
- **Why it works?**

$$\hat{p}(x, v) = \begin{cases} \dfrac{1}{Z_p}, & 0 \leq v \leq \tilde{p}(x), \\ 0, & \text{otherwise}, \end{cases}$$

$$\int \hat{p}(x, v)\, dv = \int_0^{\tilde{p}(x)} \frac{1}{Z_p}\, dv = \frac{\tilde{p}(x)}{Z_p} = p(x).$$

- **The Sampling Gibbs Cycle:**
  1. **Vertical Step (Sample $\mathbf{u}$):** Sample $u^{(t+1)}$ uniformly from the interval $[0, p(\mathbf{x}^{(t)})]$.
  2. **Horizontal Step (Sample $\mathbf{x}$):** Sample $\mathbf{x}^{(t+1)}$ uniformly from the "slice" or region $S = \{\mathbf{x} : u^{(t+1)} < p(\mathbf{x})\}$.

Contents

Hamiltonian Monte Carlo: basic idea

- Designed for **high-dimensional** distributions.
- Introduce an **auxiliary momentum** variable $v \sim \mathcal{N}(0, \Sigma)$ to define the **Hamiltonian**

$$H(x, v) = E(x) + K(v) = E(x) + \tfrac{1}{2} v^\top \Sigma^{-1} v,$$

  where $E(x) = -\log p(x)$ is the **potential energy**.
- **Hamiltonian dynamics:**

$$\frac{dx}{dt} = \Sigma^{-1} v, \qquad \frac{dv}{dt} = -\nabla E(x).$$

HMC visualization

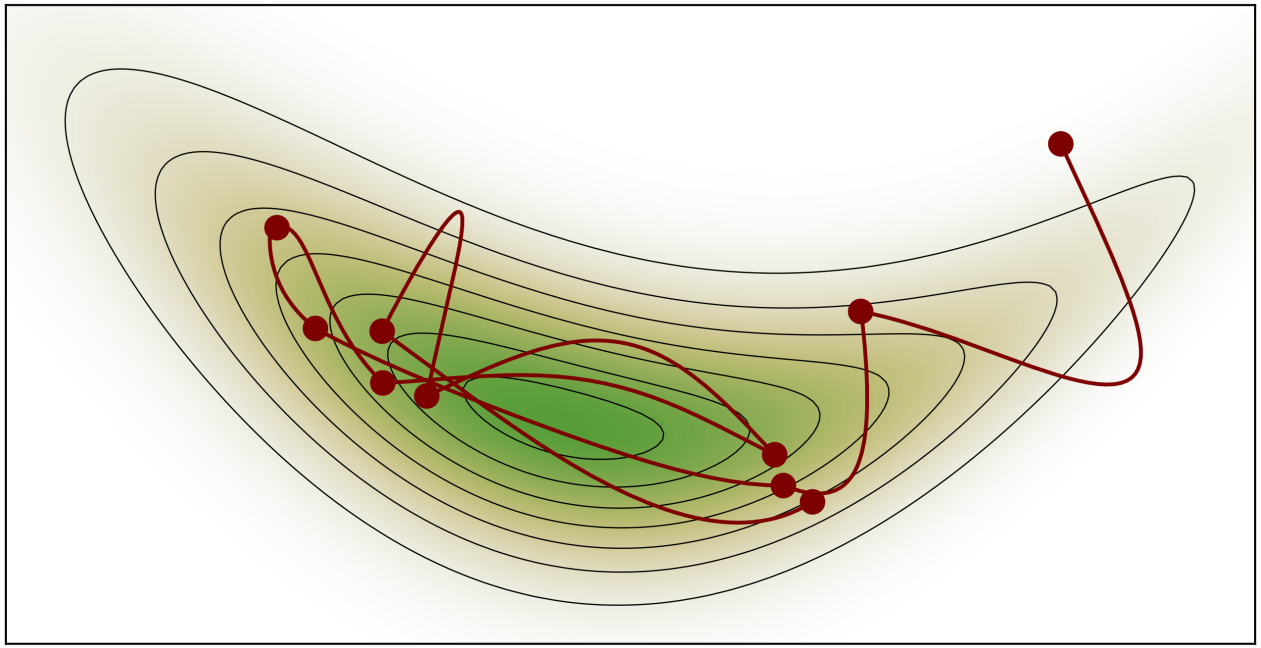


Figure 2: HMC visualization

Numerical approximation: leapfrog integrator

- Given **step size** $\eta > 0$, **mass matrix** Σ, and L **number of updates** the leapfrog updates are:

$$
\begin{aligned}
v_{l+\frac{1}{2}} &= v_l - \frac{\eta}{2} \nabla E(x_l), \\
x_{l+1} &= x_l + \eta \, \Sigma^{-1} v_{l+\frac{1}{2}}, \\
v_{l+1} &= v_{l+\frac{1}{2}} - \frac{\eta}{2} \nabla E(x_{l+1}).
\end{aligned}
$$

- Metropolis **accept/reject** based on the change in H.

$$
A = \min \left\{ 1, \exp \left[-H(x^*, v^*) + H(x_{t-1}, v_{t-1}) \right] \right\}.
$$

HMC Algorithm

Algorithm 1 Hamiltonian Monte Carlo

Input: η, L, Σ, number of samples N
Initialize x_0
for $t = 1$ **to** N **do**
    Sample new momentum $v_{t-1} \sim \mathcal{N}(0, \Sigma)$
    Set $(x'_0, v'_0) \leftarrow (x_{t-1}, v_{t-1})$
    for $\ell = 1$ **to** L **do**
        Leapfrog Update.
    end for
    Proposal: $(x^*, v^*) \leftarrow (x'_L, v'_L)$
    Accept/reject probability:
    $\alpha \leftarrow \min\left\{1, \exp\left[-H(x^*, v^*) + H(x_{t-1}, v_{t-1})\right]\right\}$
end for

HMC Extension: NUTS

No-U-Turn Sampler (NUTS) **Motivation:** HMC requires of the number of leapfrog steps L, the step size η, and the mass matrix Σ.

- **Choosing L:** builds trajectories forward and backward in time and stops automatically when the path *starts to turn back* on itself (a "no U-turn" condition).
- **Choosing η:** The step size is adapted during a **burn-in**.
- **Choosing Σ:** The mass matrix is estimated during **burn-in**,

$$\Sigma = \mathbb{E}\left[(\mathbf{X} - \overline{\mathbf{X}})(\mathbf{X} - \overline{\mathbf{X}})^T\right]$$

[Code S5]

HMC Extension: Riemann Manifold HMC

Idea: Replace the fixed mass matrix Σ with a **position-dependent metric** $G(x)$, so that HMC moves on a *Riemannian manifold* adapted to the **local geometry** of the target.

i. **Hessian**

$$\Sigma(x) = \nabla^2 E(x).$$

ii. **Fisher information matrix:**

$$\Sigma(x) = \mathcal{I}(x) = -\,\mathbb{E}_{p(x|\mathcal{D})}\left[\nabla_x^2 \log p(x \mid \mathcal{D})\right].$$

HMC Extensions: Stochastic Gradient HMC

- Langevin dynamics ($L = 1$): special case of HMC,

$$x_{t+1} = x_t - \eta \nabla E(x_t) + \sqrt{2\eta}\, \xi_t, \qquad \xi_t \sim \mathcal{N}(0, I).$$

- Stochastic Gradient Langevin Dynamics (SGLD): minibatch estimate of the gradient.
- Variance reduction: SGLD with control variates (SGLD-CV): use a reference point x^{ref} (e.g. updated when $t \equiv 0 \pmod{\tau}$)

$$\widehat{\nabla}_{\text{cv}} E(x_t) = \nabla E(x^{\text{ref}}) + \frac{N}{B} \sum_{n \in S_t} [\nabla E_n(x_t) - \nabla E_n(x^{\text{ref}})].$$

Stochastic Gradient HMC (SG-HMC)

- **Idea:** Combine the previous approaches
- Use a noisy gradient estimator

$$g(x_t, \xi_t) \approx \nabla E(x_t),$$

where ξ_t encodes the randomness.

A simple SG-HMC update can be written as:

$$x_{t+1} = x_t + \eta\, v_t - \frac{\eta^2}{2}\, g(x_t, \xi_t),$$

$$v_{t+1} = v_t - \frac{\eta}{2}\, g(x_t, \xi_t) - \frac{\eta}{2}\, g(x_{t+1}, \xi_{t+1/2}).$$

Contents

Convergence

Motivation I: The initial state may be far from the high-probability region of the target distribution.

- We introduce the **burn-in period**.

Practical convergence diagnostics

Motivation II: Even after burn-in, we must check whether the chain has *effectively* is exploring the target distribution well.

- **Multiple chains:** for comparison.
- **Autocorrelation plots:** ℓ autocorrelation $\rho_\ell = \mathrm{corr}(x_i, x_{i+\ell})$.
- **Potential scale reduction (R-hat):** compare variance *between* chains and *within* chains.
- **Effective Sample Size (ESS):**
  - Integrated autocorrelation time $\tau_{\text{int}} = 1 + 2\sum_{\ell=1}^{\infty} \rho_\ell$.
  - For a chain of length n, the effective sample size is

$$\mathrm{ESS} = \frac{n}{\tau_{\text{int}}}.$$

[Code S5]

Contents

Extensions I: Reversible Jump MCMC (RJMCMC)

- **Objective:** The dimension of the parameter vector is not fixed.

  Example: Gaussian mixture model (GMM) where the number of components K is unknown.

- A Markov chain state is (k, θ_k):

$$\pi(k, \theta_k) \propto p(\text{data} \mid k, \theta_k)\, p(\theta_k \mid k)\, p(k)$$

- **Key idea of RJMCMC:** enlarge the space with auxiliary variables and use *dimension-matching transformations* and work on a fixed-dimensional joint space.

RJMCMC: Proposals with Auxiliary Variables

- State: (k, θ_k) in model $\mathcal{M}_k$.
- Propose move to model $\mathcal{M}_{k'}$:
  1. Model index:
  $$k' \sim q(k' \mid k).$$
  2. Auxiliary variables:
  $$u \sim q_{k \to k'}(u \mid \theta_k), \quad u \in \mathbb{R}^{r_{k \to k'}}.$$
  3. Dimension-matching transformation between models:
  $$(\theta_{k'}, u') = T_{k \to k'}(\theta_k, u), \quad (\theta_k, u) = T_{k' \to k}(\theta_{k'}, u'),$$
  with inverse maps $T_{k' \to k} = T_{k \to k'}^{-1}$

RJMCMC: Acceptance Probability

$$A = \min\{1, \alpha\}$$

$$\alpha = \left\{ \underbrace{\frac{\pi(k', \theta_{k'})}{\pi(k, \theta_k)}}_{\substack{\text{posterior} \\ \text{ratio}}} \underbrace{\frac{q(k \mid k')}{q(k' \mid k)}}_{\substack{\text{model index} \\ \text{proposal ratio}}} \underbrace{\frac{q_{k' \to k}(u' \mid \theta_{k'})}{q_{k \to k'}(u \mid \theta_k)}}_{\substack{\text{auxiliary} \\ \text{proposal ratio}}} \underbrace{\left| \det \frac{\partial T_{k \to k'}(\theta_k, u)}{\partial(\theta_k, u)} \right|}_{\substack{\text{Jacobian} \\ \text{term}}} \right\}.$$

[Code S6]

Extensions II: Simulated Annealing (SA)

- Used to find the global optimum of a **multimodal** energy function $E(x)$.
- Define a tempered distribution:

$$p_T(x) \propto \exp\left(-\frac{E(x)}{T}\right), \qquad T \downarrow 0.$$

- Cooling schedule: e.g. $T_{t+1} = \gamma T_t$, $\gamma \in (0, 1)$.

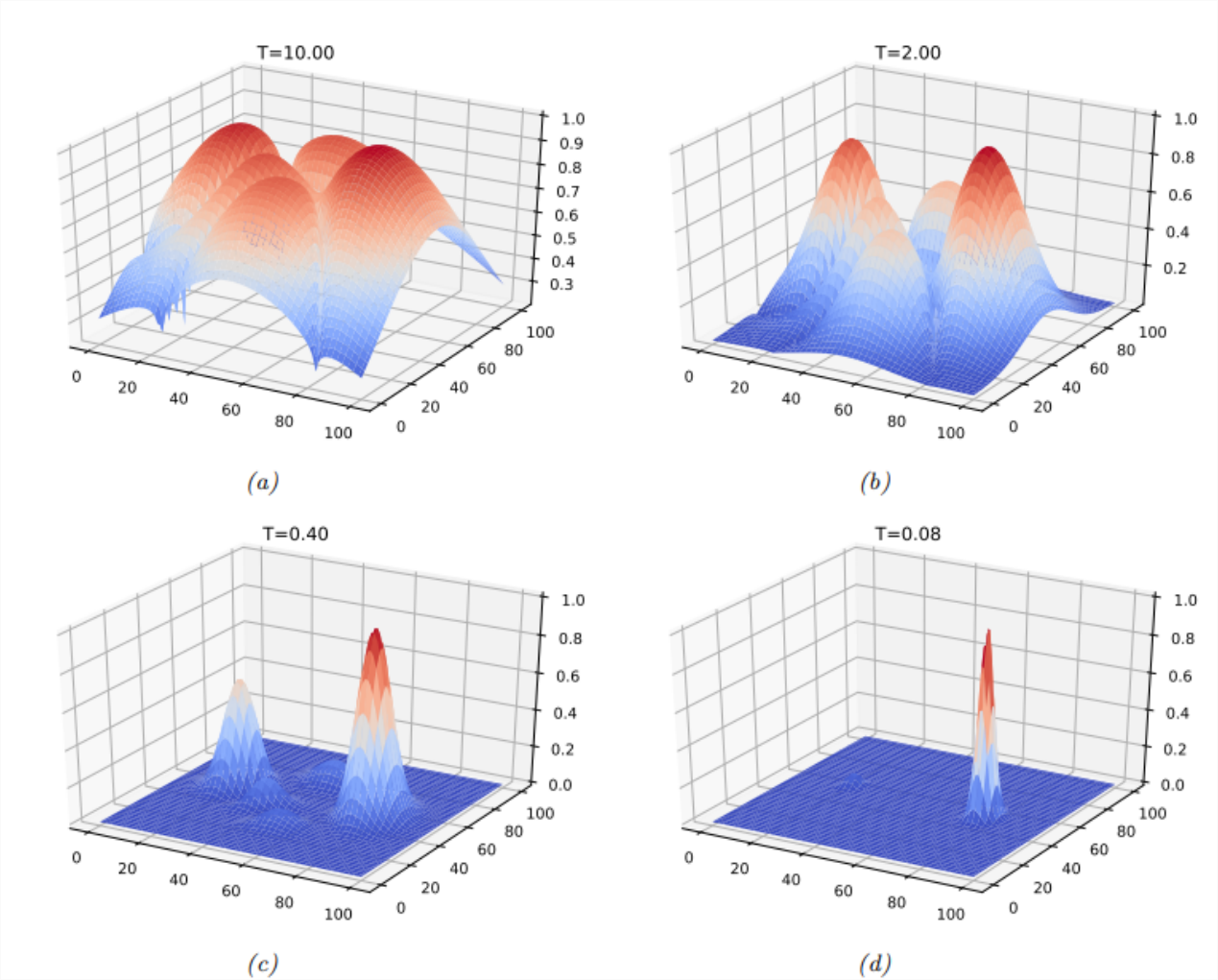


Figure 3: Temperature effect on Simulated Annealing

Contents

Key Takeaways

Core concepts:

1. Understand the **key concepts** of MCMC
2. Review different algorithms: **MH, Auxiliary variables, Gibss, HMC.**
3. Review **cutting-edge** gradient based MCMC algorithms and extensions
4. Get the sufficent knowledge to **verify convergence and performance**.
5. Your own code: `Numpyro, Pymc, Pyro.` or `STAN, OpenBUGS`

Extra refs

Langevin Dynamic: Girolami, M. and Calderhead, B. (2011). *Riemann manifold Langevin and Hamiltonian Monte Carlo methods*. JRSSB.

RJMCMC Green, P. J. (1995). *Reversible Jump Markov Chain Monte Carlo computation and Bayesian model determination*. Biometrika, 82, 711–732.

Thinning: Riabiz, M. et al. (2022). *Optimal thinning of MCMC output*. JRSSB.

HMC with repulsive forces: Gallego, V. and Ríos Insua, D. (2020). *Stochastic gradient MCMC with repulsive forces*. arXiv:1812.00071.

Questions?

Next Session

Sequential Monte Carlo (Ch. 13)

Dec 17, 2025

Mario Chacón-Falcón