Adversarial Bayesian Simulation

Veronika Ročková and *Yuexi Wang*

Booth School of Business
University of Chicago

Bayesian Inference with Intractable Likelihoods

Our framework: data $X^{(n)} = \{X_i\}_{i=1}^n$ realized from $P_0 = P_{\theta_0}$ indexed by $\theta_0 \in \Theta$ with a prior $\pi(\theta)$.

We assume that P_θ, for each $\theta \in \Theta$, admits a density p_θ.

We want to draw from the posterior

$$\pi_n(\theta \mid X^{(n)}) = \frac{p_\theta(X^{(n)})\pi(\theta)}{\int_\Theta p_\vartheta(X^{(n)})\, d\Pi(\vartheta)}. \tag{1}$$

Our focus is on situations where the likelihood p_θ is too costly to evaluate ☹ *but can be readily sampled from ☺.*

- ⇝ Lotka-Volterra model: Population dynamics of animals in ecology
- ⇝ Heston model: Stochastic volatility dynamics in finance
- ⇝ Dynamic choice models: consumer dynamics in marketing

Bayesian Inference with Intractable Likelihoods

ABC: A generator simulates fake data from p_θ and reduces them to summary statistics

$$\pi(\theta) \times \prod_{i=1}^{n} p_\theta(X_i^\theta)$$

$$(\theta, X^\theta) \Rightarrow \|s(X^\theta) - s(X)\| > \varepsilon$$

$$\boxed{(\theta, X^\theta) \Rightarrow \|s(X^\theta) - s(X)\| < \varepsilon}$$

$$(\theta, X^\theta) \Rightarrow \|s(X^\theta) - s(X)\| > \varepsilon$$

- ☹ *Reliance on summary statistics*
- ☹ *Only proper priors*
- ☹ *Not great for model comparisons*
- ☺ *Parallel computation feasible*

Bayesian Inference with Intractable Likelihoods

$$\pi_n(\theta|X) \propto \pi(\theta) \times \prod_{i=1}^{n} p_\theta(X_i)$$

Likelihood of $s(X)$

$$p_\theta(s(X))$$

Synthetic likelihood MH

Unbiased estimator

$$\prod_{i=1}^{n} \hat{p}_\theta(X_i)$$

Pseudo-marginal MH

Bayesian Synthetic Likelihood: Constructs a likelihood from summary statistics

☹ *Reliance on summary statistics*

Pseudo-marginal MH: Replace the likelihood in the MH ratio with an importance sampling estimate

☹ *Many simulation realizations*

☹ *May not yield the correct stationary distribution*

Turning GANs into Posterior Simulators

Generative Adversarial Networks are a two-player minimax game

$$(g^*, d^*) = \arg\min_{g} \max_{d} \left[\mathrm{E}_{X \sim P_0} \log d(X) + \mathrm{E}_{Z \sim \pi_Z} \log(1 - d(g(Z))) \right] \quad (2)$$

g^* minimizes the *Jensen-Shannon divergence*

$$JS(P_0, P_g), \quad \text{where} \quad g(Z) \sim P_g \quad \text{and} \quad Z \sim \pi_Z.$$

Wasserstein GANs instead minimize

$$d_W\Big(P_0, P_g\Big) = \sup_{f \in \mathcal{F}_W} \left| E_{X \sim P_0} f(X) - E_{X \sim P_g} f(X) \right| \text{ where } \mathcal{F}_W = \{ f : \|f\|_L \leq 1 \}.$$

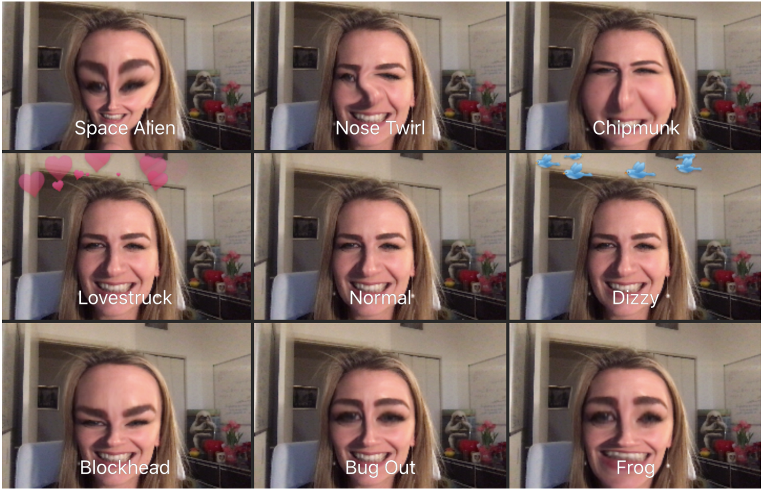
In practice, one replaces expectations with averages and restricts g and d to neural networks.

Generative Adversarial Networks

Generator against the Adversary

Generative Adversarial Networks

Generator against the Adversary



The Generator

Generate latent data $Z \sim \pi_Z$ for some distribution π_Z.

Filter Z through a deterministic mapping $g_{\boldsymbol{\beta}}$ such that

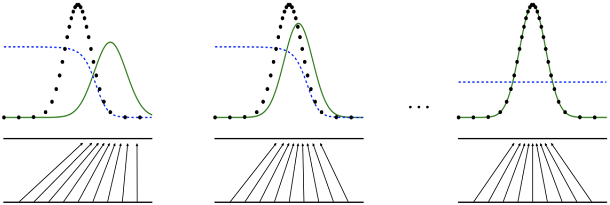
$$g_{\boldsymbol{\beta}}(Z) \sim P_\theta.$$

Based on the feedback from the Classifier, β is updated so that P_θ is closer and closer to P_0, where the game solution satisfies

$$g^*(Z) \sim P_0.$$

You can think of $\widetilde{X}_i^\theta = g_{\boldsymbol{\beta}}(Z_i) \sim P_\theta$ as the **'fake' data**.

Classifier against the Generator



The Classifier

The classification problem defined through

$$\max_{d \in \mathcal{D}} \left[\frac{1}{n} \sum_{i=1}^{n} \log d(X_i) + \frac{1}{m} \sum_{i=1}^{m} \log(1 - d(\widetilde{X}_i^{\theta})) \right], \tag{3}$$

where $d : \mathcal{X} \to (0, 1)$ (1 for 'real' and 0 for 'fake' data)

Oracle discriminator

$$d_{\theta}^{*}(X) := \frac{p_{\theta_0}(X)}{p_{\theta_0}(X) + p_{\theta}(X)}.$$

The optimal Generator leaves d_{θ}^{*} maximally **confused** (assigning score 1/2), which occurs when $p_{\theta} = p_{\theta_0}$.

Contrastive Learning for Bayesian Simulation

For iid data ($p_\theta^{(n)} = \prod_i p_\theta(X_i)$), we can rewrite the likelihood as

$$p_\theta^{(n)} = p_0^{(n)} \times \exp\left(\sum_{i=1}^{n} \log \frac{1 - d_\theta^*(X_i)}{d_\theta^*(X_i)}\right).$$

---

(1) Likelihood estimator? Deploy $\widehat{d}_{n,m}(\cdot)$ (e.g. neural network)

$$\widehat{p}_\theta^{(n)} = p_0^{(n)} \times \exp\left(\sum_{i=1}^{n} \log \frac{1 - \widehat{d}_{n,m}^{\theta}(X_i)}{\widehat{d}_{n,m}^{\theta}(X_i)}\right).$$

Kaji and Rockova (2021): *Metropolis Hastings via Classification*

---

(2) **KL estimator?**

$$\hat{K}(\boldsymbol{X}, \tilde{\boldsymbol{X}}^{\theta}) = \mathrm{P}_n \log \frac{\widehat{d}_{n,m}^{\theta}}{1 - \widehat{d}_{n,m}^{\theta}} = \frac{1}{n} \sum_{i=1}^{n} \log \frac{\widehat{d}_{n,m}^{\theta}(X_i)}{1 - \widehat{d}_{n,m}^{\theta}(X_i)}. \tag{4}$$

Wang, Kaji and Rockova (2022): *ABC via Classification*

☹ *Both require iid data and to run classification at every iteration!*

Conditional GANs

GANs can be trained to simulate from *conditional distributions*.

Consider a two-player minimax game

$$(g^*, d^*) = \arg\min_{g \in \mathcal{G}} \max_{d \in \mathcal{D}} D(g, d)$$

prescribed by

$$D(g, d) = E_{(X,\theta)\sim\pi(X,\theta)} \log d(X, \theta) + E_{X\sim\pi(X), Z\sim\pi_Z} \log[1 - d(X, g(Z, X))].$$

Uniformly on $\mathcal{X}$ and Θ (for 'flexible' $\mathcal{G}$ and $\mathcal{D}$), the solution (g^*, d^*) satisfies

$$\pi_{g^*}(\theta \mid X) = \frac{\pi(X, \theta)}{\pi(X)} = \pi(\theta \mid X)$$

and

$$d_g^*(X, \theta) = \frac{\pi(X, \theta)}{\pi(X, \theta) + \pi_g(\theta \mid X)\pi(X)} \quad \text{for any } g \in \mathcal{G}.$$

Bayesian GANs

The **B-GAN** Algorithm:

Simulate the ABC reference table $\{(\theta_i, X_i)\}_{i=1}^T$ from

$$\pi(\theta, X) = \pi(\theta) p_\theta^{(n)}(X^{(n)}) \quad \text{and} \quad \{Z_j\}_{j=1}^T \overset{\text{iid}}{\sim} \pi_Z(\cdot).$$

Choose function classes $\mathcal{G}$ and $\mathcal{D}$ (e.g. neural networks parametrized by β and ω).

Train the empirical version of Wasserstein conditional GANs

$$\widehat{\boldsymbol{\beta}}_T = \arg\min_{\boldsymbol{\beta}: g_{\boldsymbol{\beta}} \in \mathcal{G}} \left[\max_{\boldsymbol{\omega}: f_{\boldsymbol{\omega}} \in \mathcal{F}_W} \left| \sum_{j=1}^T f_{\boldsymbol{\omega}}\big(X_j, g_{\boldsymbol{\beta}}(Z_j, X_j)\big) - \sum_{j=1}^T f_{\boldsymbol{\omega}}(X_j, \theta_j) \right| \right]. \tag{5}$$

Simulate

$$\widetilde{\theta}_j = g_{\widehat{\boldsymbol{\beta}}_T}(Z_j, X_0) \quad \text{for} \quad Z_j \overset{\text{iid}}{\sim} \pi_Z. \tag{6}$$

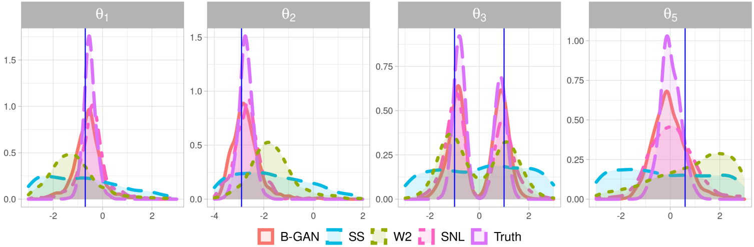
Observed data X_0 at evaluation stage, not training stage!

Toy Example

$X = (x_1, x_2, x_3, x_4)'$ consists of $n = 4$ two-dimensional Gaussian observations with $x_j \sim \mathcal{N}(\mu_{\boldsymbol{\theta}}, \Sigma_{\boldsymbol{\theta}})$ parametrized by $\boldsymbol{\theta} = (\theta_1, \theta_2, \theta_3, \theta_4, \theta_5)'$, where

$$\mu_{\boldsymbol{\theta}} = (\theta_1, \theta_2)' \quad \text{and} \quad \Sigma_{\boldsymbol{\theta}} = \begin{pmatrix} s_1^2 & \rho s_1 s_2 \\ \rho s_1 s_2 & s_2^2 \end{pmatrix}$$

with $s_1 = \theta_3^2$, $s_2 = \theta_4^2$ and $\rho = \tanh(\theta_5)$.



$T = 100K$, batchsize for SGD is 6400

Sequential Refinement

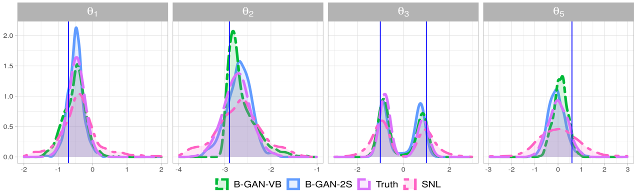
☹ **B-GAN is not trained on observed data!**

☹ We want $\pi(\theta \mid X)$ *at observed data $X = X_0$*, not at any X.

☹ The ABC reference table $\{(\theta_j, X_j)\}_{j=1}^T$ may not contain enough data points X_j in the vicinity of X_0 to train the simulator when the prior $\pi(\theta)$ is too vague.

☺ Use pilot simulator $g_{\widehat{\beta}_T}(Z, X_0)$ in (6) obtained under the original prior $\pi(\theta)$ as a proposal for the next round

☺ The 'wrong' prior can be corrected for by importance re-weighting with weights $r(\theta) = \pi(\theta)/\widetilde{\pi}(\theta)$.



Toy Example

Performance summary

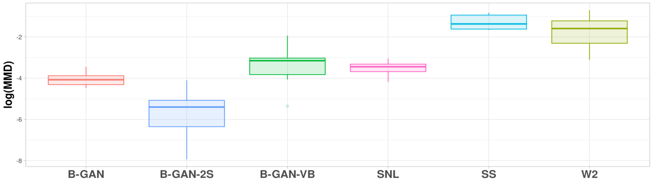


Figure: Maximum Mean Discrepancies (MMD, log scale) between the true posteriors and the approximated posteriors. The box-plots are computed from 10 repetitions.

TV Bounds: The Three Terms

(1) The ability of the *critic* to tell the true model apart from the approximating model

$$\mathcal{A}_1(\mathcal{F}, \widehat{\boldsymbol{\beta}}_T) \equiv \inf_{\boldsymbol{\omega}: f_{\boldsymbol{\omega}} \in \mathcal{F}} \left\| \log \frac{\pi(\theta \mid X)}{\pi_{\widehat{\boldsymbol{\beta}}_T}(\theta \mid X)} - f_{\omega}(X, \theta) \right\|_{\infty} \tag{7}$$

(2) The ability of the *generator* to approximate the average true posterior

$$\mathcal{A}_2(\mathcal{G}) \equiv \inf_{\boldsymbol{\beta}: g_{\boldsymbol{\beta}} \in \mathcal{G}} \left[E_X \left\| \log \frac{\pi_{\boldsymbol{\beta}}(\theta \mid X)}{\pi(\theta \mid X)} \right\|_{\infty} \right]^{1/2}, \tag{8}$$

(3) The *complexity* of the (generating and) critic function classes measured by the pseudo-dimension $Pdim(\cdot)$.

We denote with $\mathcal{H} = \left\{ h_{\boldsymbol{\omega},\boldsymbol{\beta}} : h_{\boldsymbol{\omega},\boldsymbol{\beta}}(Z, X) = f_{\boldsymbol{\omega}}\big(g_{\boldsymbol{\beta}}(Z, X), X\big) \right\}$ a structured composition of networks $f_{\boldsymbol{\omega}} \in \mathcal{F}$ and $g_{\boldsymbol{\beta}} \in \mathcal{G}$.

TV Bounds

Let $\widehat{\beta}_T$ be as in (5) where $\mathcal{F} = \{f : \|f\|_\infty \leq B\}$ for some $B > 0$.

Denote with E the expectation with respect to $\{(X_j, \theta_j)\}_{j=1}^T \overset{\text{iid}}{\sim} \pi(X, \theta)$ and $\{Z_j\}_{j=1}^T \overset{\text{iid}}{\sim} \pi_Z$ in the reference table.

Prior Concentration: Assume

$$\Pi[B_n(\theta_0; \epsilon)] \geq \mathrm{e}^{-C_2 n \epsilon^2} \quad \text{for some } C_2 > 0 \text{ and } \epsilon > 0. \tag{9}$$

Then for $T \geq Pdim(\mathcal{F}) \vee Pdim(\mathcal{H})$ we have for any $C > 0$

$$P_{\theta_0}^{(n)} \mathrm{E}\, d_{TV}^2 \left(\pi(\theta \mid X_0), \pi_{\widehat{\beta}_T}(\theta \mid X_0)\right) \leq \mathcal{C}_n^T(\widehat{\beta}_T, \epsilon, C),$$

where, for some $\widetilde{C} > 0$ and $Pmax \equiv Pdim(\mathcal{F}) \vee Pdim(\mathcal{H})$,

$$C_n^T(\widehat{\beta}_T, \epsilon, C) = \frac{1}{C^2 n \epsilon^2} + \frac{\mathrm{e}^{(1+C_2+C)n\epsilon^2}}{4} \left[2\mathcal{A}_1(\mathcal{F}, \widehat{\beta}_T) + \frac{B\,\mathcal{A}_2(\mathcal{G})}{\sqrt{2}} + 4\widetilde{C}\,B \sqrt{\frac{\log T \times Pmax}{T}} \right].$$

Implicit Variational Bayes

The goal of VB is to find a set of parameters $\boldsymbol{\beta}^*$ that maximize ELBO

$$\log \pi(X_0) \geq \mathcal{L}(\boldsymbol{\beta}) \equiv \int \log\left(\frac{\pi(X_0, \theta)}{q_{\boldsymbol{\beta}}(\theta \mid X_0)}\right) q_{\boldsymbol{\beta}}(\theta \mid X_0) \mathrm{d}\,\theta. \tag{10}$$

The tightness increases with expressiveness of $q_{\boldsymbol{\beta}}(\cdot)$, where the equality occurs when $q_{\boldsymbol{\beta}}(\theta \mid X_0) = \pi(\theta \mid X_0)$.

Implicit VB defines $q_{\boldsymbol{\beta}}(\theta \mid X_0)$ through a push-forward mapping $g_{\boldsymbol{\beta}}$.

We can re-write the ELBO in terms of Kullback-Leibler discrepancy

$$\mathcal{L}(\boldsymbol{\beta}) = -\mathrm{KL}\left(q_{\boldsymbol{\beta}}(\theta \mid X_0) \mid \pi(\theta \mid X_0)\right) + C$$

☹ We cannot evaluate the *conditional* density ratio in the ELBO

☺ We can estimate the ratio of *joint* distributions with a different conditional, given X, but the same marginal $\pi(X)$.

Adversarial Variational Bayes

☺ Joint LRT trick: define

$$\frac{d^*_{g_\beta}(X,\theta)}{1-d^*_{g_\beta}(X,\theta)} = \frac{\pi(X,\theta)}{q_\beta(\theta \mid X)\pi(X)}, \tag{11}$$

where $d^*_{g_\beta} : (\mathcal{X} \times \Theta) \to (0,1)$

The variational lower bound (10) can be re-written as

$$\mathcal{L}(\beta) \equiv E_{\theta \sim q_\beta(\theta \mid X_0)}\left[\operatorname{logit}\left(d^*_{g_\beta}(X_0,\theta)\right)\right] + C. \tag{12}$$

Note that $d^*_{g_\beta}(\theta, X)$ is a solution to

$$d^*_{g_\beta}(\theta, X) = \arg\max_{d \in \mathcal{D}} D(g_\beta, d). \tag{13}$$

Adversarial VB is a max-max game!

Given $\beta^{(t)}$: find $\psi^{(t+1)}$ such that

$$\psi^{(t+1)} = \arg\max_{\psi} D(g_{\beta^{(t)}}, d_\psi).$$

Given $\psi^{(t+1)}$: find $\beta^{(t+1)}$

$$\beta^{(t+1)} = \arg\max_{\beta} \mathrm{E}_{\theta \sim q_\beta(\theta \mid X_0)}\left[\operatorname{logit}\left(d_{\psi^{(t+1)}}(\theta, X_0)\right)\right]$$

...and there are Wasserstein versions

Instead of KL, we can minimize Wasserstein distance between $\pi(\theta \mid X_0)$ and $q_{\boldsymbol{\beta}}(\theta \mid X_0)$:

$$\boldsymbol{\beta}^* = \arg \min_{\boldsymbol{\beta}: g_{\boldsymbol{\beta}} \in \mathcal{G}} \sup_{f_{\boldsymbol{\omega}} \in \mathcal{F}_W} \left| E_{\theta \sim q_{\boldsymbol{\beta}}(\theta \mid X_0)} \left(\frac{\pi(\theta \mid X_0)}{q_{\boldsymbol{\beta}}(\theta \mid X_0)} - 1 \right) f_{\boldsymbol{\omega}}(\theta) \right|, \qquad (14)$$

Using the ABC reference table $\{(\theta_j, X_j)\}_{j=1}^T \overset{\text{iid}}{\sim} \pi(\theta, X), \{Z_j\}_{j=1}^T \overset{\text{iid}}{\sim} \pi_Z(\cdot)$,

- update $\boldsymbol{\omega}^{(t+1)}$,given $\boldsymbol{\beta}^{(t)}$,

$$\boldsymbol{\omega}^{(t+1)} = \arg \max_{\boldsymbol{\omega}: f_{\boldsymbol{\omega}} \in \mathcal{F}} \left[\sum_{j=1}^{T} f_{\boldsymbol{\omega}}(X_j, g_{\boldsymbol{\beta}^{(t)}}(Z_j, X_j)) - \sum_{j=1}^{T} f_{\boldsymbol{\omega}}(X_j, \theta_j) \right] \qquad (15)$$

- update $\boldsymbol{\beta}^{(t+1)}$, given $\boldsymbol{\omega}^{(t+1)}$,

$$\boldsymbol{\beta}^{(t+1)} = \arg \min_{\boldsymbol{\beta}: g_{\boldsymbol{\beta}} \in \mathcal{G}} \left[\sum_{j=1}^{T} f_{\boldsymbol{\omega}^{(t+1)}}(X_0, g_{\boldsymbol{\beta}}(Z_j, X_0)) + C \right], \qquad (16)$$

where C does not depend on $\boldsymbol{\beta}$, given the most recent update $\boldsymbol{\omega}^{(t+1)}$.

Lotka-Volterra Model

The Lotka-Volterra (LV) model describes population evolutions in ecosystems where predators interact with prey.

The model is deterministically prescribed via a system of first-order non-linear ODEs with four parameters $\theta = (\theta_1, \theta_2, \theta_3, \theta_4)'$ controlling

(1) the rate $r_1^t = \theta_1 X_t Y_t$ of a predator being born,

(2) the rate $r_2^t = \theta_2 X_t$ of a predator dying,

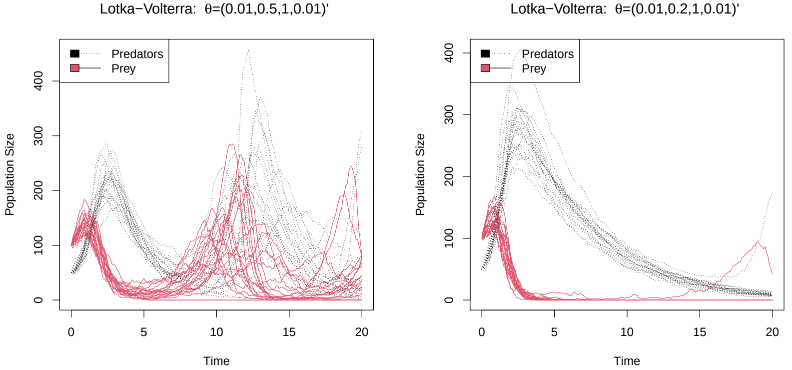
(3) the rate $r_3^t = \theta_3 Y_t$ a prey being born and

(4) the rate $r_4^t = \theta_4 X_t Y_t$ a prey dying.

Despite easy to sample from (using the Gillespie algorithm), the likelihood for this model is unavailable which makes this model a natural candidate for ABC

The pseudo-marginal approach far from straightforward, if at all possible ☹

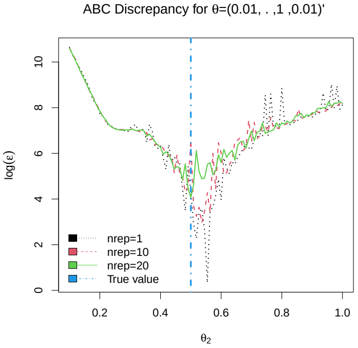
Lotka-Volterra: A Closer Look

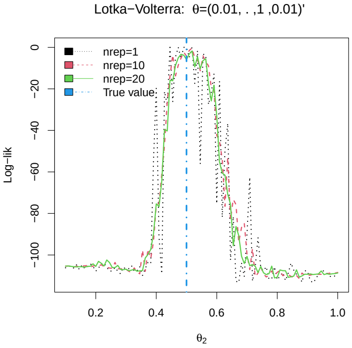


Simulation is started at $X_0 = 50$ and $Y_0 = 100$ simulated over 20 time units and recorded observations every 0.1 time units, resulting in a series of $T = 201$ observations each.

True values $\theta^0 = (0.01, 0.5, 1, 0.01)$

Prepping for ABC





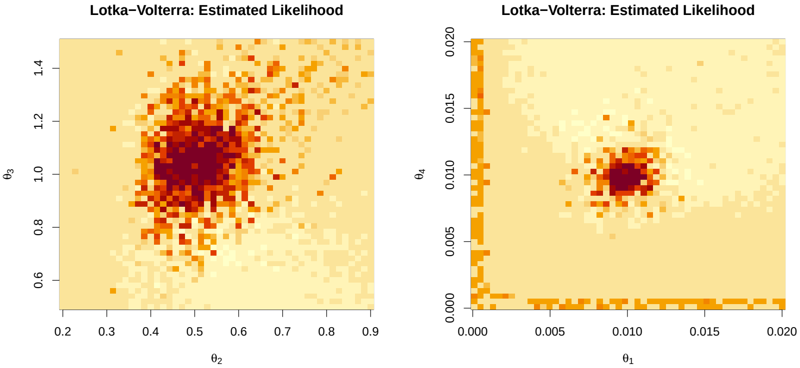
LEFT:

ABC tolerance (based on summary statistics)

RIGHT:

Classification-based log-lik estimator running LASSO (`glmnet`)

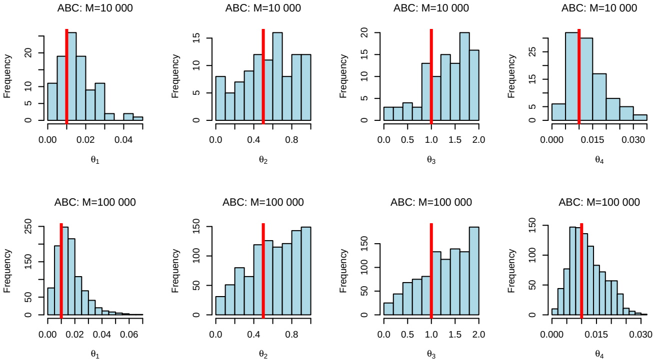
Likelihood is Spiky!



$\rightsquigarrow$ ABC will need a *very* informative prior

ABC Results

Uniform Prior: on $[0, 0.1] \times [0, 1] \times [0, 2] \times [0, 0.1]$

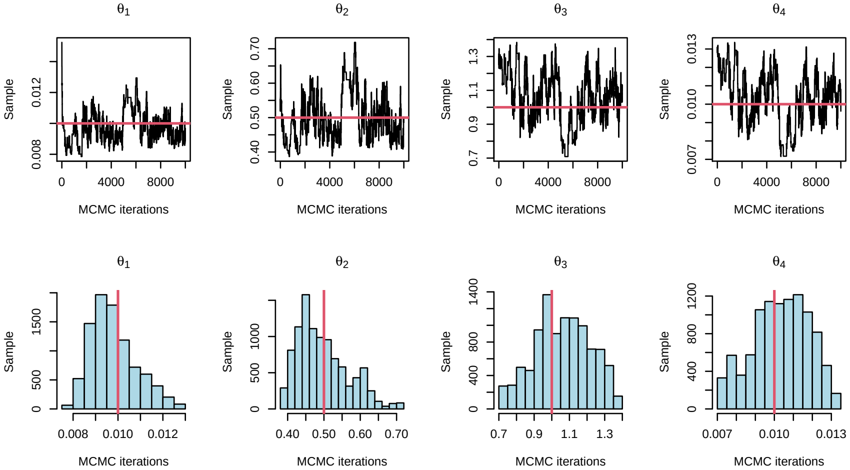


Upper panel: $M = 10\,000$ and $r = 100$

Lower panel: $M = 100\,000$ and $r = 1\,000$.

MHC (Kaji and Rockova (2021) Results

Initialized at posterior mean from a pilot ABC run.



MCMC trace plots (with $M = 10\,000$) and histograms (without a burnin $1\,000$)

MHC: Posterior Summary Statistics

	$\theta_1^0 = 0.01$			$\theta_2^0 = 0.5$			$\theta_3 = 1$			$\theta_4 = 0.01$		
Method	$\bar{\theta}$	l	u	$\bar{\theta}$	l	u	$\bar{\theta}$	l	u	$\bar{\theta}$	l	u
ABC1	0.015	0.003	0.038	0.554	0.037	0.985	1.315	0.189	1.955	0.012	0.004	0.029
ABC2	0.016	0.003	0.042	0.604	0.087	0.980	1.259	0.205	1.971	0.013	0.003	0.024
MHC	0.01	0.008	0.014	0.531	0.41	0.685	1.029	0.791	1.301	0.010	0.007	0.014

ABC1: $M = 10\,000$ and $r = 100$ (accepted samples)

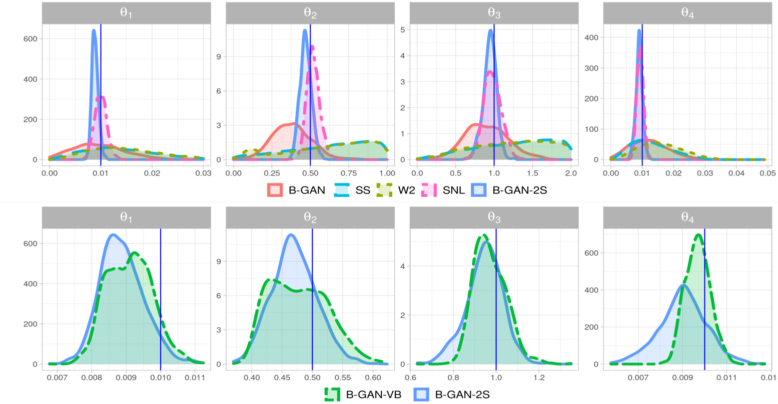
ABC2: $M = 100\,000$ and $r = 1\,000$ (accepted samples)

MHC: $M = 10\,000$ with burnin 1 000

$\bar{\theta}$ denotes posterior mean, l and u denote the lower and upper boundaries of 95% credible intervals.

What about $n = 1$?

We compare B-GAN with Sequential Neural Likelihood (SNL), W2 ABC, and Summary Statistics ABC



Sequential refinement and VB refinement work well.

B-GAN Performance

Summary statistics of the approximated posteriors (averaged over 10 repetitions).

	$\theta_1 = 0.01$		$\theta_2 = 0.5$		$\theta_3 = 1.0$		$\theta_4 = 0.01$	
	bias	CI width	bias	CI width	bias	CI width	bias	CI width
(scale)	$(\times 10^{-3})$	$(\times 10^{-2})$	$(\times 10^{-1})$				$(\times 10^{-2})$	$(\times 10^{-2})$
B-GAN	4.15	1.89	1.09	0.45	0.24	1.00	0.49	2.18
B-GAN-2S	**0.70**	**0.21 (0.9)**	0.42	**0.10 (0.7)**	**0.11**	0.33 (0.9)	0.13	0.34 (0.8)
B-GAN-VB	1.02	0.25 (0.7)	**0.38**	0.11 (0.9)	**0.11**	**0.29 (0.8)**	**0.12**	**0.29 (0.7)**
SNL	1.05	0.44	0.45	0.17	0.13	0.48	0.15	0.52
SS	9.58	3.80	2.49	0.91	0.49	1.76	0.68	2.72
W2	10.99	4.02 (0.9)	2.42	0.84	0.47	1.73	0.79	2.82

Bold fonts mark the best model of each column. The coverage of the 95% credible intervals are 1 unless otherwise noted in the parentheses.

Wang, Y. and Rockova, V. (2022) *Adversarial Bayesian Simulation*

Thank you!