Replica Exchange SGLD Methods: General ideas and numerics

Kaiwen Zhang, SEAS BS ‘24 Applied Mathematics
21 April 2023

References

Li, Guanxun, et al. "Fast Replica Exchange Stochastic Gradient Langevin Dynamics." *arXiv preprint arXiv:2301.01898* (2023).

Lin, Guang, et al. “Multi-variance replica exchange SGMCMC for inverse and forward problems via Bayesian PINN.” *Journal of Computational Physics, Volume 460,* 2022, 111173, ISSN 0021-9991, https://doi.org/10.1016/j.jcp.2022.111173.

Chen, Yi, et al. "Accelerating nonconvex learning via replica exchange Langevin diffusion." *arXiv preprint arXiv:2007.01990* (2020).

Deng, Wei, et al. "Non-convex learning via replica exchange stochastic gradient mcmc." *International Conference on Machine Learning*. PMLR, 2020.

Outline

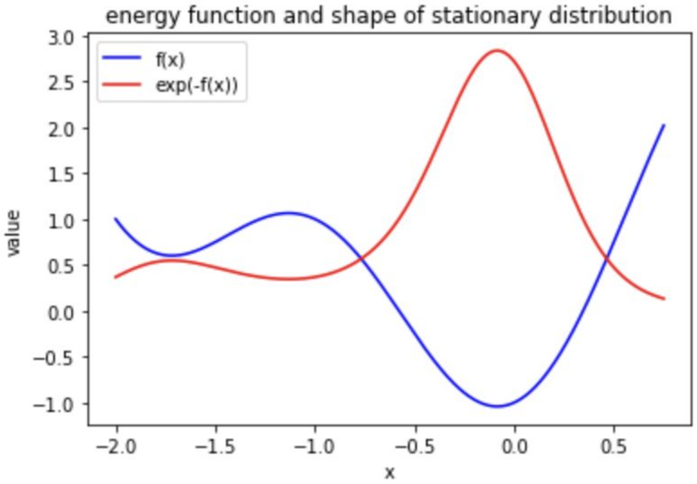
- The minimization problem
- Gradient descent, stochastic gradient descent (SGD)
- Langevin dynamics (LD), stochastic gradient LD (SGLD)
- Replica exchange SGLD
- Recent advances: m-reSGLD and f-reSGLD

The minimization problem and its formulation

Problem: minimize energy function E(x) with an algorithm, given incomplete information, including function values, derivatives, or their estimations.

Reformulation (MCMC): sample a set (chain) of arguments sequentially in the domain, such that the asymptotic distribution of samples is concentrated around the minimizer of the energy function.

- Used in analysis of minimization algorithms



Example of an energy function and an ideal distribution of samples

Gradient descent, stochastic gradient descent

Idea: the energy function decreases fastest along the negative gradient direction

Updating scheme (1D):

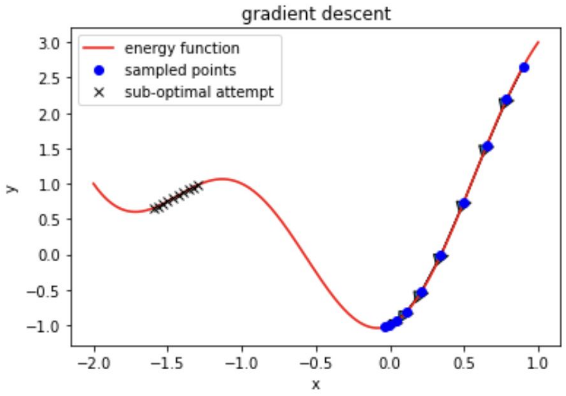
$$x_{n+1} = x_n - a\nabla E(x_n)$$

Convergence: derivative (hopefully) approaches 0 as sampled arguments approach internal minimum

Stochastic gradient:

$$x_{n+1} = x_n - a\nabla \hat{E}(x_n)$$

- Why: partial information, deliberate noise
- How: finite difference scheme, manually sample
- Applicable to all stochastic gradient schemes



Gradient descent, optimal and sub-optimal chains

Remark: the step size in SGD should really be a function of iteration step which decays to zero for sake of convergence

Langevin dynamics (LD)

Motivation: more deliberate noise

Updating scheme of LD:

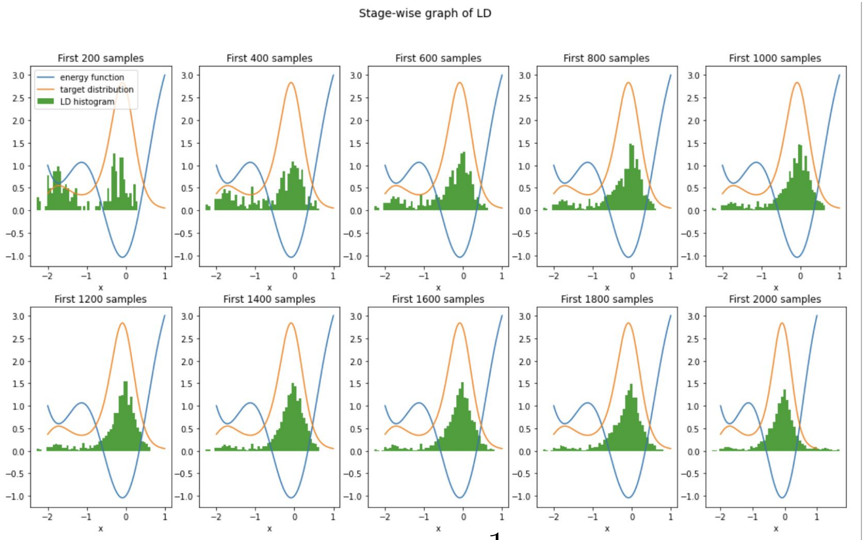
$$x_{k+1} = x_k - s_k \nabla U(x_k) + \sqrt{2 s_k \tau} \xi_k$$

ξ, the noise term: follows a standard normal distribution independent of iteration count
s, step size: depends on iteration count and decays to zero
τ, temperature: a positive number

Note:

- Shrinking step size function
- Root 2: probably has to do with Gaussian
- Initial guess matters, is another research field

Convergence: stationary distribution: $\exp\left(\frac{U}{\tau}\right)$



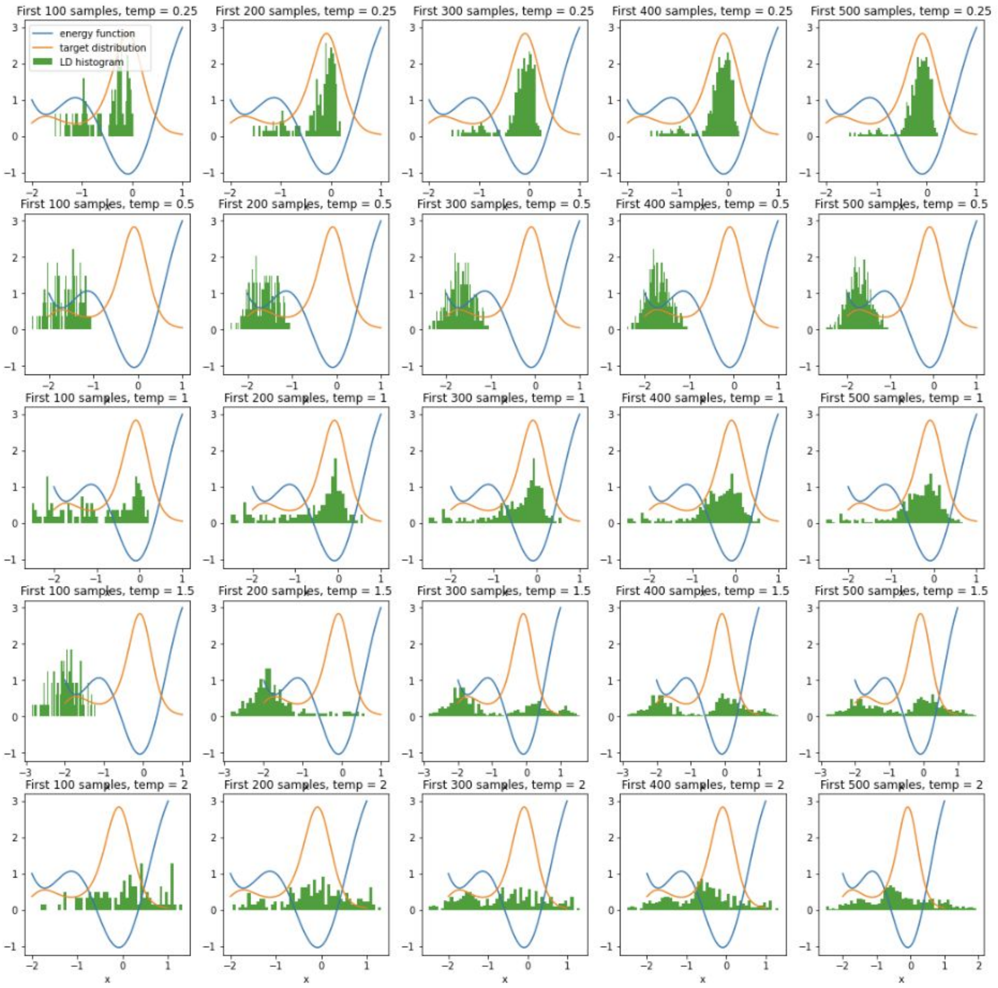
Step size: $$s_k = \frac{1}{(\log(2x+1)+5)^2}$$

Temperature = 1, number of iterations = 2000, initial guess = -1.3

Temperature τ; stochastic gradient

$$x_{k+1} = x_k - s_k \nabla U(x_k) + \sqrt{2 s_k \tau}\, \xi_k$$

- Low temp: highly concentrating
- High temp: rapidly exploring
- Add noise: stochastic gradient (SGLD)
- **Tradeoff**: speed v. concentration



Replica exchange SGLD (reSGLD)

Tradeoff: concentration v. efficiency

Solution: two communicating chains

Replica exchange SGLD updating scheme:

$$\hat{\theta}_{k+1}^{(1)} = \hat{\theta}_k^{(1)} - \eta_k \nabla \hat{U}(\theta_k^{(1)}) + \sqrt{2\eta_k \tau_1} \xi_k^{(1)}$$
$$\hat{\theta}_{k+1}^{(2)} = \hat{\theta}_k^{(2)} - \eta_k \nabla \hat{U}(\theta_k^{(2)}) + \sqrt{2\eta_k \tau_2} \xi_k^{(2)}$$

Swapping: two chains can swap positions with probability: (notice the estimated function values)

$$p(swap) = a \min\{1, \exp[\tau_\delta(\hat{U}(x_h^{(1)}) - \hat{U}(x_h^{(2)}) - \tau_\delta \sigma_U^2)]\}$$

where $\tau_\delta = \frac{1}{\tau_1} - \frac{1}{\tau_2}$

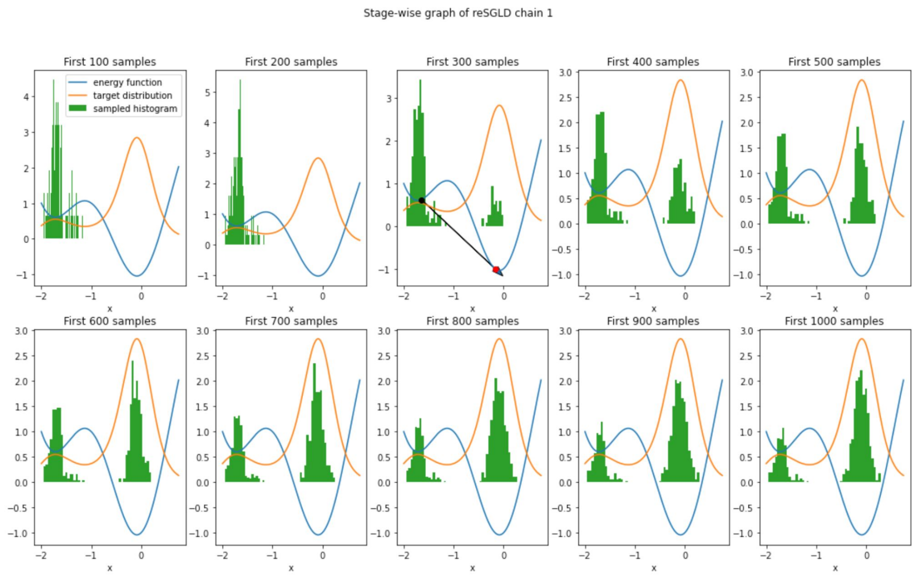
Swapping rate:

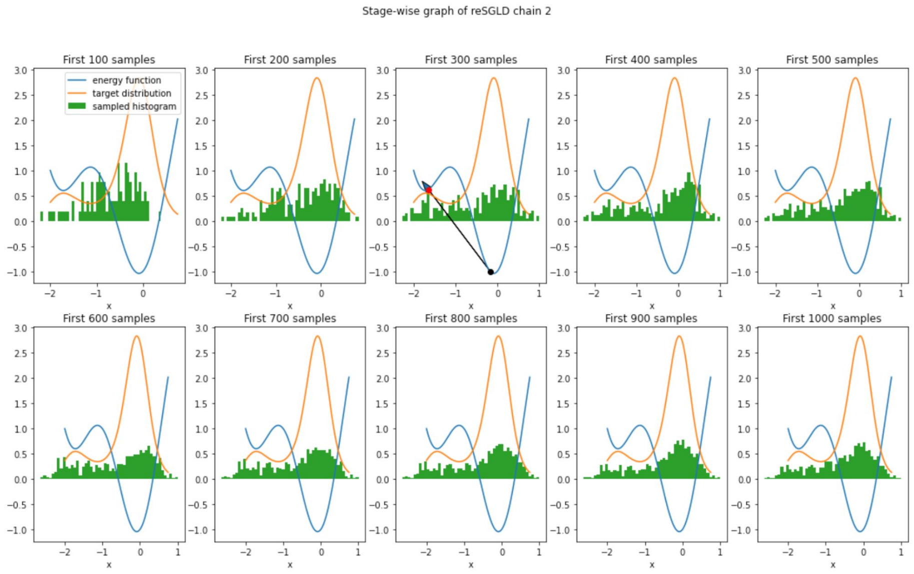
- Depends on difference of temperature
- Trusts difference in function values
- Penalizes large variance
- Admits manual manipulation

Remarks:

- Ways to perform estimation: finite difference, or manual simulation, pros and cons
- Swaps often acts as fail-safe mechanism

reSGLD in action: temp 1 = 0.25, temp 2 = 2, a = 5, sigma_grad = 1, sigma_u = 1





Remark: swap rarely happens; low temp chain captures global minimum faster (than running one-chain only)

Improvements on reSGLD

Motivations:

- More flexible in parameters
- Faster implementation

New ideas:

- Multi-variance reSGLD (m-reSGLD)
- Fast-tempering reSGLD (f-reSGLD)

Multi-variance reSGLD

Improvements:

- Two chains with different estimators
- Updated swapping rate: weighted sum

Motivation:

- Generalize parameter choice

Drawback:

- Slow implementation

Two implementations: finite difference gradient v. normally distributed gradient

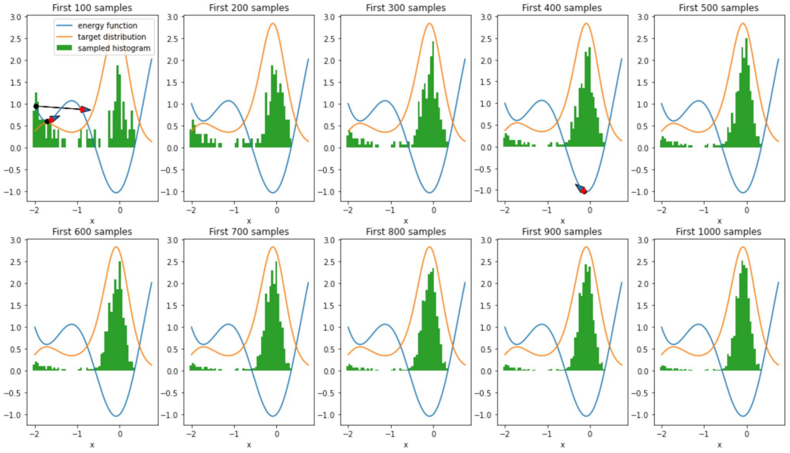
Updating scheme:

$$\hat{\theta}_{k+1}^{(1)} = \hat{\theta}_k^{(1)} - \eta_k \nabla \hat{U}_1(\theta_k^{(1)}) + \sqrt{2\eta_k \tau_1}\,\xi_k^{(1)}$$
$$\hat{\theta}_{k+1}^{(2)} = \hat{\theta}_k^{(2)} - \eta_k \nabla \hat{U}_2(\theta_k^{(2)}) + \sqrt{2\eta_k \tau_2}\,\xi_k^{(2)}$$

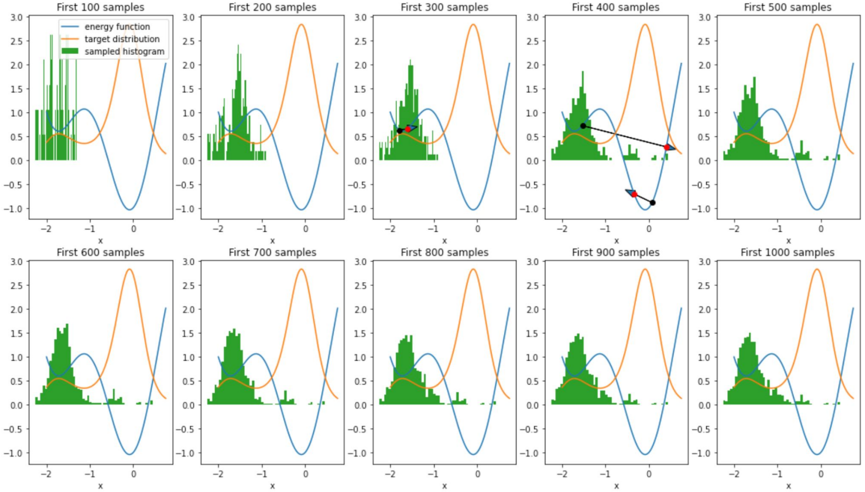
$$p(swap) = a\eta_k \min\{1, \hat{S}(\hat{\theta}_k^{(1)}, \hat{\theta}_k^{(2)})\}, \text{ where}$$

$$\hat{S}(\theta^{(1)}, \theta^{(2)}) = exp\{\tau_\delta[a_1(\hat{U}_1(\theta^{(1)}) - \hat{U}_1(\theta^{(2)})) + a_2(\hat{U}_2(\theta^{(1)}) - \hat{U}_2(\theta^{(2)})) - (a_1^2\sigma_1^2 + a_2^2\sigma_2^2)\tau_\delta]\}$$

m-reSGLD with finite difference: sensitivity to precision

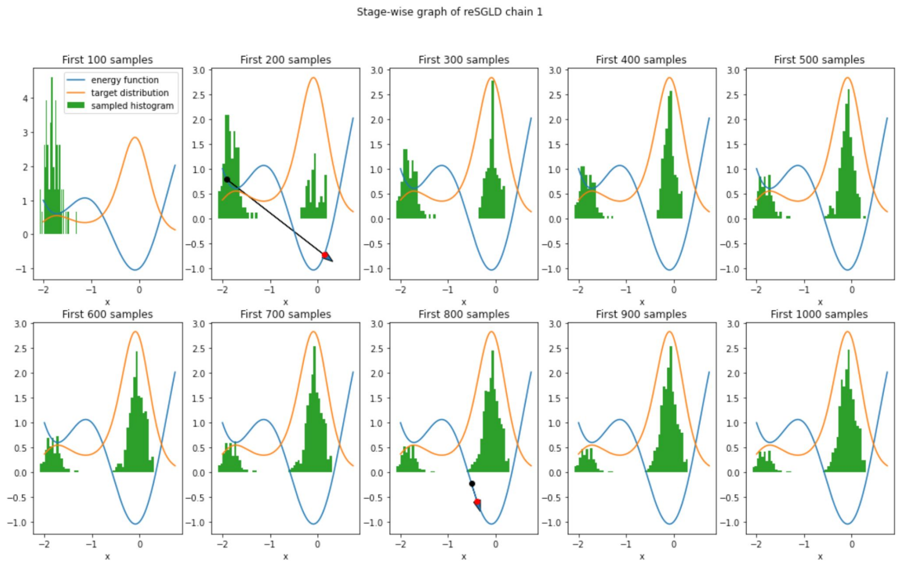


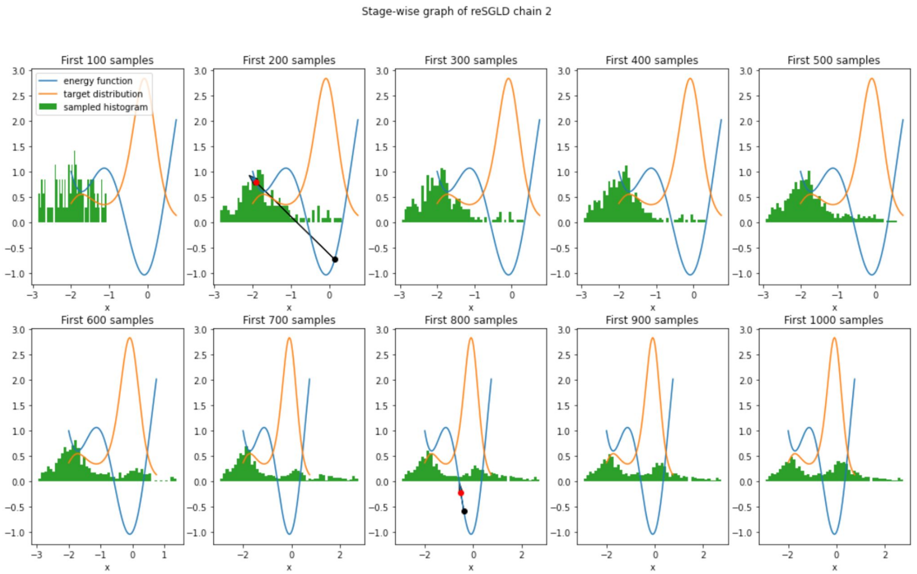
1e-4 and 2e-4 precisions



0.001 and 0.005 precisions - method fails

m-reSGLD with normally distributed gradient





Parameters: t_1 = 0.25, t_2 = 2, a = 5, sigma_1 = 0.5, sigma_2 = 1, sigma_grad_1 = sigma_grad_2 = 1
Notice the additional freedom one gets on parameters

Fast tempering reSGLD (f-reSGLD)

Goal: correct bias, accelerate implementation

Solution:

- bias correction term
- updated swapping rate

Updating scheme:

- Define corrections from gradient estimator:

$$\hat{\nabla U}(\theta) \sim N(\nabla U(\theta), s^2) \qquad c_k^2 = \tau\eta_k - \frac{1}{2}\eta_k^2 s^2$$

- Updating scheme:

$$\widetilde{\theta}^{(1)}_{k+1,\eta} = \widetilde{\theta}^{(1)}_{k,\eta} - \eta_k \widehat{\nabla U}_1(\widetilde{\theta}^{(1)}_{k,\eta}) + \sqrt{2}\widehat{c}_{1,k}(\widetilde{\theta}^{(1)}_{k,\eta})\xi^{(1)}_k$$

$$\widetilde{\theta}^{(2)}_{k+1,\eta} = \widetilde{\theta}^{(2)}_{k,\eta} - \eta_k \widehat{\nabla U}_2(\widetilde{\theta}^{(2)}_{k,\eta}) + \sqrt{2}\widehat{c}_{2,k}(\widetilde{\theta}^{(2)}_{k,\eta})\xi^{(2)}_k,$$

Swapping rate: (faster because of less evaluations of energy function)

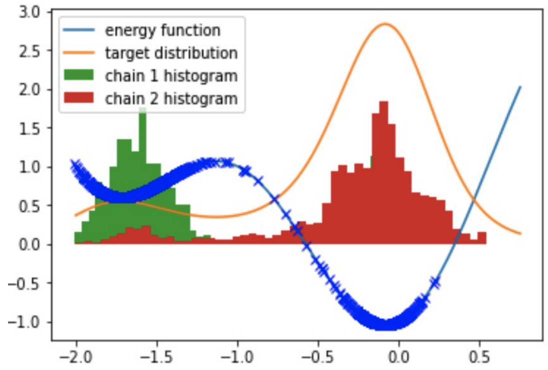
$$\boldsymbol{p} = a\eta_k \min 1, \boldsymbol{S}(\theta_k^{(1)}, \theta_k^{(2)})$$

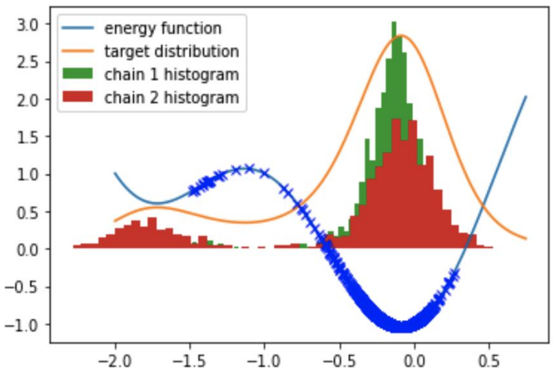
$$S(\theta^{(1)}, \theta^{(2)}) = \exp\{\tau_\delta[\hat{U}_1(\theta^{(1)}) - \hat{U}_2(\theta^{(2)}) - \frac{1}{2}\tau_\delta(\sigma_1^2(\theta^{(1)}) + \sigma_2^2(\theta^{(2)}))]\}$$

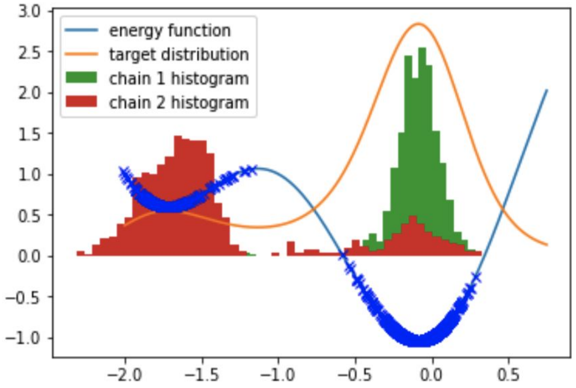
Comments:

- Implementation is indeed faster
- It’s doubtful whether the bias correction works
- Two independent estimators might be redundant for real world applications

f-reSGLD in action, and a concluding remark







Remark: it is likely impossible to say anything deterministic about behavior in “small” number of iterations.

Thank you!

Kaiwen (Kevin) Zhang
21 April 2023