02.20 The performance of estimating genetic parameters by Hamiltonian Monte Carlo and No-U-Turn Samper

M. Nishio, A Arakawa, K. Ishii, & O. Sasaki

NARO Institute of Livestock and Grassland Science, Tsukuba, 305-0901 Ibaraki, Japan, mtnishio@affrc.go.jp

Objective

This study compared the performance of estimating genetic parameters for Gibbs sampling (GS), Hamiltonian Monte Carlo (HMC) and No-U-Turn Sampler (NUTS) in both simulated and real pig data.

Theory

◆HMC

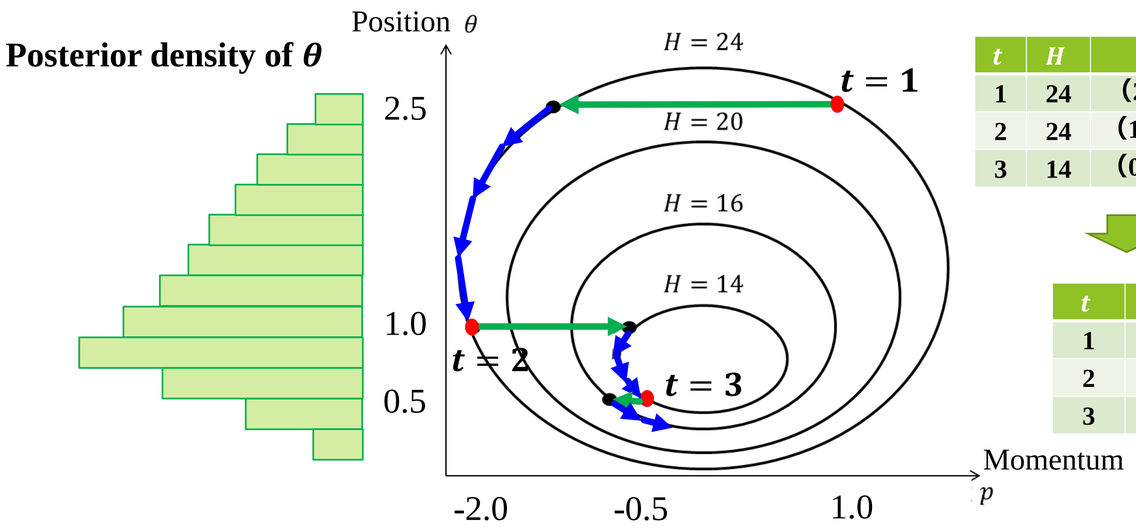
HMC is a Metropolis algorithm which uses Hamiltonian dynamics to create proposals.

$$\underset{\text{Hamiltonian}}{H(\boldsymbol{\theta},\mathbf{p})} = \underset{\text{Potential energy}}{U(\boldsymbol{\theta})} + \underset{\text{Kinetic energy}}{K(\mathbf{p})}$$

Let $f(\boldsymbol{\theta})$ be the posterior of parameter $\boldsymbol{\theta}$. Let $\mathbf{p}$ be an auxiliary momentum variable following a standard normal distribution. In HMC, $U(\boldsymbol{\theta})$ and $K(\mathbf{p})$ are defined as $U(\boldsymbol{\theta}) = -f(\boldsymbol{\theta})$ and $K(\mathbf{p}) = \mathbf{p}'\mathbf{p}/2$. The joint density of $(\boldsymbol{\theta},\mathbf{p})$ has the form:

$$f(\boldsymbol{\theta},\mathbf{p}) \propto \exp\left(f(\boldsymbol{\theta}) - \frac{1}{2}\mathbf{p}'\mathbf{p}\right) = \exp(-U(\boldsymbol{\theta}) - K(\mathbf{p})) = \exp(-H(\boldsymbol{\theta},\mathbf{p}))$$

HMC draws from the joint space of $(\boldsymbol{\theta},\mathbf{p})$, discards $\mathbf{p}$, and retains $\boldsymbol{\theta}$ as samples from $f(\boldsymbol{\theta})$. At each iteration, the HMC algorithm first generates the variables $\mathbf{p}$ and then follows with a Metropolis update that includes many leapfrog steps along a trajectory while maintaining the total energy of the system.



t	H	(θ, p)
1	24	(2.5, 1.0)
2	24	(1.0, -2.0)
3	14	(0.5, -0.5)

t	θ
1	2.5
2	1.0
3	0.5

Leapfrog step (step size: ε, number of steps: L)

Move by random number from univariate normal distribution

HMC is a powerful algorithm because it will achieve a high level of acceptance ratio and requires only first-order posterior information for the leapfrog step. However, its performance depends strongly on choosing suitable values for two tuning parameters: ε and L.

◆NUTS

NUTS automatically selects an appropriate value of L in each iteration in order to maximize the distance at each leapfrog step and avoid the random-walk behavior. Let Q be the half the squared distance between the current position $\boldsymbol{\theta}^*$ and the initial position $\boldsymbol{\theta}$ at each leapfrog step. The motivation is to run leapfrog steps until $\boldsymbol{\theta}^*$ starts to move back towards $\boldsymbol{\theta}$ (U-Turn). This is accomplished by the algorithm in which one runs leapfrog steps until the derivative of Q with respect to step (τ) becomes less than 0:

$$\frac{\partial Q}{\partial \tau} = \frac{\partial}{\partial \tau}\frac{(\boldsymbol{\theta}^* - \boldsymbol{\theta})'(\boldsymbol{\theta}^* - \boldsymbol{\theta})}{2} = (\boldsymbol{\theta}^* - \boldsymbol{\theta})'\mathbf{p} < 0$$

NUTS automatically tunes ε by applying dual averaging algorithm (Nesterov, 2009) to obtain the high acceptance ratio.

Models

◆Univariate animal model

$$\mathbf{y} = \mathbf{Xb} + \mathbf{Za} + \mathbf{e} \qquad \mathbf{a} \sim \mathrm{N}(\mathbf{0}, \mathbf{A}\sigma_a^2), \mathbf{e} \sim \mathrm{N}(\mathbf{0}, \mathbf{I}\sigma_e^2)$$

◆First-order derivatives for log posterior

$$\frac{d}{d\mathbf{b}}\log p(\boldsymbol{\theta}|\mathbf{y}) = \frac{\mathbf{X}'(\mathbf{y} - \mathbf{Xb} - \mathbf{Za})}{\sigma_e^2} \qquad \frac{d}{d\sigma_a^2}\log p(\boldsymbol{\theta}|\mathbf{y}) = -\frac{n_q}{2\sigma_a^2} + \frac{\mathbf{a}'\boldsymbol{A}^{-1}\mathbf{a}}{2\sigma_a^4}$$

$$\frac{d}{d\mathbf{a}}\log p(\boldsymbol{\theta}|\mathbf{y}) = -\frac{\boldsymbol{A}^{-1}\mathbf{a}}{\sigma_a^2} + \frac{\mathbf{Z}'(\mathbf{y} - \mathbf{Xb} - \mathbf{Za})}{\sigma_e^2}$$

n_q: number of animals

n: number of records

$$\frac{d}{d\sigma_e^2}\log p(\boldsymbol{\theta}|\mathbf{y}) = -\frac{n}{2\sigma_e^2} + \frac{(\mathbf{y} - \mathbf{Xb} - \mathbf{Za})'(\mathbf{y} - \mathbf{Xb} - \mathbf{Za})}{2\sigma_e^4}$$

◆Sampling

- Iteration: 10,000 (The first 1,000 iterations were discarded)
- Hyperparameter : ε = 0.01 and L=100 (simulated data)
  (HMC) ε = 0.001~10 and L = 3~200 (pig data)

Data

◆Simulated data

- The data were generated by using QMSim. (Sargolzaei & Schenkel, 2009)
- Infinitesimal model
- Population size: 1000 (500 males and 500 females)
- True heritability: 0.2 (Phenotypic variance: 1.0)
- Replicate: 5
- Fixed effect: sex

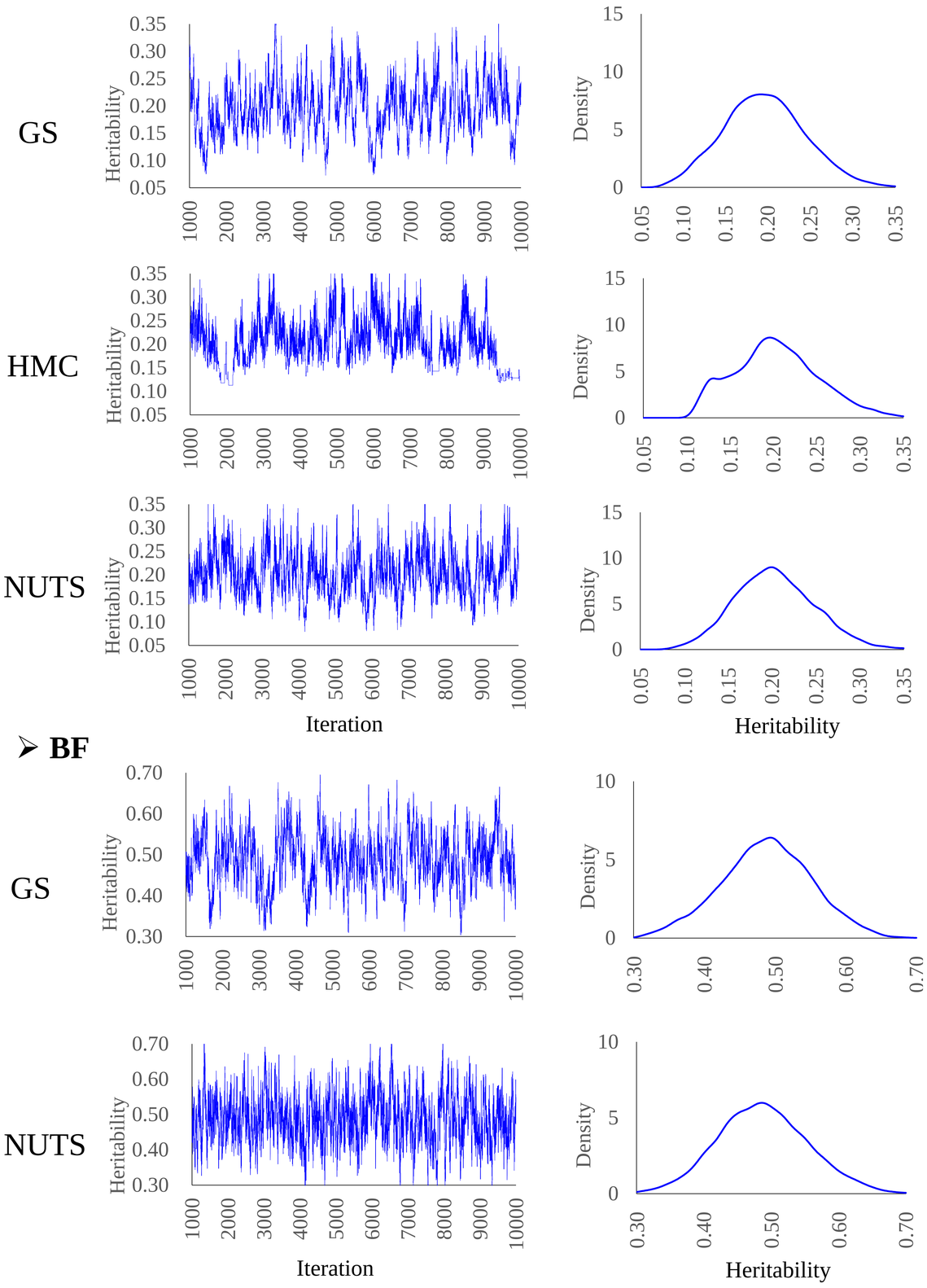
◆Pig data

- Duroc purebred pigs at the National Livestock Breeding Center in Japan.
- 1,521 pedigree data and 991 records
- Trait: backfat thickness (BF) and loin eye muscle (LEA)
- Fixed effect: sex (three classes; boar, barrow and gilt)
  generation (seven classes)

Results

◆Trace plot and posterior density of heritability

➢ Simulated data



➢ BF

◆Estimates of heritability (SD) and effective sample size (ESS)

Trait	Method	Estimates	ESS
Simulated data	GS	0.21 (0.05)	77
	HMC	0.25 (0.05)	88
	NUTS	0.22 (0.05)	157
BF	GS	0.48 (0.06)	164
	HMC	-	-
	NUTS	0.49 (0.06)	259
LEA	GS	0.57 (0.07)	142
	HMC	-	-
	NUTS	0.57 (0.07)	233

※ HMC could not estimate heritability in pig data.

Conclusions

- NUTS was computationally efficient approach in the field of animal breeding.
- HMC might required hands-on tuning of hyperparameters according to a trait and a population structure.

Acknowledgement: This research was supported by grants from the Project of the Bio-oriented Technology Research Advancement Institution, NARO (the special scheme project on advanced research and development for next-generation technology).