Shanghai Jiao Tong University, June 10th, 2020

Maximum conditional entropy Hamiltonian Monte Carlo sampler

Jinglai Li

School of Mathematics

UNIVERSITY OF BIRMINGHAM

joint work with Tengchao Yu (SJTU) and Hongqiao Wang (CSU)

Motiviating Question

How to draw samples from a generic distribution?

- Statistical Physics: Energy Landscape
- Applied Mathematics: Rare event simulation
- Statistics: Bayesian inference
- Machine Learning: Bayesian Neural Network

Motiviating Question

How to draw samples from a generic distribution?

- Statistical Physics: Energy Landscape
- Applied Mathematics: Rare event simulation
- Statistics: Bayesian inference
- Machine Learning: Bayesian Neural Network

Motiviating Question

How to draw samples from a generic distribution?

- Statistical Physics: Energy Landscape
- Applied Mathematics: Rare event simulation
- Statistics: Bayesian inference
- Machine Learning: Bayesian Neural Network

Motiviating Question

How to draw samples from a generic distribution?

- Statistical Physics: Energy Landscape
- Applied Mathematics: Rare event simulation
- Statistics: Bayesian inference
- Machine Learning: Bayesian Neural Network

Motiviating Question

How to draw samples from a generic distribution?

- Statistical Physics: Energy Landscape
- Applied Mathematics: Rare event simulation
- Statistics: Bayesian inference
- Machine Learning: Bayesian Neural Network

Markov Chain Monte Carlo

- MCMC is arguably the most popular method to sample a distribution (first introduced by Metropolis *et al.* in 1953).
- MCMC repeatedly performs the following iterations:

  1. Draw a sample from the proposal distribution: $\mathbf{v} \sim \pi(\cdot|\mathbf{u}_k)$.
  2. Compute the acceptance probability:

$$a(u, v) = \min\{1, \frac{\pi(\mathbf{v})\pi(\mathbf{u}_k|\mathbf{v})}{\pi(\mathbf{u}_k)\pi(\mathbf{v}|\mathbf{u}_k)}\}.$$

  3. If $\rho \sim U[0, 1] < a$, let $\mathbf{u}_{k+1} = \mathbf{v}$, else let $\mathbf{u}_{k+1} = \mathbf{u}_k$.

- The key of MCMC is to design an efficient proposal distribution.

Hamiltonian Monte Carlo

- HMC constructs a very efficient proposal by using Hamiltonian Dynamics (proposed by [Duane et al., 1987] as Hybrid MC).
- Construct a Hamiltonian system:
  - Assume the target distribution is $\pi(\mathbf{x}) = \exp(-U(\mathbf{x}))$.
  - Let $\mathbf{x}$ be the position variable, and $U(\mathbf{x})$ be the potential energy
  - Introduces an auxiliary variable $\mathbf{p}$ to be the momentum of the system with kinetic energy $K(\mathbf{p})$.
  - The kinetic energy is usually taken to be of a quadratic form:

$$K(\mathbf{p}) = \frac{1}{2}\mathbf{p}^T M^{-1}\mathbf{p},$$

    where M is a pd symmetric matrix (called the mass matrix).
  - The dynamics of the constructed system is governed by

$$\frac{d\mathbf{x}}{dt} = \frac{\partial H}{\partial \mathbf{p}}, \quad \frac{d\mathbf{p}}{dt} = -\frac{\partial H}{\partial \mathbf{x}}. \tag{H}$$

Hamiltonian Monte Carlo

- Now recast the problem as to sample from the joint distribution

$$\pi(\mathbf{x}, \mathbf{p}) \propto \exp[-H(\mathbf{x}, \mathbf{p})], \quad H(\mathbf{x}, \mathbf{p}) = U(\mathbf{x}) + K(\mathbf{p}),$$

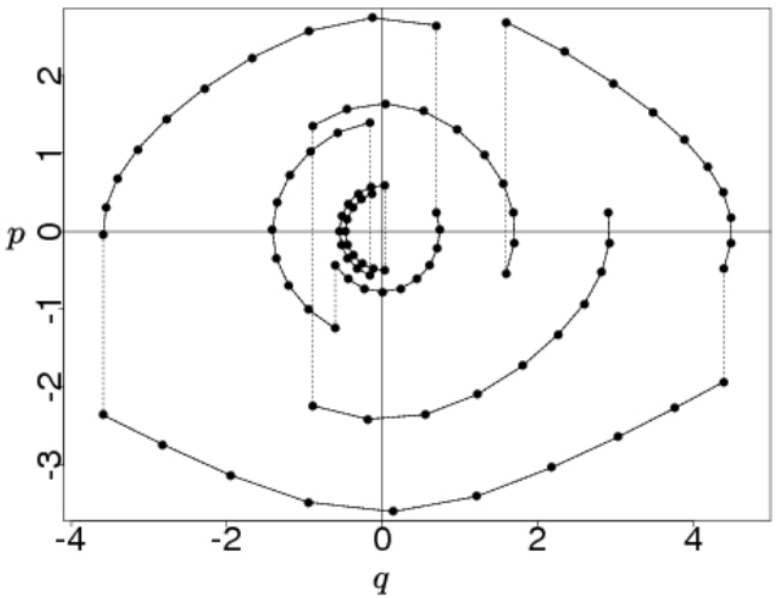
  and marginalizing the samples over $\mathbf{p}$ yields the samples of $\pi(\mathbf{x})$.

- HMC iteration: let $\mathbf{x}_0$ be the current position,
  - Sample an initial momentum state $\mathbf{p}_0$ from $N(0, M)$;
  - Solve the Eq. (H) with initial condition $(\mathbf{x}_0, \mathbf{p}_0)$, for a given amount of time T, obtaining the new states $(\mathbf{x}_T, \mathbf{p}_T)$;
  - Accept the new position $\mathbf{x}_T$ with probability

$$\min[1, \exp(H(\mathbf{x}_0, \mathbf{p}_0) - H(\mathbf{x}_T, \mathbf{p}_T))]. \qquad \text{(Acc)}$$

  Since the Hamiltonian system preserves its total energy, it should be clear that if we can solve the Eq. (H) exactly, the acceptance probability is simply one, meaning that all the proposed samples are accepted.

A schematic illustration of HMC



Leafrog Integrator

- In practice, however, Eq. (H) must be solved numerically and a numerical integrator is needed.
- To maintain detailed balance, we need the integrator to be time-reversible and symplectic.
- The leapfrog integrator is commonly used for its ability to preserve the time-reversibility and the symplecticity.
- When the integration time T is large, one often use multiple leapfrog steps to integrate Eq. (H) from 0 to T.

Algorithm 1 The leapfrog algorithm

function $(\mathbf{x}_T, \mathbf{p}_T)$=LEAPFROG$(\mathbf{x}_0, \mathbf{p}_0, M, U, \epsilon, L)$
  for $i = 1$ to L **do**
    Set $\mathbf{p} \leftarrow \mathbf{p}_0 + \frac{1}{2}\epsilon\nabla U(\mathbf{x}_0)$;
    Set $\mathbf{x} \leftarrow \mathbf{x}_0 + \epsilon M^{-1}\mathbf{p}$
    Set $\mathbf{p} \leftarrow \mathbf{p} + \frac{1}{2}\epsilon\nabla U(\mathbf{x})$;
  end for
  Set $\mathbf{x}_T \leftarrow \mathbf{x}$ and $\mathbf{p}_T \leftarrow \mathbf{p}$;
end function

- The acceptance probability < 1 due to numerical error.

Key Parameters in HMC

- The performance of HMC depends critically on the following algorithm parameters.
- The matrix M in the kinetic formula, called the mass matrix, represents the "directional mass" of the system.

  In standard HMC, M is simply taken to be identity, but can be chosen to improve HMC performance?
- The total integration time T, for how long one integrates the system before accepting/rejecting the sample.
- The numerical stepsize ϵ (or equivalently the number of steps L).
- Note the difference: M and T are HMC parameters and ϵ (or L) is the leapfrog parameter.
- Tuning these parameters are a key issue in HMC.

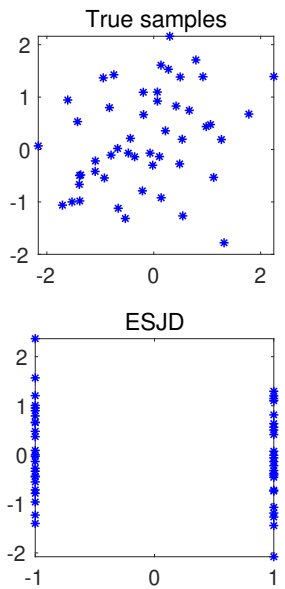
Parameter Tuning in HMC

- Ideal case: suppose that we can solve the Hamiltonian system exactly, achieving the 100% acceptance probability.
- How to choose matrix M and time T for the best efficiency of the HMC algorithm?

  To answer this question, we first need to establish an optimality criterion or a performance measure of the HMC proposal.
- A natural choice for such a measure is the Expected Squared Jumping Distance (ESJD) [Pasarica and Gelman, 2007]:

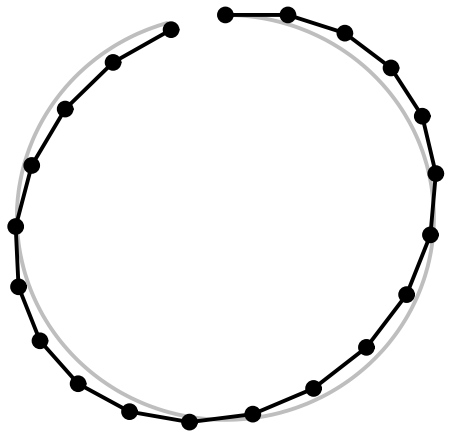
$$\mathbb{E}[\|\mathbf{x}_T - \mathbf{x}_0\|^2],$$

  seeking to move the largest distance from the present location.
- However, maximizing ESJD may lead to problematic proposals: in the simple Gaussian case, the resulting chain becomes periodic and loses its ergodicity.



No-U-turn sampler

- The No-U-turn sampler (NUTS) [Hoffman and Gelman, 2014] can address the periodic issue in ESJD.
- Try to avoid trajectories to retract its steps (i.e. "U-turn").



- Intuition: keep moving until the chain begins to make "U-turn" defined via the instantaneous distance gain: $(\mathbf{X}_T - X_0)^T \mathbf{p}$.
- Simply stopping when $(\mathbf{X}_T - X_0)^T \mathbf{p} < 0$ can not preserve reversibility and so NUTS uses a delicate and sophisticated scheme (constructing a binary tree) to achieve No-U-turn.

Kolmogorov-Sinai Entropy

- Maximizing the Kolmogorov-Sinai entropy (KSE) rate:

$$H_{KS} = \lim_{n \to \infty} \mathbb{H}(\mathbf{x}_n | \mathbf{x}_{n-1}, ..., \mathbf{x}_1)$$

- For a stationary time-reversible Markov chain, the KSE is equal to the conditional entropy (CE):

$$\mathbb{H}(\mathbf{x}_T | \mathbf{x}_0) = \int \log \pi(\mathbf{x}_T | \mathbf{x}_0) \pi(\mathbf{x}_T, \mathbf{x}_0) d\mathbf{x}_T d\mathbf{x}_0,$$

  which means that the algorithm parameters are determined by maximizing the CE of the chain.

- Intuition: maximizing the proposal entropy at each step helps to explore the state space.

- It is suggested in [Mihelich et al., 2018] that maximizing the KSE can yield very close results to those of minimizing the mixing time, but not theoretical justification to our knowledge.

Near-Gaussian distributions

- Computing CE for a general proposal is highly challenging.
- We focus on a special case where the target distribution is near-Gaussian. Such near-Gaussian distributions arise frequently in Bayesian inference, especially when the amount of data is large, thanks to asymptotic normality [Gelman et al., 2013].
- In this case we can first derive the algorithm parameters by assuming the target distribution is exactly Gaussian, and it should be sensible to use derived parameter values for the actual target distribution provided it does not deviate too far from Gaussian.
- In fact, as will be demonstrated by examples the method is rather forgiving in terms of Gaussianity (e.g. unimodal).
- Assume that the target distribution is approximated by a Gaussian $\pi(\mathbf{x}) \approx \mathcal{N}(\mu, \Sigma)$, where we take $\mu = 0$ for simplicity.
- The resulting HMC proposal $\pi(\mathbf{x}_T|\mathbf{x}_0)$ is also Gaussian.

Optimality results

- When $\pi(\mathbf{x}_T|\mathbf{x}_0)$ is Gaussian, the CE becomes
$$\mathbb{H}(\mathbf{x}_T|\mathbf{x}_0) = \mathbb{E}_{\mathbf{x}_0}[\log\det \mathrm{Cov}[\mathbf{x}_T|\mathbf{x}_0]].$$
- The parameters are determined by,
$$\max_{\{M\in\mathcal{M}_\Sigma^+, T\in R^+\}} \mathbb{H}(\mathbf{x}_T|\mathbf{x}_0) = \mathbb{E}_{\mathbf{x}_0}[\log\det \mathrm{Cov}[\mathbf{x}_T|\mathbf{x}_0]], \qquad \text{(MCE)}$$
where $\mathcal{M}_\Sigma^+$ is the class of p.d. matrices that commutes with Σ.
- The following theorem states our main results:

Theorem

Suppose that $\pi(x) = \mathcal{N}(0,\Sigma)$, $x_0 \sim \pi(x)$, $p_0 \sim \mathcal{N}(0, M)$, *and* $x(t)$ *is the solution of the Hamiltonian system* (H). *Let* $x_T = x(T)$, *and a solution of the optimization problem* (MCE) *is*
$$M = \Sigma^{-1}, \quad \text{and} \quad T = (2m+1)\pi/2,$$
for an arbitrary non-negative integer m.

Remarks on the results

- In practice, since the Hamiltonian system needs to be solved numerically, it is certainly desirable to use smaller integration time T, and thus in the HMC algorithm we set $m = 1$ and $T = \pi/2$.
- While the trick to improve the efficiency of HMC by choosing $M = \Sigma^{-1}$ has long been known from an intuitive perspective [Neal, 2011], we are able to provide a justification for it based on the maximum CE (MCE) principle.
- Another important issue in the method is that it requests the knowledge of the target covariance matrix. Here we follow the idea of the adaptive MCMC algorithms to estimate the covariance from the sample history [Haario et al., 2001].

The number of leapfrog steps

- The number of leapfrog steps, L, is another key algorithm parameter to be specified.
- If we increase L, the numerical integration becomes more accurate and the acceptance probability approaches to 1, and the price to pay is that more leapfrog steps means higher computational cost.
- Since the total integration time T is fixed, it is reasonable to assume that increasing L, which in turn increases the acceptance probability, improves the performance of the algorithm.
- Based on this idea, we shall seek the value of L that provides the highest acceptance rate per computational cost (measured by L).
  1. draw a number of samples with L and record the average acceptance probability Acc;
  2. if Acc/L is larger than that of the previous step, let $L = \rho L$ and return to step (1); otherwise stoping updating L.

ρ is a prescribed scalar larger than one.

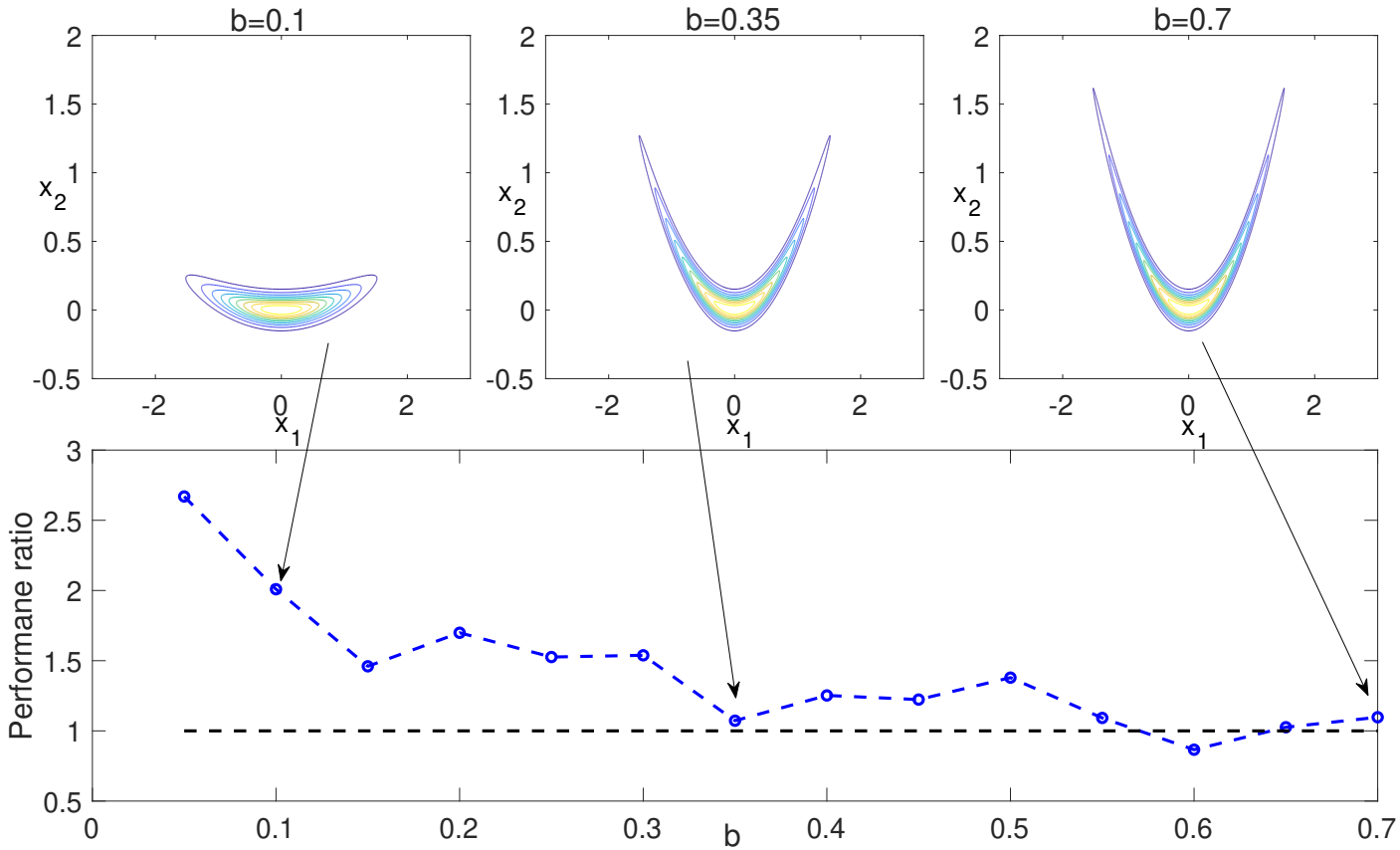
The Maximum Conditional Entropy Sampler

Algorithm 2 Maximum Conditional Entropy Sampler

Require: $U(\mathbf{x})$, Acc_{min}, N_0, L_0, L_{max}, N_{max}, N_M, N_L, ρ, I_{max}

Initialization: draw N_0 samples $\{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_{N_0}\}$ using standard HMC sampler.
Estimate the sample covariance matrix $\hat{\Sigma}$ of $\{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_{N_0}\}$;
Let $M = \hat{\Sigma}^{-1}$;
Let $T = \pi/2$;
Let $Acc_{old} = 0$, $L_{old} = L_0$, $L = L_0$, and $I_L = 1$,$I_M = 0$;
for $t = N_0$ to N_{max} **do**
    $\epsilon = T/L$;
    Draw $\mathbf{p}_t \sim N(0,\ M)$;
    $(\mathbf{x}^*,\ \mathbf{p}^*) = \text{leapfrog}(\mathbf{x}_t, \mathbf{p}_t, M, U(\mathbf{x}), \epsilon, L)$;
    Draw $u \sim U(0, 1)$;
    if $u < \min\{1, \exp(H(\mathbf{x}_t, \mathbf{p}_t) - H(\mathbf{x}^*, \mathbf{p}^*))\}$ **then**
        $\mathbf{x}_{t+1} = \mathbf{x}^*$;
    else
        $\mathbf{x}_{t+1} = \mathbf{x}_t$;
    end if
    if $t \mod N_L = 0$ **then**
        Let Acc be the average acceptance probability of the last N_L samples;
        if $t < N_M$ and ($I_M = 1$ or $Acc > 0$) **then**
            Update the sample covariance matrix $\hat{\Sigma}$ with the last N_L samples;
            Let $M = \hat{\Sigma}^{-1}$;
            $I_M = 1$;
        end if
        if $I_L = 1$ **then**
            if $L = L_{max}$ **then**
                $I_L = 0$;
                if $Acc/L < Acc_{old}/L_{old}$ **then**
                    $L = L_{old}$;
                end if
            else
                if $Acc > Acc_{min}$ **then**
                    if $Acc/L < Acc_{old}/L_{old}$ **then**
                        $I_{count} = I_{count} + 1$;
                        if $I_{count} \geq I_{max}$ **then**
                            $I_L = 0$, $L = L_{old}$;
                        end if
                    else
                        $Acc_{old} = Acc$, $L_{old} = L$;
                        $I_{count} = 0$;
                        $L = \min\{[\rho L_{old}], L_{max}\}$
                    end if
                else
                    $Acc_{old} = Acc$, $L_{old} = L$;
                    $I_{count} = 0$;
                    $L = \min\{[\rho L_{old}], L_{max}\}$.
                end if
            end if
        end if
    end if
end for

The Rosenbrock distribution

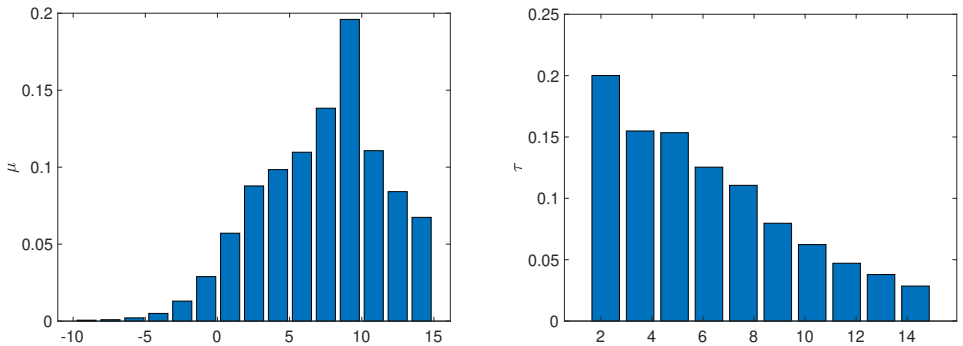


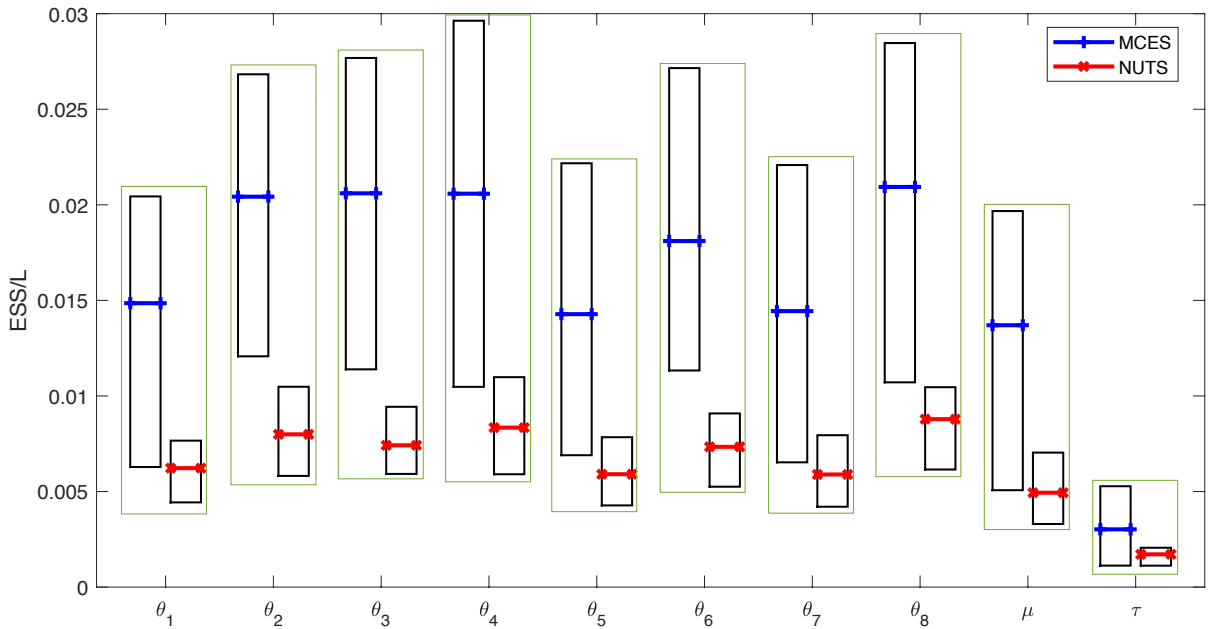
The ESS/L ratio between MCES and NUTS.

Effective sample size (ESS) is a common performance measure of MCMC. The results suggest that MCES is rather forgiving in terms of the near-Gaussian assumption.

The eight school problem

Estimating the effectiveness of training programs of eight schools for preparing their students for a SAT-V test. Ten parameters where at least two of them are substantially non-Gaussian.





MCES results have higher mean and larger variance. Lowest ESS/L on τ.

The log-Gaussian Cox Process

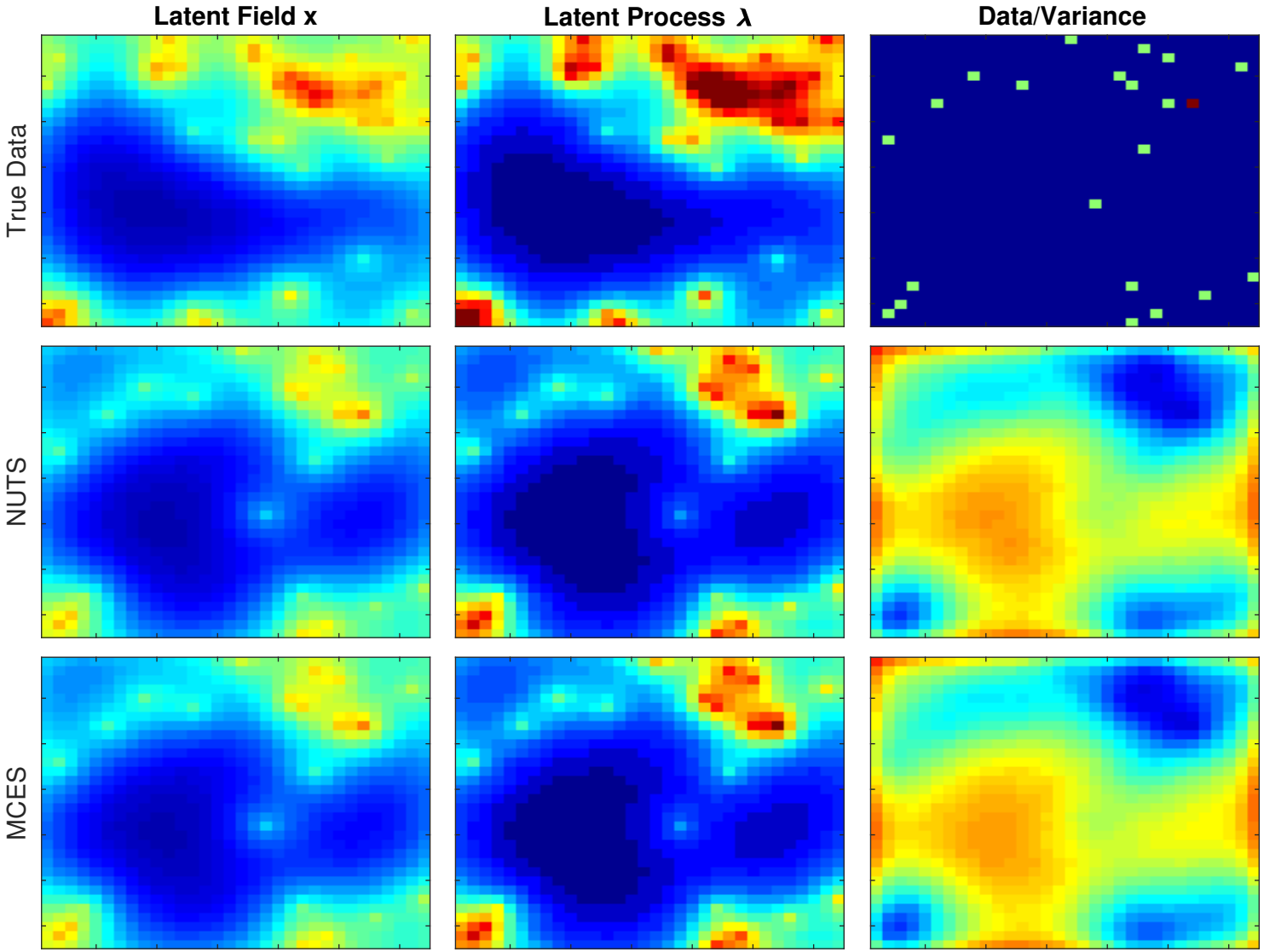
- The Log-Gaussian Cox process (LGCP) is a widely used inference model for spatial point process data [Moller et al., 1998], and has applications in ecology, geology, seismology, and neuroimaging.
- Mathematically it is a hierarchical structure consisting of a Poisson point process with a random log-intensity given by a Gaussian random field.

  Let $\boldsymbol{X}(z)$ be a Gaussian random field, and let $\boldsymbol{Y} = \{y_{ij}\}$ be the data set where the data points $y_{i,j}$ are Poisson distributed with mean $\lambda(z_{i,j})$:

$$y_{i,j} \sim \text{Poisson}(\cdot, \lambda(z_{i,j})), \quad \lambda(z) = s \exp(\mathbf{X}(z)),$$

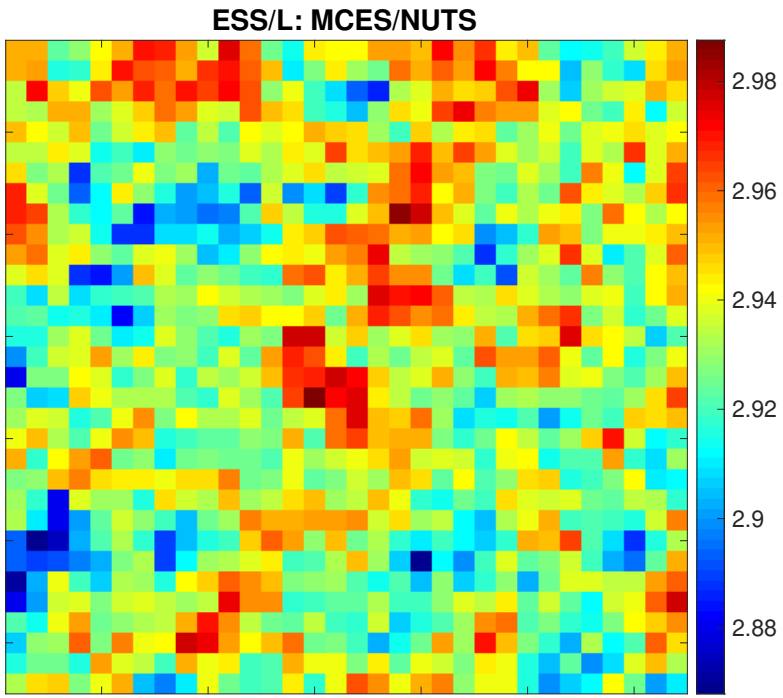
  where s is a positive scalar parameter. The goal of the problem is to compute the posterior distribution of $\boldsymbol{X}$ given the data $\boldsymbol{Y}$.
- Sampling the posterior distribution for the LGCP model is computationally challenging largely due to the high dimensionality of such problems. In this example we consider a 32×32 image, yielding a 1024-dimensional problem.

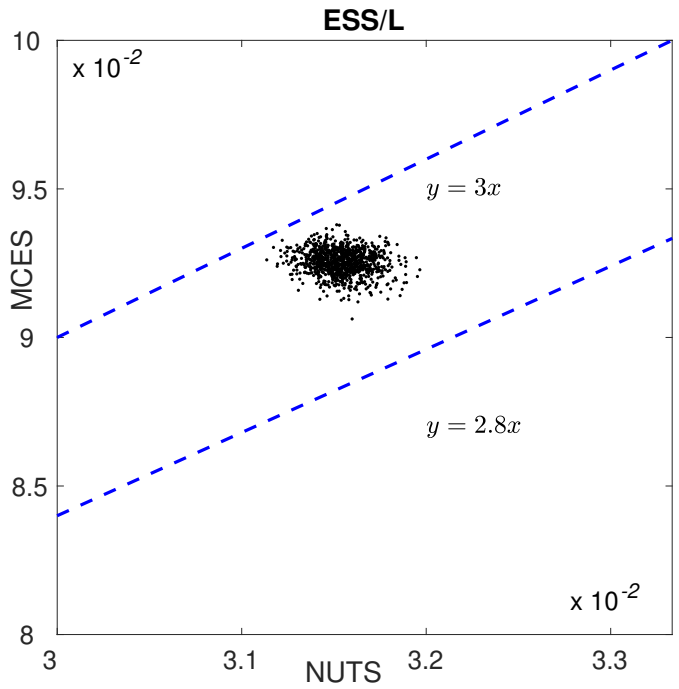
The posterior distribution



Posteriors computed by MCES and NUTS are rather similar.

Performance comparison





ESS/L of MCES is around 3 times as that of NUTS.

Conclusions

- Propose a CE/KSE based criterion for tuning HMC proposals and we expect that it can be extended to generic MCMC.
- Develop an adaptive Maximum CE HMC sampler.
- Key limitation: the near-Gaussian assumption. More tests needed to understand how limiting this assumption is.
- Theoretical question: can CE/KSE be linked to the mixing time?
- Work-in-progress: develop a local and non-adaptive version by using the Hessian information; apply it to large scale Bayesian Neural Networks.

Preprint: Yu, T., Wang, H., Li, J. (2019). *Maximum conditional entropy Hamiltonian Monte Carlo sampler*. arXiv:1910.05275.
Source code: `https://github.com/SiriusYtc/MCES`

Main references (an incomplete list)


Duane, S., Kennedy, A. D., Pendleton, B. J., and Roweth, D. (1987).
Hybrid monte carlo.
Physics letters B, 195(2):216–222.


Gelman, A., Stern, H. S., Carlin, J. B., Dunson, D. B., Vehtari, A., and Rubin, D. B. (2013).
Bayesian data analysis.
Chapman and Hall/CRC.


Haario, H., Saksman, E., Tamminen, J., et al. (2001).
An adaptive metropolis algorithm.
Bernoulli, 7(2):223–242.


Hoffman, M. D. and Gelman, A. (2014).
The no-u-turn sampler: adaptively setting path lengths in hamiltonian monte carlo.
Journal of Machine Learning Research, 15(1):1593–1623.


Mihelich, M., Dubrulle, B., Paillard, D., Kral, Q., and Faranda, D. (2018).
Maximum kolmogorov-sinai entropy versus minimum mixing time in markov chains.
Journal of Statistical Physics, 170(1):62–68.


Moller, J., Syversveen, A. R., and Waagepetersen, R. P. (1998).
Log gaussian cox processes.
Scandinavian journal of statistics, 25(3):451–482.


Neal, R. M. (2011).
Mcmc using hamiltonian dynamics.
In Brooks, S., Gelman, A., Jones, G., and Meng, X.-L., editors, *Handbook of markov chain monte carlo*, pages 131–162. Chapman & Hall/CRC.


Pasarica, C. and Gelman, A. (2007).
Adaptively scaling the metropolis algorithm using expected squared jumped distance.
Statistica Sinica, 20(1):343–364.