Enabling Hamiltonian Monte Carlo for large datasets

Matias Quiroz[1,2]

[1]School of Mathematical and Physical Sciences, University of Technology Sydney

[2]ARC Centre of Excellence for Mathematical & Statistical Frontiers

June 2019

What this talk is about

- **Markov Chain Monte Carlo** (MCMC) and **Sequential Monte Carlo** (SMC) for **"Big Data"**.
- **Collaborators** (alphabetical order):
  - *Khue-Dung Dang* (University of New South Wales).
  - *David Gunawan* (University of New South Wales).
  - *Robert Kohn* (University of New South Wales).
  - *Minh-Ngoc Tran* (University of Sydney).
  - *Mattias Villani* (Linköping University and Stockholm University).
- **Papers**:
  - [Dang et al., 2019]:
    "Hamiltonian Monte Carlo with energy conserving subsampling".
  - [Gunawan et al., 2019]:
    "Subsampling sequential Monte Carlo for static Bayesian models".
- Slides uploaded on **`www.matiasquiroz.com/news`**.

What is Big Data?

What is Big Data?

- **Big Data** is a **buzzword**.

What is Big Data?

- **Big Data** is a **buzzword**.
- My **"Big Data"** scenarios (*n* observations, *d* variables / parameters):

What is Big Data?

- **Big Data** is a **buzzword**.
- My **"Big Data"** scenarios (n observations, d variables / parameters):
  [**NOT** Velocity, Volume, Value, Variety, and Veracity]

What is Big Data?

- **Big Data** is a **buzzword**.
- My **"Big Data"** scenarios (n observations, d variables / parameters):
  [**NOT** Velocity, Volume, Value, Variety, and Veracity]
  1. **Huge** n, **small** d ("Tall and skinny" data).
  2. **Huge** n, **moderately large** d ("Tall and not-so-skinny" data).
  3. **Large** n, **large** d ("Large" data). **Big model**.
  4. **Huge** n, **huge** d (" Humongous" data). **Huge model**.
- 1.-2.. Posterior simulation methods (**MCMC**, **SMC**) to scale these.
- 3.-4.. **Variational inference** to scale these (not this talk).

Motivation and our approach

- **MCMC** and **SMC** to compute the expectation of $f(\theta)$ w.r.t.

$$\pi(\theta|\mathbf{y}) = \frac{p(\mathbf{y}|\theta)p(\theta)}{p(\mathbf{y})}.$$

- Markov chain Monte Carlo **MCMC** - Bayesian workhorse for 3 decades.
- Sequential Monte Carlo **SMC** - Alternative to **MCMC** that is parallelizable.
- **MCMC is often slow**
  - Need to evaluate the likelihood function in each iteration.
  - Many iterations (sampling algorithm)...
  - ... especially if the Markov chain moves slowly
- Similar obstacles for **SMC**.
- **Key idea**: **Subsampling approach** to deal with large n: **estimate the likelihood** from a subsample. Faster!

MCMC: The Metropolis-Hastings algorithm

- A simulation algorithm to sample $\pi(\theta|\mathbf{y}) \propto p(\mathbf{y}|\theta)p(\theta)$.

- Initialize $\theta^{(0)}$ and iterate for $i = 1, 2, ..., N$
  1. Sample $\theta_p \sim q\left(\cdot|\theta^{(i-1)}\right)$ and
  2. Compute the **acceptance probability**

$$\alpha = \min\left(1, \frac{p\left(\mathbf{y}|\theta_p\right)p(\theta_p)}{p\left(\mathbf{y}|\theta^{(i-1)}\right)p(\theta^{(i-1)})}\frac{q\left(\theta^{(i-1)}|\theta_p\right)}{q\left(\theta_p|\theta^{(i-1)}\right)}\right)$$

  3. With probability α set $\theta^{(i)} = \theta_p$ and $\theta^{(i)} = \theta^{(i-1)}$ otherwise.

- Efficiency depends on **the proposal** $q()$.
- **Random-walk**: $q(\cdot|\theta^{(i-1)}) = \mathcal{N}(\cdot|\theta^{(i-1)}, \Sigma)$
- $\Sigma = O(d^{-1})$, $d = \dim(\theta)$, for optimality $\implies$ moves **slowly** for large d.

- An efficient MH proposal that **maintains** $\alpha \approx 1$ + **moves** θ **far**.
- Samples on an **augmented space** $\pi(\theta, \vec{p}) \propto \exp\left(-\mathcal{H}(\theta, \vec{p})\right)$,

$$\mathcal{H}(\theta, \vec{p}) = -\log \pi(\theta|\mathbf{y}) + K(\vec{p}), \quad \mathcal{K}(\vec{p}) = \frac{1}{2}\vec{p}^\top M^{-1} p.$$

- Move around $\mathcal{H}(\theta, \vec{p})$ using **Hamiltonian Dynamics** (HD).

$$\frac{d\theta_l}{dt} = \frac{\partial \mathcal{H}(\theta, \vec{p})}{\partial \vec{p}_l}, \quad \frac{d\vec{p}_l}{dt} = -\frac{\partial \mathcal{H}(\theta, \vec{p})}{\partial \theta_l}, \quad l = 1, \ldots, d,$$

- Some **nice properties** of $\mathcal{H}$ and its dynamics
  1. **Reversibility**: The mapping from t to $t+s$ is one-to-one (inverse exist).
  2. **Energy conservation**: $\frac{d}{dt}\mathcal{H}(\theta, \vec{p}) = 0$
  3. **Volume preservation**: The mapping preserves volume.
- **Idea of Hamiltonian Monte Carlo**:
  Use **HD** to construct proposals for MH sampling of $\pi(\theta, \vec{p})$! Marginal density for θ is $\pi(\theta|\mathbf{y})$.

The Metropolis-Hastings algorithm with HD proposal

- **Recall**: wish to sample $\pi(\theta, \vec{p}) \propto \exp\left(-\mathcal{H}(\theta, \vec{p})\right)$.

- Initialize $\theta^{(0)}, \vec{p}^{(0)}$ and iterate for $i = 1, 2, ..., N$
  1. Use **HD** with integration time L to obtain $\theta_p, \vec{p}_p$.
  2. Compute the **acceptance probability**

$$\alpha = \min\left(1, \frac{\exp\left(-\mathcal{H}(\theta_p, \vec{p}_p)\right)}{\exp\left(-\mathcal{H}(\theta^{(i-1)}, \vec{p}^{(i-1)})\right)}\right)$$

  3. With probability α set $\theta^{(i)}, \vec{p}^{(i)} = \theta_p, \vec{p}_p$, otherwise $\theta^{(i)}, \vec{p}^{(i)} = \theta^{(i-1)}, \vec{p}^{(i-1)}$.

The Metropolis-Hastings algorithm with HD proposal

- **Recall**: wish to sample $\pi(\theta, \vec{p}) \propto \exp\left(-\mathcal{H}(\theta, \vec{p})\right)$.

- Initialize $\theta^{(0)}, \vec{p}^{(0)}$ and iterate for $i = 1, 2, \ldots, N$
  1. Use **HD** with integration time L to obtain $\theta_p, \vec{p}_p$.
  2. Compute the **acceptance probability**
  $$\alpha = \min\left(1, \frac{\exp\left(-\mathcal{H}(\theta_p, \vec{p}_p)\right)}{\exp\left(-\mathcal{H}(\theta^{(i-1)}, \vec{p}^{(i-1)})\right)}\right)$$
  3. With probability α set $\theta^{(i)}, \vec{p}^{(i)} = \theta_p, \vec{p}_p$, otherwise $\theta^{(i)}, \vec{p}^{(i)} = \theta^{(i-1)}, \vec{p}^{(i-1)}$.

- **Remark 1**: The proposal ratio disappears due to **reversibility**.

The Metropolis-Hastings algorithm with HD proposal

- **Recall**: wish to sample $\pi(\theta, \vec{p}) \propto \exp\left(-\mathcal{H}(\theta, \vec{p})\right)$.

- Initialize $\theta^{(0)}, \vec{p}^{(0)}$ and iterate for $i = 1, 2, ..., N$
  1. Use **HD** with integration time L to obtain $\theta_p, \vec{p}_p$.
  2. Compute the **acceptance probability**

$$\alpha = \min\left(1, \frac{\exp\left(-\mathcal{H}(\theta_p, \vec{p}_p)\right)}{\exp\left(-\mathcal{H}(\theta^{(i-1)}, \vec{p}^{(i-1)})\right)}\right)$$

  3. With probability α set $\theta^{(i)}, \vec{p}^{(i)} = \theta_p, \vec{p}_p$, otherwise $\theta^{(i)}, \vec{p}^{(i)} = \theta^{(i-1)}, \vec{p}^{(i-1)}$.

- **Remark 1**: The proposal ratio disappears due to **reversibility**.

- **Remark 2**: No determinant of the Jacobian for the HD mapping due to **volume preservation**.

The Metropolis-Hastings algorithm with HD proposal

- **Recall**: wish to sample $\pi(\theta, \vec{p}) \propto \exp\left(-\mathcal{H}(\theta, \vec{p})\right)$.

- Initialize $\theta^{(0)}, \vec{p}^{(0)}$ and iterate for $i = 1, 2, ..., N$
  1. Use **HD** with integration time L to obtain $\theta_p, \vec{p}_p$.
  2. Compute the **acceptance probability**
  $$\alpha = \min\left(1, \frac{\exp\left(-\mathcal{H}(\theta_p, \vec{p}_p)\right)}{\exp\left(-\mathcal{H}(\theta^{(i-1)}, \vec{p}^{(i-1)})\right)}\right)$$
  3. With probability α set $\theta^{(i)}, \vec{p}^{(i)} = \theta_p, \vec{p}_p$, otherwise $\theta^{(i)}, \vec{p}^{(i)} = \theta^{(i-1)}, \vec{p}^{(i-1)}$.

- **Remark 1**: The proposal ratio disappears due to **reversibility**.
- **Remark 2**: No determinant of the Jacobian for the HD mapping due to **volume preservation**.
- **Remark 3**: $\mathcal{H}(\theta_p, \vec{p}_p) = \mathcal{H}(\theta^{(i-1)}, \vec{p}^{(i-1)})$ due to **energy conservation**!...

The Metropolis-Hastings algorithm with HD proposal

- **Recall**: wish to sample $\pi(\theta, \vec{p}) \propto \exp\left(-\mathcal{H}(\theta, \vec{p})\right)$.

- Initialize $\theta^{(0)}, \vec{p}^{(0)}$ and iterate for $i = 1, 2, ..., N$
  1. Use **HD** with integration time L to obtain $\theta_p, \vec{p}_p$.
  2. Compute the **acceptance probability**

$$\alpha = \min\left(1, \frac{\exp\left(-\mathcal{H}(\theta_p, \vec{p}_p)\right)}{\exp\left(-\mathcal{H}(\theta^{(i-1)}, \vec{p}^{(i-1)})\right)}\right)$$

  3. With probability α set $\theta^{(i)}, \vec{p}^{(i)} = \theta_p, \vec{p}_p$, otherwise $\theta^{(i)}, \vec{p}^{(i)} = \theta^{(i-1)}, \vec{p}^{(i-1)}$.

- **Remark 1**: The proposal ratio disappears due to **reversibility**.
- **Remark 2**: No determinant of the Jacobian for the HD mapping due to **volume preservation**.
- **Remark 3**: $\mathcal{H}(\theta_p, \vec{p}_p) = \mathcal{H}(\theta^{(i-1)}, \vec{p}^{(i-1)})$ due to **energy conservation**!...
- .... thus $\alpha = 1$, **always**! If L is large enough, distant moves and $\alpha = 1$!

Hamiltonian Monte Carlo in practice

- This all **sounds great**...

Hamiltonian Monte Carlo in practice

- This all **sounds great**...
- ... **but**...

Hamiltonian Monte Carlo in practice

- This all **sounds great**...
- ... **but**...
  1. We cannot solve HD except for **toy models**.

Hamiltonian Monte Carlo in practice

- This all **sounds great**...
- ... **but**...
  1. We cannot solve HD except for **toy models**.
  2. **No free lunch** - comes at increased cost due to $\nabla_\theta \log p(\theta|\mathbf{y})$...

Hamiltonian Monte Carlo in practice

- This all **sounds great**...
- ... **but**...
  1. We cannot solve HD except for **toy models**.
  2. **No free lunch** - comes at increased cost due to $\nabla_\theta \log p(\theta|\mathbf{y})$...
  3. ... particularly **for large** n: $\nabla_\theta \log p(\theta|\mathbf{y}) = \nabla_\theta \log p(\theta) + \sum_{i=1}^{n} \nabla_\theta \log p(\mathbf{y}_i|\theta)$.

Hamiltonian Monte Carlo in practice

- This all **sounds great**...
- ... **but**...
  1. We cannot solve HD except for **toy models**.
  2. **No free lunch** - comes at increased cost due to $\nabla_\theta \log p(\theta|\mathbf{y})$...
  3. ... particularly **for large** n: $\nabla_\theta \log p(\theta|\mathbf{y}) = \nabla_\theta \log p(\theta) + \sum_{i=1}^{n} \nabla_\theta \log p(\mathbf{y}_i|\theta)$.
- Solving 1.: **Symplectic integrators**. Conserve energy approximately, $\alpha \approx 1$.
- Solution to 2. + 3.:
  Data subsampling - estimate the gradient unbiasedly from a subsample of observations?
- **Naive subsampling** does not conserve energy [Betancourt, 2015].
  Acceptance probability drops to zero quickly as d increases.
- [Chen et al., 2014] **modifies the dynamic** to fix this. **Does not** have an accept/reject step. Step-size of the discretization of the dynamics **needs to be small**.
- **Our contribution**: How to subsample such that the energy is conserved?

Energy conserving subsampling

- Let $\mathbf{u}$ be the **observation indices** to sample. $|\mathbf{u}| = m$.
- Let $\widehat{\mathcal{H}}(\theta, \vec{p}) = -\log \widehat{L}(\theta, \mathbf{u}) - \log p(\theta) + \mathcal{K}(\vec{p})$, where $\widehat{L}(\theta, \mathbf{u})$ is an estimator of $L(\theta) = p(\mathbf{y}|\theta)$.
- Consider an **augmented target**: $\pi(\theta, \vec{p}, \mathbf{u}) \propto \exp\left(-\widehat{\mathcal{H}}(\theta, \vec{p})\right) p(\mathbf{u})$.
- Which $\widehat{L}(\theta, \mathbf{u})$ to use? Follow [Quiroz et al., 2018, JASA].
- Estimate $L(\theta) = \exp\left(\ell(\mathbf{y}|\theta)\right)$ by **bias-correcting** $\exp\left(\widehat{\ell}(\mathbf{y}|\theta, \mathbf{u})\right)$, where

$$\mathrm{E}_{\mathbf{u}}\left(\widehat{\ell}(\mathbf{y}|\theta, \mathbf{u})\right) = \ell(\mathbf{y}|\theta) = \sum_{i=1}^{n} \ell(y_i|\theta) = \left[\log p(\mathbf{y}|\theta)\right].$$

- **Difference estimator** with **control variates** $q_i(\theta) \approx \ell(y_i|\theta)$

$$\widehat{\ell}(\mathbf{y}|\theta, \mathbf{u}) = \sum_{i=1}^{n} q_i(\theta) + \frac{n}{m}\sum_{i\in\mathbf{u}} d_i(\theta),\ d_i(\theta) = \ell(y_i|\theta) - q_i(\theta).$$

- **Can derive** $\nabla_\theta \log \widehat{L}(\theta, \mathbf{u})$ to use in **our HD**.

Energy conserving subsampling, cont.

- **Sample** $\pi(\theta, \vec{p}, \mathbf{u})$ by
  1. $\theta, \vec{p}|\mathbf{u}$ - HMC step with energy $\widehat{\mathcal{H}}$ computed **from a subsample** $\mathbf{u}$
  2. $\mathbf{u}|\theta, \vec{p}$ - Block pseudo-marginal step given the **parameters and momentum**.
- Marginalizing $\vec{p}, \mathbf{u}$ gives the **perturbed posterior** in [Quiroz et al., 2018].
- The **perturbed posterior** has TV-norm error of $\mathcal{O}(n^{-1}m^{-2})$.
- For example, if $m = \mathcal{O}(n^{1/2})$ then **the error** is $\mathcal{O}(n^{-2})$.
- **Before** HMC:
  - $n = 4.7$ millons data points.
  - **logistic regression** with $d = 9$ parameters.
- **After** HMC:
  - $n = 4.7$ millons data points.
  - **additive splines logistic regression** with $d = 81$ parameters (10 knots for each of 8 covariates + intercept)
- **Compare against** [Welling and Teh, 2011, Chen et al., 2014, Baker et al., 2017].

Speed-ups bankruptcy example

	# evaluations	RCT	IF
HMC	110601×10^6	7691.8	2.20
$\text{HMC-ECS}_{\text{P}}$	14.02×10^6	1	2.20
SG-HMC_1	120×10^6	9.49	2.42
SG-HMC_2	14×10^6	100.29	226.75
SGLD	11×10^6	230	649.0

Table 1 : Comparison between HMC (full data), HMC-ECS [Dang et al., 2019], Stochastic gradient HMC (SG-HMC) [Chen et al., 2014] and Stochastic gradient Langevin Dynamics (SGLD) [Welling and Teh, 2011]. RCT is relative to our method $\text{HMC-ECS}_{\text{P}}$.

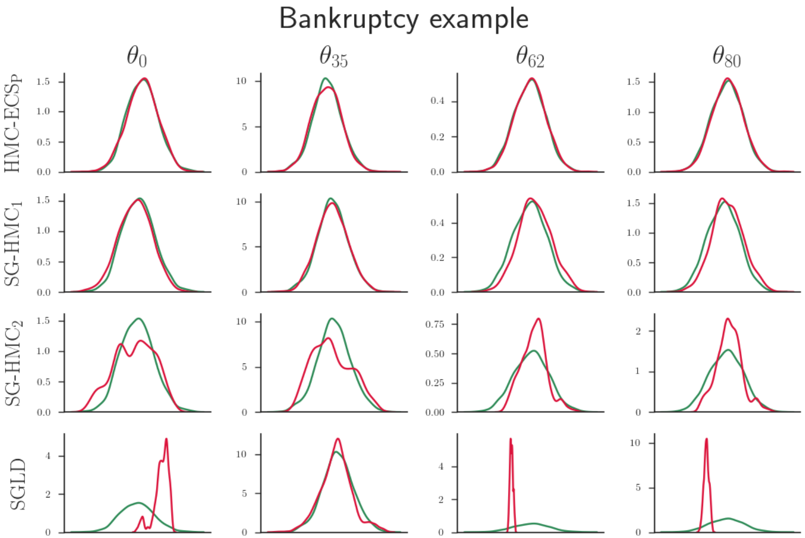
$$\text{IF} := 1 + 2\sum_{l=1}^{\infty} \rho_l$$

ρ_l is the l-lag autocorrelation of the MCMC chain

$$\text{RCT}_{\mathcal{A}} := \frac{\text{CT}_{\mathcal{A}}}{\text{CT}_{\text{HMC-ECS}}}.$$

for $\text{CT}_{\mathcal{A}} := \text{IF}_{\mathcal{A}} \times$ Total number of density and gradient evaluations.

Accuracy bankruptcy example



Concluding the Hamiltonian part of the talk

- Presented **Hamiltonian Monte Carlo** (HMC). Maybe useful for you? HMC for DSGE models?
- An approach for **energy conserving subsampling** Hamiltonian Monte Carlo.
- **Game changer**: Allows us to increase $d = 9$ [Quiroz et al., 2018] to $d = 81$.
- Unlike [Chen et al., 2014], our bias is **not a function** of the step-size of the discretization.
- Outperforms popular **Machine Learning** approaches.

Sequential Monte Carlo (SMC)

- An **alternative** algorithm to **compute expectations** wrt. $\pi(\theta|\mathbf{y})$.
- Provides an estimate of $p(\mathbf{y}) = \int p(\mathbf{y}|\theta)p(\theta)d\theta$ as **a by-product**.
- An **alternative** algorithm to **compute expectations** wrt. $\pi(\theta|\mathbf{y})$.
- **Basic idea of SMC**:

  (i) Samples from an **initial distribution** and (ii) **reweights the samples** by bringing in **information about** θ provided by the data **sequentially**.
- Initial distribution in a **Bayesian context**: The prior $p(\theta)$.
- How to bring in information **sequentially**?
  1. **Data annealing**: $p(\mathbf{y}_{1:T}|\theta)$ for a sequence of Ts with the last $= n$.
  2. **Likelihood annealing**: $p(\mathbf{y}|\theta)^{a_p}$, for $a_0 = 0 < a_1 < \cdots < a_P = 1$.
- How to reweight samples? **Importance sampling**.
- Our paper focuses on **likelihood annealing**.

Sequential Monte Carlo (SMC)

- An **alternative** algorithm to **compute expectations** wrt. $\pi(\theta|\mathbf{y})$.
- Provides an estimate of $p(\mathbf{y}) = \int p(\mathbf{y}|\theta)p(\theta)d\theta$ as **a by-product**.
- An **alternative** algorithm to **compute expectations** wrt. $\pi(\theta|\mathbf{y})$.
- **Basic idea of SMC**:

  (i) Samples from an **initial distribution** and (ii) **reweights the samples** by bringing in **information about** θ provided by the data **sequentially**.
- Initial distribution in a **Bayesian context**: The prior $p(\theta)$.
- How to bring in information **sequentially**?
  1. **Data annealing**: $p(\mathbf{y}_{1:T}|\theta)$ for a sequence of Ts with the last $= n$.
  2. **Likelihood annealing**: $p(\mathbf{y}|\theta)^{a_p}$, for $a_0 = 0 < a_1 < \cdots < a_P = 1$.
- How to reweight samples? **Importance sampling**.
- Our paper focuses on **likelihood annealing**.
- Simple idea: Estimate the **annealed likelihood** by **data subsampling**.

Subsampling Sequential Monte Carlo (SMC)

- At each p, SMC obtains a **weighted sample** from: $\pi_p(\theta) \propto p(\mathbf{y}|\theta)^{a_p} p(\theta)$.
- **Subsampling SMC**: at each p, target $\pi_p(\theta, \mathbf{u}) \propto \widehat{L}_p(\theta, \mathbf{u}) p(\theta) p(\mathbf{u})$.
- $\widehat{L}_p(\theta, \mathbf{u})$ is an (approximately) **unbiased** estimator of $p(\mathbf{y}|\theta)^{a_p}$.
- **Bias-correct** the exponent of the unbiased estimator $a_p \widehat{\ell}(\mathbf{y}|\theta, \mathbf{u})$ of $a_p \ell(\mathbf{y}|\theta)$
- $\widehat{L}_p(\theta, \mathbf{u}) = \exp\left(a_p \widehat{\ell}(\mathbf{y}|\theta, \mathbf{u}) - \text{bias-correction}\right)$
- Implement **sequential Monte Carlo** with this estimator.
- At sequence P ($a_P = 1$) we get the **target** in [Quiroz et al., 2018].
- TV-norm error of $\mathcal{O}(n^{-1} m^{-2})$. Approximate, but **very accurate** for large n.

- Initial **particle cloud** and **weights** $\left\{\theta_{1:M}^{(0)}, \mathbf{u}_{1:M}^{(0)}, W_{1:M}^{(0)}\right\}$.
- Obtained by generating the $\left\{\theta_{1:M}^{(0)}, \mathbf{u}_{1:M}^{(0)}\right\}$ from $p\left(\theta\right)$ and $p\left(\mathbf{u}\right)$, and assigning **equal weights** $W_{1:M}^{(0)} = 1/M$,
- The **weighted particles** $\left\{\theta_{1:M}^{(p-1)}, \mathbf{u}_{1:M}^{(p-1)}, W_{1:M}^{(p-1)}\right\}$ at the $(p-1)$st stage are a sample from $\pi_{p-1}\left(\theta, \mathbf{u}\right)$.
- **Propagated to next** $\pi_p(\theta, \mathbf{u})$ by updating the weights $W_{1:M}^{(p)} = w_{1:M}^{(p)} / \sum_{i=1}^{M} w_i^{(p)}$, where

$$w_i^{(p)} = W_i^{(p-1)} \exp\left(\left(a_p - a_{p-1}\right)\widehat{\ell}(\mathbf{y}|\theta_i^{(p-1)}, \mathbf{u}_i^{(p-1)}) - \text{BC}\right).$$

- At $p = P$, $\left\{\theta_{1:M}^{(P)}, \mathbf{u}_{1:M}^{(P)}, W_{1:M}^{(P)}\right\}$ is from $\pi\left(\theta, \mathbf{u}\right)$.

Subsampling Sequential Monte Carlo (SMC), cont.

- **SMC problem**: The weights will concentrate on **a few particles**.
- **Effective Sample Size** at stage p $\mathrm{ESS}_p = \left(\sum_{i=1}^{M}\left(W_i^{(p)}\right)^2\right)^{-1}$.
- **Solution**: At stage p, obtain new particles by resampling particles at stage $p-1$ with probability $W_{1:M}^{(p)}$. Reset $W_i^{(p)} = 1/M$.
- ... Gives the next problem... particles with **large weights** are **duplicated**.
- Want to **move** the particles **without distorting** their distribution.
- Apply a $\pi_p(\theta, \mathbf{u})$-**invariant Markov kernel** to each of the particles.

Subsampling Sequential Monte Carlo (SMC), cont.

- **SMC problem**: The weights will concentrate on **a few particles**.
- **Effective Sample Size** at stage p $\mathrm{ESS}_p = \left(\sum_{i=1}^{M}\left(W_i^{(p)}\right)^2\right)^{-1}$.
- **Solution**: At stage p, obtain new particles by resampling particles at stage $p-1$ with probability $W_{1:M}^{(p)}$. Reset $W_i^{(p)} = 1/M$.
- ... Gives the next problem... particles with **large weights** are **duplicated**.
- Want to **move** the particles **without distorting** their distribution.
- Apply a $\pi_p(\theta, \mathbf{u})$-**invariant Markov kernel** to each of the particles.
- Any candidate that can:
  (i) make distant moves + **(ii) maintain a high acceptance probability**?

Subsampling Sequential Monte Carlo (SMC), cont.

- **SMC problem**: The weights will concentrate on **a few particles**.
- **Effective Sample Size** at stage p $\mathrm{ESS}_p = \left(\sum_{i=1}^{M}\left(W_i^{(p)}\right)^2\right)^{-1}$.
- **Solution**: At stage p, obtain new particles by resampling particles at stage $p-1$ with probability $W_{1:M}^{(p)}$. Reset $W_i^{(p)} = 1/M$.
- ... Gives the next problem... particles with **large weights** are **duplicated**.
- Want to **move** the particles **without distorting** their distribution.
- Apply a $\pi_p(\theta, \mathbf{u})$-**invariant Markov kernel** to each of the particles.
- Any candidate that can:
  (i) **make distant moves** + **(ii)** **maintain a high acceptance probability**?
- HMC (on the tempered posterior) with **energy conserving subsampling**!

SMC results: CPU and marginal likelihood estimation

	log marginal likelihood	CPU time (hrs)	P	R
Student-t regression				
$(n = 500,000, m = 1,200)$				
Full data SMC	-815,775.82 (0.39)	5.92	126	4
Subsampling SMC	-815,773.49 (0.59)	0.57	127	4
Laplace approximation	-815,683.52			
Poisson regression				
$(n = 200,000, m = 500)$				
SMC	-260,888.69 (1.40)	0.94	80	4
Subsampling SMC	-260,887.87 (0.27)	0.14	80	5
Laplace approximation	-260,895.78			

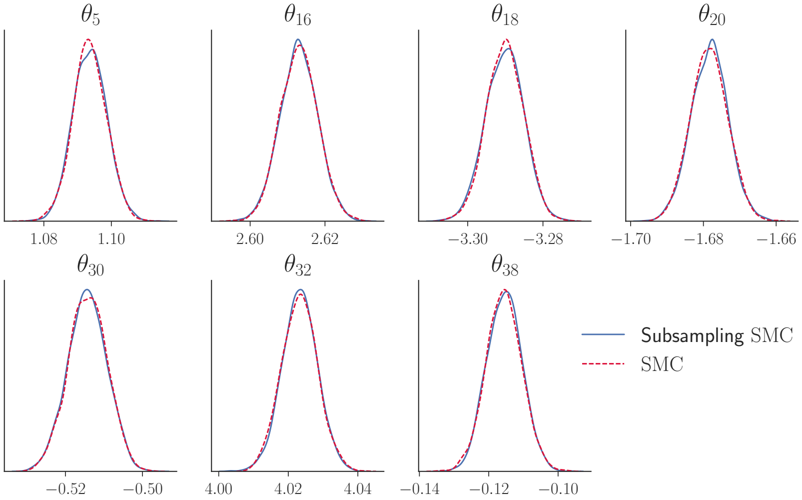


Figure 1 : Kernel density estimates of a subset of the marginal posterior densities of $\boldsymbol{\theta}$ for a Student-t regression model. The density estimates are obtained by SMC (full data) and Subsampling SMC.

SMC results: Posterior accuracy, part II

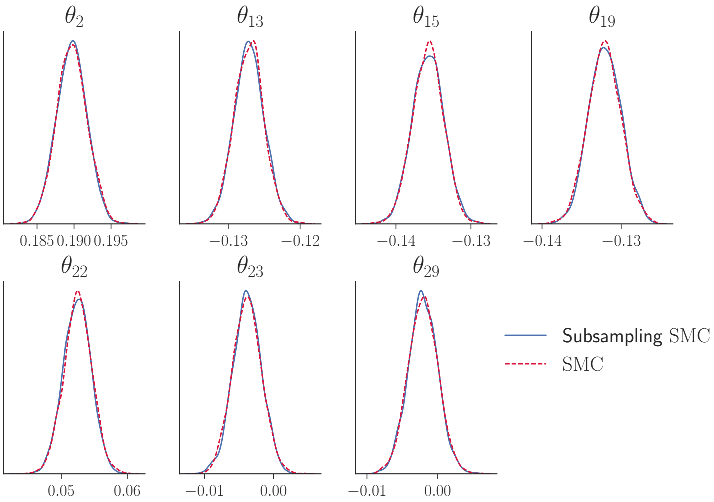


Figure 2 : Kernel density estimates of a subset of the marginal posterior densities of $\boldsymbol{\theta}$ for a Poisson regression model. The density estimates are obtained by SMC (full data) and Subsampling SMC.

Concluding the SMC part of the talk

- Presented the general idea for **sequential Monte Carlo** (SMC). Maybe useful for you?
- An approach to **speed up SMC** by data subsampling.
- SMC accurately estimates **marginal likelihoods**...
- ... which are very useful for **Bayesian model comparison**.
- SMC is, unlike MCMC, **trivial to parallelize** (in the particle dimension).

Thank you for listening!

Questions?

You can find our papers on
https://arxiv.org/abs/1708.00955
https://arxiv.org/abs/1805.03317

References I

Baker, J., Fearnhead, P., Fox, E. B., and Nemeth, C. (2017).
Control variates for stochastic gradient MCMC.
arXiv preprint arXiv:1706.05439.

Betancourt, M. (2015).
The fundamental incompatibility of scalable Hamiltonian Monte Carlo and naive data subsampling.
In *International Conference on Machine Learning*, pages 533–540.

Chen, T., Fox, E., and Guestrin, C. (2014).
Stochastic gradient Hamiltonian Monte Carlo.
In *International Conference on Machine Learning*, pages 1683–1691.

Dang, K.-D., Quiroz, M., Kohn, R., Tran, M.-N., and Villani, M. (2019).
Hamiltonian Monte Carlo with energy conserving subsampling.
arXiv preprint arXiv:1708.00955v3.

Gunawan, D., Dang, K.-D., Quiroz, M., Kohn, R., and Tran, M.-N. (2019).
Subsampling sequential Monte Carlo for static Bayesian models.
arXiv preprint arXiv:1805.03317v2.

References II

Quiroz, M., Kohn, R., Villani, M., and Tran, M.-N. (2018).
Speeding up MCMC by efficient data subsampling.
Journal of American Statistical Association, (To appear).

Welling, M. and Teh, Y. W. (2011).
Bayesian learning via stochastic gradient Langevin dynamics.
In *Proceedings of the 28th International Conference on Machine Learning*, pages 681–688.