Advances in sequential Monte Carlo methods

PhD Final Seminar

Joshua J Bon

Acknowledgement of Country

QUT acknowledges the Turrbal and Yugara as the First Nations owners of the lands where QUT now stands. We pay respect to their Elders, lores, customs and creation spirits. We recognise that these lands have always been places of teaching, research and learning. QUT acknowledges the important role Aboriginal and Torres Strait Islander people play within the QUT community.

Acknowledgements

Supervisors

- **Christopher Drovandi**
- **Anthony Lee** (Bristol University)
- **Tony Pettitt**

Institutions and Groups

- **School of Mathematical Sciences, QUT**
- **ARC Centre of Excellence for Mathematical and Statistical Frontiers**
- **QUT Centre for Data Science**
- **Bayesian Research and Applications Group**

Advances in sequential Monte Carlo methods

Adaptive methods for sequential Monte Carlo **tuned with particles**

Applied to

1. SMC for static models with **expensive likelihoods** (*Statistics and Computing*)
2. Particle filters for dynamic models with **intractable transition densities** ⭐
3. SMC for large sparse **binary** state-spaces ⭐

But why?

Advances in sequential Monte Carlo methods

Along the way

In addition to discussing my thesis:

1. Feynman-Kac models are elegant descriptions of SMC

2. Twisted Feynman-Kac models are exciting

3. Designing SMC to exploit problem structure

But first…

🌈 Everything is an integral

If μ is a probability distribution (or measure) on measurable space $(\mathsf{X}, \mathcal{X})$

$$\text{For } S \in \mathcal{X}, \quad \mu(S) = \int_S \mu(\mathrm{d}x)$$

$$\text{For } \varphi : \mathsf{X} \to \mathbb{R}, \quad \mu(\varphi) = \int_{\mathsf{X}} \varphi(x)\mu(\mathrm{d}x)$$

Or with density (mass) function $p_\mu(x)$

$$\mu(\mathrm{d}x) = p_\mu(x)\mathrm{d}x, \quad \text{for } x \in \mathsf{X}$$

- Implicit integrals are defined by $\varphi(x)\mu(\mathrm{d}x)$
- φ is assumed to be measurable

🌈 Everything is an integral

If K is a Markov kernel (or non-negative kernel) on measure space $(\mathsf{X}, \mathcal{X})$

$$\text{For } v \in \mathsf{X}, S \in \mathcal{X}, \quad K(v, S) = \int_S K(v, \mathrm{d}x)$$

$$\text{For } v \in \mathsf{X}, \varphi : \mathsf{X} \to \mathbb{R},$$

$$K(\varphi)(v) = \int_{\mathsf{X}} \varphi(x) K(v, \mathrm{d}x)$$

Or with density (mass) function $p_K(v, x)$

$$K(v, \mathrm{d}x) = p_K(v, x)\mathrm{d}x, \quad \text{for } v, x \in \mathsf{X}$$

- For a fixed $v \in \mathsf{X}$, $K(v, \cdot)$ is a measure
- For a fixed (measurable) function φ, $K(\varphi)(\cdot)$ is a measurable function
  - allows $\mu(K(\varphi))$ notation

Monte Carlo for Bayesian inference

Bayesian inference

- Prior ν
    - with density p_ν
- Likelihood $L(x)$ dependent on some data y
- Combined to give a posterior π
    - with density $p_\pi(x) \propto L(x)p_\nu(x)$

What's the normalising constant Z?

If $X \sim \pi$, how do I calculate $\mathbb{E}[\varphi(X)]$?

Monte Carlo

Vanilla Monte Carlo

$$\zeta^i \overset{\text{iid}}{\sim} \pi = \text{Law}(X) \qquad \mathbb{E}[\varphi(X)] \approx \frac{1}{N}\sum_{i=1}^{N}\varphi(\zeta^i) \qquad \pi(\varphi) \equiv \mathbb{E}[\varphi(X)]$$

Importance Sampling

$$\zeta^i \overset{\text{iid}}{\sim} \nu = \text{Law}(Y) \qquad \mathbb{E}[\varphi(X)] = \mathbb{E}[\varphi(Y)w(Y)] \approx \frac{1}{N}\sum_{i=1}^{N}\varphi(\zeta^i)w(\zeta^i)$$

Importance sampling

If $X \sim \mu$ and $Y \sim \nu$, importance sampling is based on the identity

$$\mathbb{E}[\varphi(X)] = \mathbb{E}\left[\varphi(Y)\frac{p_\mu(Y)}{p_\nu(Y)}\right]$$

If we only have access to unnormalised $\tilde{p}_\mu$,

$$\mathbb{E}[\varphi(X)] \approx \frac{\sum_{i=1}^N \varphi(\zeta^i) w(\zeta^i)}{\sum_{i=1}^N w(\zeta^i)}$$

where $\zeta^i \overset{\text{iid}}{\sim} \nu$ and $w(y) = \frac{\tilde{p}_\mu(y)}{p_\nu(y)}$

Sequential Monte Carlo: spicy importance sampling

🌶️ **Sequence of distributions**:

- $\gamma_0, \gamma_1, \ldots, \gamma_{n-1}, \gamma_n$

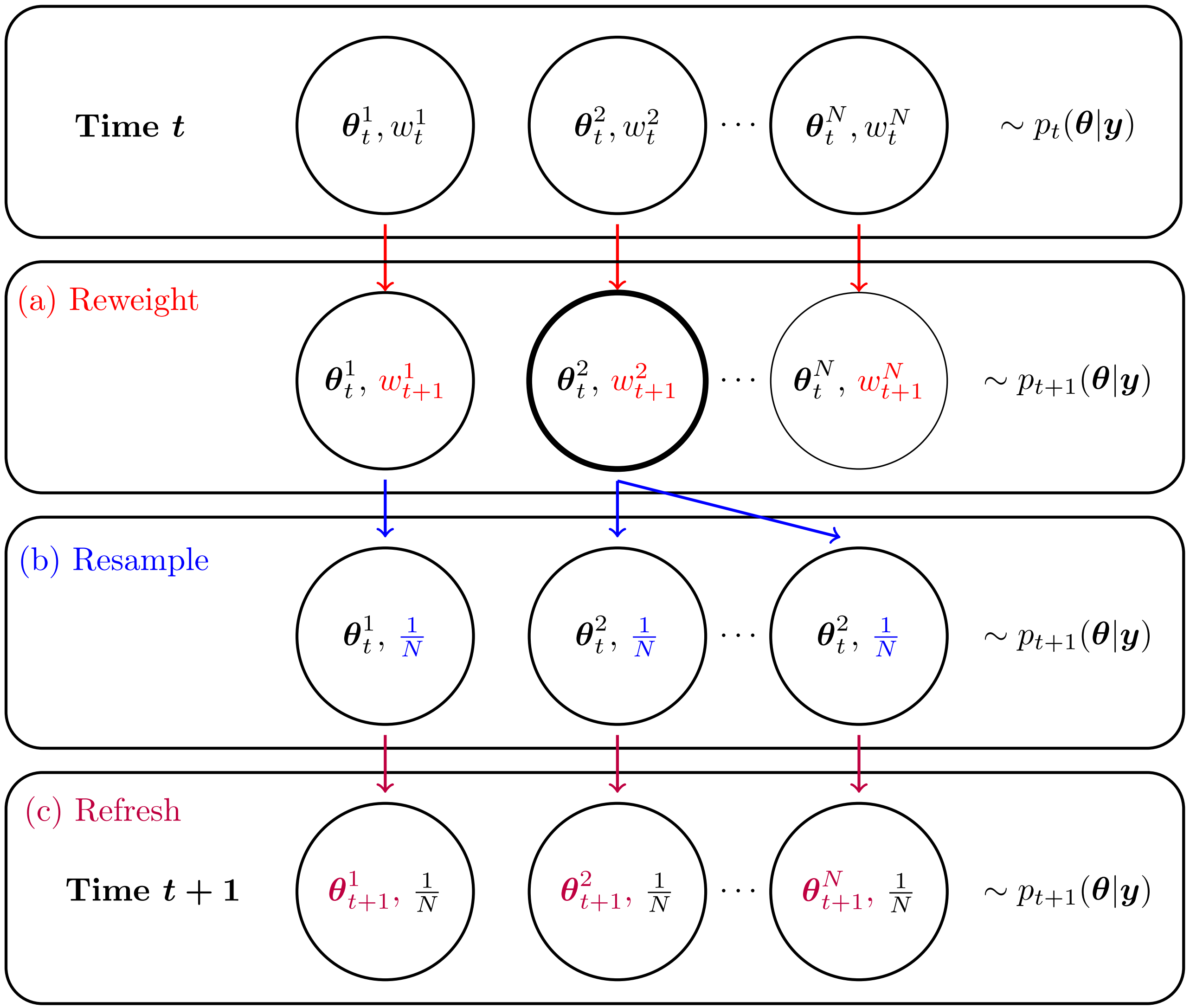
🌶️ **Weight & resample**: Randomly select particles according to weight.

- Survival of the fittest

🌶️ **Update**: Mutate samples

- Static models: Avoid degeneracy in particles (often MCMC)
- Dynamic models: Match next distribution

Example SMC algorithm: Resample-move for static models



Feynman-Kac models

Discrete-time Feynman-Kac models

A recipe for sequential Monte Carlo algorithms, with ingredients...

(In)homogeneous Markov chain

$$\boldsymbol{P}_n(\mathrm{d}x_{0:n}) = M_0(\mathrm{d}x_0)\prod_{t=1}^{n} M_t(x_{t-1}, \mathrm{d}x_t) \quad \text{on } (\mathsf{X}^{n+1}, \mathcal{X}^{\otimes n+1})$$

Potential functions

$$G_0(x_0), G_1(x_1), \ldots, G_n(x_n)$$

(But does not specify resampling mechanism)

Dynamic model example: Linear Gaussian Markov model

$$M_0(\mathrm{d}x_0) = \mathcal{N}(x_0; a_0, \Sigma_0)\mathrm{d}x_0$$

$$M_p(x_{p-1}, \mathrm{d}x_p) = \mathcal{N}(x_p; Ax_{p-1}, \Sigma_M)\mathrm{d}x_p$$

$$G_p(x_p) = \mathcal{N}(y_p; x_p, \Sigma_G)$$

- Inference over latent states
- Unbiased estimate of normalising constant

$$\begin{tikzcd}
\text{MP} & X_0 \arrow[r, "M_1"] \arrow[d, "\mathcal{N}"'] & X_1 \arrow[r, "M_2"] \arrow[d, "\mathcal{N}"'] & \dots \arrow[r, "M_n"] & X_n \arrow[d, "\mathcal{N}"'] \\
\text{Obs} & Y_0 & Y_1 & \dots & Y_n
\end{tikzcd}$$

❓ Static models have a fixed number of unknown quantities

On the path space

A sequence of iterated reweight and mutate steps…

- $\gamma_0(\mathrm{d}x_0) = M_0(\mathrm{d}x_0)$
- $\boldsymbol{\gamma}_1(\mathrm{d}x_{0:1}) = \gamma_0(\mathrm{d}x_0) G_0(x_0) M_1(x_0, \mathrm{d}x_1)$
- $\boldsymbol{\gamma}_2(\mathrm{d}x_{0:2}) = \boldsymbol{\gamma}_1(\mathrm{d}x_{0:1}) G_1(x_1) M_2(x_1, \mathrm{d}x_2)$

On the path space

A sequence of iterated reweight and mutate steps…

- $\gamma_p(\mathrm{d}x_{0:p}) = \gamma_{p-1}(\mathrm{d}x_{0:p-1}) G_{p-1}(x_{p-1}) M_p(x_{p-1}, \mathrm{d}x_p)$

$$\gamma_n(\mathrm{d}x_{0:n}) = M_0(\mathrm{d}x_0) \prod_{p=1}^{n} G_{p-1}(x_{p-1}) M_p(x_{p-1}, \mathrm{d}x_p)$$

Where is G_n?

$$\hat{\gamma}_n(\mathrm{d}x_{0:n}) = \gamma_n(\mathrm{d}x_{0:n}) G_n(x_n)$$

Predictive path measure γ_n, updated path measure $\hat{\gamma}_n$

Marginal measures

With $\gamma_0 = M_0$ we can define using a recursive formula

$$\gamma_p(S) = \int_{\mathsf{X}} \gamma_{p-1}(\mathrm{d}x_{p-1}) G_{p-1}(x_{p-1}) M_p(x_{p-1}, S)$$

$$\hat{\gamma}_p(\mathrm{d}x_p) = \gamma_p(\mathrm{d}x_p) G_p(x_p)$$

Normalised?

$$\eta_p(S) = \gamma_p(S)/\gamma_p(1)$$

$$\hat{\eta}_p(S) = \hat{\gamma}_p(S)/\hat{\gamma}_p(1)$$

Feynman-Kac summary

Marginal space

$x_p \in \mathsf{X}$

	Measure	**Distribution**
Predictive	γ_p	η_p
Updated	$\hat{\gamma}_p$	$\hat{\eta}_p$
Normalised	❌	✔️

for $p \in \{0, 1, \ldots, n\}$

Path space

$x_{0:p} \in \mathsf{X}^{p+1}$

	Measure	**Distribution**
Predictive	$\boldsymbol{\gamma}_p$	$\boldsymbol{\eta}_p$
Updated	$\hat{\boldsymbol{\gamma}}_p$	$\hat{\boldsymbol{\eta}}_p$
Normalised	❌	✔️

for $p \in \{0, 1, \ldots, n\}$

Feynman-Kac summary

- The Feynman-Kac model defines the sequence of distributions and how we traverse through them
  - γ_{p-1} reweight G_{p-1}, mutate M_p to γ_p
- Infinitely many models to perform the same statistical inference
- Many **bad** models for SMC possible (variance of estimates)

How do we improve an existing Feynman-Kac model?

Twisted Feynman-Kac models

Twist and ~~shout~~ tilt

Exponential tilting/twisting

For some distribution μ, define

$$\nu(\mathrm{d}x) = \frac{\mu(\mathrm{d}x)\exp(-\lambda x)}{Z}$$

with $Z = \int_{\mathsf{X}} \exp(-\lambda x)\mu(\mathrm{d}x)$.

Change of measure

$$\nu(\mathrm{d}x) = \frac{\mu(\mathrm{d}x)\psi(x)}{Z}$$

with $Z = \mu(\psi) = \int_{\mathsf{X}} \psi(x)\mu(\mathrm{d}x)$.

Twisting Feynman-Kac models

Twist the Markov kernels

$$M_0^{\psi}(\mathrm{d}x_0) = \frac{M_0(\mathrm{d}x_0)\psi_0(x_0)}{M_0(\psi_0)}$$

$$M_p^{\psi}(x_{p-1}, \mathrm{d}x_p) = \frac{M_p(x_{p-1}, \mathrm{d}x_p)\psi_p(x_p)}{M_p(\psi_p)(x_{p-1})}$$

Twisting Feynman-Kac models

Also "twist" the potential functions

$$G_0^{\psi}(x_0) = \frac{G_0(x_0)}{\psi_0(x_0)} M_1(\psi_1)(x_0) M_0(\psi_0)$$

$$G_p^{\psi}(x_p) = \frac{G_p(x_p)}{\psi_p(x_p)} M_{p+1}(\psi_{p+1})(x_p)$$

$$G_n^{\psi}(x_n) = \frac{G_n(x_n)}{\psi_n(x_n)}$$

Twisted mutation kernels and potential functions are then used by standard SMC

Twisting Feynman-Kac models

Why??

$$\hat{\gamma}_n^{\psi}(\mathrm{d}x_{0:n}) = M_0(\mathrm{d}x_0)G_0^{\psi}(x_0)\prod_{p=1}^{n} M_p(x_{p-1}, \mathrm{d}x_p)G_p^{\psi}(x_p)$$

$$\gamma_p^{\psi}(\mathrm{d}x_p) = \gamma_p(\mathrm{d}x_p)\psi_p(x_p)$$

$$\eta_p^{\psi}(\mathrm{d}x_p) = \frac{\eta_p(\mathrm{d}x_p)\psi_p(x_p)}{\eta_p(\psi_p)}$$

$$\hat{\eta}_n^{\psi} = \hat{\eta}_n$$

$$\hat{\gamma}_n^{\psi} = \hat{\gamma}_n$$

$$\hat{Z}_{\psi} = \hat{Z}$$

Optimal twisting

If we choose

$$\psi_p^\star(x_p) = \mathbb{E}\left(\prod_{t=p}^{n} G_t(X_t) \,\middle|\, X_p = x_p\right)$$

with $X_{0:n} \sim \boldsymbol{P}_n$, then

$$\hat{Z}_\psi^N \overset{a.s.}{=} \hat{Z}$$

and we have a **perfect** sampler for **finite** N (!) for the terminal distribution

Iterated APF and Controlled SMC

- IAPF (Guarniero et al, 2017) and CSMC (Heng et al, 2020) use twisted FK models
- Learn twisting functions using a recursion

The optimal twisting functions satisfy

$$\psi_n^\star(x_n) = G_n(x_n)$$

$$\psi_p^\star(x_p) = G_p(x_p) M_{p+1}(\psi_{p+1}^\star)(x_p)$$

Motivates recursive, iterative learning…

Algorithm: For class of approximate twisting function $\tilde{\psi}_p \in \mathsf{H}$

1. Run particle filter with current ψ, generate ζ_p^i
2. Find new approximation of $\tilde{\psi}_p(\zeta_p^i) \approx G_p(\zeta_p^i) M_{p+1}(\tilde{\psi}_{p+1})(\zeta_p^i)$ with backward recursion
3. Repeat

Existing applications of recursive learning

Twisted mutation M_p^ψ can be sampled from and constant $M_p(\psi_p)$ can be calculated **analytically**.

Normal models and exponential-quadratic ψ

$$M_p = \mathcal{N}(f(x_{p-1}), \Sigma)$$

$$\psi_p(x_p) = \exp\left(-x_p^\top A x_p + b x_p + c\right)$$

where A is PSD.

Then $M_p^\psi = \mathcal{N}(f_\psi(x_{p-1}), \Sigma_\psi)$ can be calculated **analytically**

❗ Almost all use-cases are normal

Beyond exact twisting ⭐

Extending beyond the analytical case

What if there is no class of appropriate ψ functions that are tractable?

- M_p intractable
- M_p^{ψ} unable to be sampled from

Idea: Rejection sampler + Unbiased estimate of twisted potentials

Extending beyond the analytical case

Rejection sampler

- Restrict $\psi : \mathsf{X} \to [0, 1]$
- Propose $x \sim M_p$ until $\psi_p(x) > U$, where $U \sim \text{Uniform}(0, 1)$

Accepted realisations have the correct twisted distribution: $x \sim M_p^{\psi}$

Extending beyond the analytical case

Unbiased estimate of twisted potentials

$$\tilde{G}_p^{\psi}(x_p) = \frac{G_p(x_p)}{\psi_p(x_p)} \tilde{M}_{p+1}(\psi_{p+1})(x_p)$$

$$\tilde{M}_{p+1}(\psi_{p+1})(x_p) = K^{-1} \sum_{i=1}^{K} \psi_{p+1}(u_{p+1}^i) \qquad u_{p+1}^i \sim M_{p+1}(x_p, \cdot)$$

Twisted models by rejection-sampling and unbiased potentials

Applicable to **any** Feynman-Kac model with bounded ψ and where mutations can be sampled from, using

- Rejection for twisted mutation
  - Exact twist
  - Potentially costly sampler
- Monte Carlo for twisted potential
  - Simple
  - Potentially noisy estimate

How to address concerns?

Controlling cost of rejection sampler

- Rejection sampler for M_p^ω has acceptance rate:

$$M_p(\omega_p)$$

- Conditional on x_{p-1} but embedded within SMC

- Average acceptance rate

$$\hat{\eta}_{p-1}^\omega(M_p(\omega_p)) \quad \text{or} \quad M_0(\omega_0)$$

Controlling cost of rejection sampler

Prop. If we have a ψ-twisted Feynman-Kac model, then the average acceptance rates of a ω-twisted model can be written as

$$\alpha_p^\omega \equiv \hat{\eta}_{p-1}^\omega(M_p(\omega_p)) = \frac{\hat{\eta}_{p-1}^\psi(M_p(\omega_p)^2 \cdot M_p(\psi_p)^{-1})}{\hat{\eta}_{p-1}^\psi(M_p(\omega_p) \cdot M_p(\psi_p)^{-1})} \text{ for } p \in [n],$$

$$\alpha_0^\omega \equiv M_0(\omega_0).$$

❗ A quantity for average RS acceptance rates within a particle filter

‼ A way to estimate without ever running the same particle filter

Target α: Use in the iterative learning algorithm and temper the new twisting functions

Sketch of proof

New twisting functions decomposed as $\omega_p = \psi_p \cdot \phi_p$

- Two elegant properties
  - $(\hat{\gamma}^{\psi}_{p-1})^{\phi} = \hat{\gamma}^{\psi \cdot \phi}_{p-1}$
  - $(\hat{\gamma}_{p-1})^{\phi}(\mathrm{d}x_{p-1}) = \hat{\gamma}_{p-1}(\mathrm{d}x_{p-1}) M_p(\phi_p)(x_{p-1})$
- Yields
  - $\hat{\gamma}^{\omega}_{p-1}(\mathrm{d}x_{p-1}) = \hat{\gamma}^{\psi}_{p-1}(\mathrm{d}x_{p-1}) M^{\psi}_p(\phi_p)(x_{p-1})$
  - Normalise then simplify

Can't evaluate $M^{\psi}_p(\phi_p)(x_{p-1})$ analytically, but substitute $M^{\psi}_p(\phi_p) = \frac{M_p(\omega_p)}{M_p(\psi_p)}$

Analysing PF variance with estimate of twisted potentials

Prop: If the relative variance of estimated potential is uniformly bounded, that is

$$\mathrm{Var}\left(\frac{\tilde{G}_p(\tilde{x}_p)}{G_p(x_p)} \,\Big|\, x_p\right) < C$$

then the asymptotic variance satisfies

$$\tilde{\sigma}_n^2(\varphi \otimes 1) < (C+1)\sigma_n^2(\varphi)$$

Flavour of proof

- Define a new equivalent Feynman-Kac model on augmented state-space
  - Extended state-space accounts for randomness of twisted potential estimate

- Asymptotic variance

$$\tilde{\sigma}_n^2(\varphi \otimes 1) = N^{-1} \sum_{p=0}^{n} \tilde{v}_{p,n}(\varphi \otimes 1)$$

Bound each $v_{p,n}(\varphi)$ using $\mathrm{Var}\left(\frac{\tilde{G}_p(\tilde{x}_p)}{G_p(x_p)} \;\middle|\; x_p \right) < C$

❗ Effect of using estimate is isolated, does not compound (asymptotically)

Example: Linear Gaussian Markov model

Model

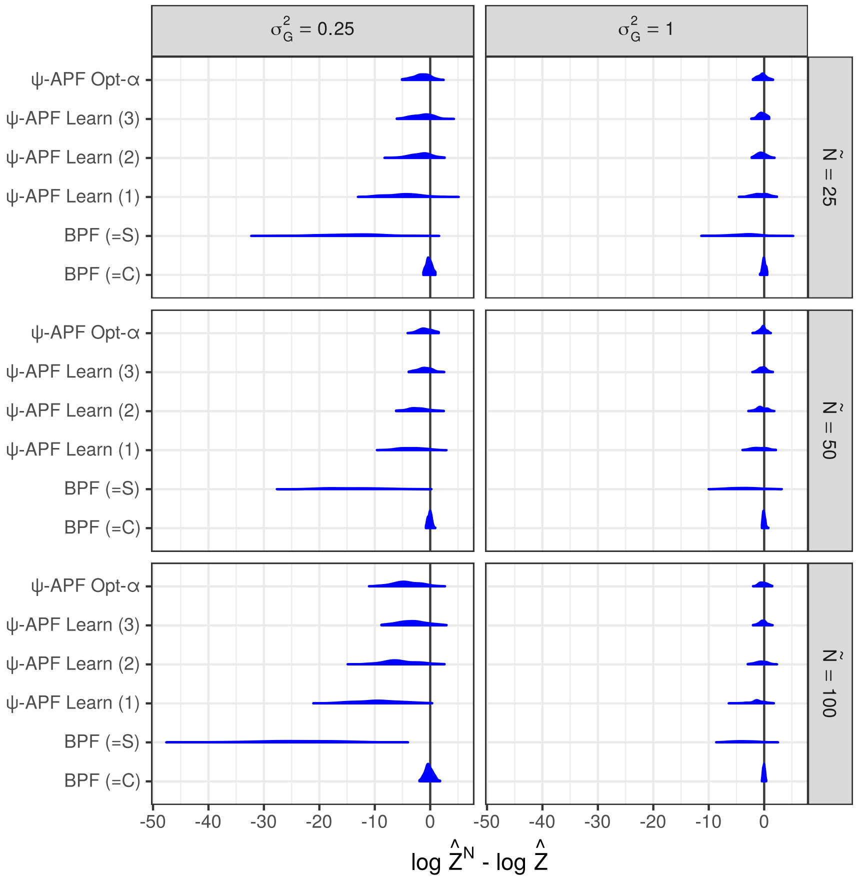
- $d = 3, n = 200$
- $a_0 = [1\ 1\ 1], \Sigma_0 = I_d$
- $\Sigma_M = I_d$
- $\Sigma_G = \sigma_G^2 I_d$ with $\sigma_G^2 \in \{0.25, 1.0\}$
- A are such that $A_{i,j} = a^{|i-j|+1}$ with $a = 0.42$

Algorithm

- 3 iterations of iterative learning
  - α target $= 0.04, 0.02, 0.01$
- $N = 200$ particles for twisted models
- Dynamic multinomial resampling with $N/2$ threshold
- $\tilde{N} \in \{25, 50, 100\}$

Tested over 100 repetitions

Results



σ^2_G = 0.25
σ^2_G = 1
ψ-APF Opt-α
ψ-APF Learn (3)
ψ-APF Learn (2)
ψ-APF Learn (1)
BPF (=S)
BPF (=C)
$\tilde{N}$ = 25
$\tilde{N}$ = 50
$\tilde{N}$ = 100
-50
-40
-30
-20
-10
0
$\log \hat{Z}^N - \log \hat{Z}$

Feynman-Kac models for large binary spaces ⭐

Motivation

- Particle filters for dynamic models operate over a $n \times d$ dimensional space
- Divide this problem in n, d-dimensional sub-problems
- Multivariate binary state-spaces have 2^d possible combinations (static models)

Can we decompose a large binary state-space into sub-problems amenable to SMC?

A new Feynman-Kac model

Mutation kernels

Let x_p be the active elements, e.g. if $w = [0\ 1\ 1\ 0]$, $x = \{2, 3\}$

$$M_0(\mathrm{d}x_0) = \delta_{\varnothing}(x_0)\mathrm{d}x_0$$

$$M_p(x_{p-1}, \mathrm{d}x_p) = f_p(x_{p-1}, x_p)\mathrm{d}x_p$$

$$f_p(x_{p-1}, x_p) = \begin{cases} \alpha & \text{if } x_p = x_{p-1} \\ \frac{1-\alpha}{|\rho(x_{p-1})|} & \text{if } x_p = x_{p-1} \cup z_p, \text{ for } z_p \in \rho(x_{p-1}) \\ 0 & \text{otherwise.} \end{cases}$$

A new Feynman-Kac model

Potential functions

$$G_0(x_0) = L(x_0)^{\theta_0}$$

$$G_p(x_{p-1}, x_p) = \frac{L(x_p)^{\theta_p}}{L(x_{p-1})^{\theta_{p-1}}} \text{ for } p \in [n]$$

Path space

$$\hat{\gamma}_n(\mathrm{d}x_{0:n}) = \delta_{\varnothing}(x_0)G_0(x_0)\prod_{p=1}^{n} f_p(x_{p-1}, x_p)G_p(x_{p-1}, x_p)\mathrm{d}x_{0:n}$$

$$= L(x_n)\delta_{\varnothing}(x_0)\prod_{p=1}^{n} f_p(x_{p-1}, x_p)\mathrm{d}x_{0:n}$$

Implied prior

- Uniform (marginal) probability one element is active
- Maximum n active elements
- α controls sparsity
- If $n = k$ and $\alpha = 0.5$ then uniform probability of entire binary space

Prior is flexible for a variety of sparsity cases

SMC algorithm uses less iterations for $n \ll k$

Testing against LI-MCMC

- Locally-informed MCMC is a state-of-the-art algorithm for sampling over discrete state-spaces
- Operates by weighting the proposal kernel in a Metropolis-Hastings algorithm by transformation of unnormalised posterior density

Model

- Variable selection with conjugate linear regression $\sigma^2 = 1$
- $m \in \{50, 100\}$ observations
- $k \in \{20, 30, 40\}$ covariates
- $\beta_0 = 1$, and $\beta = [10, 5, 0, \cdots, 0, 10, 5]$
- $n = 5$ maximum number of non-zero elements
- $\alpha = 0.4$ (no new active element)

Algorithm

- Particle filter with dynamic stratified resampling
- Resampling threshold $\kappa \in \{0.1, 0.5, 1.0\}$
- $\theta_p = 1$
- $N = 10^5$ particles
- Equivalent computation for LI-MCMC (posterior evaluations)

Testing against LI-MCMC

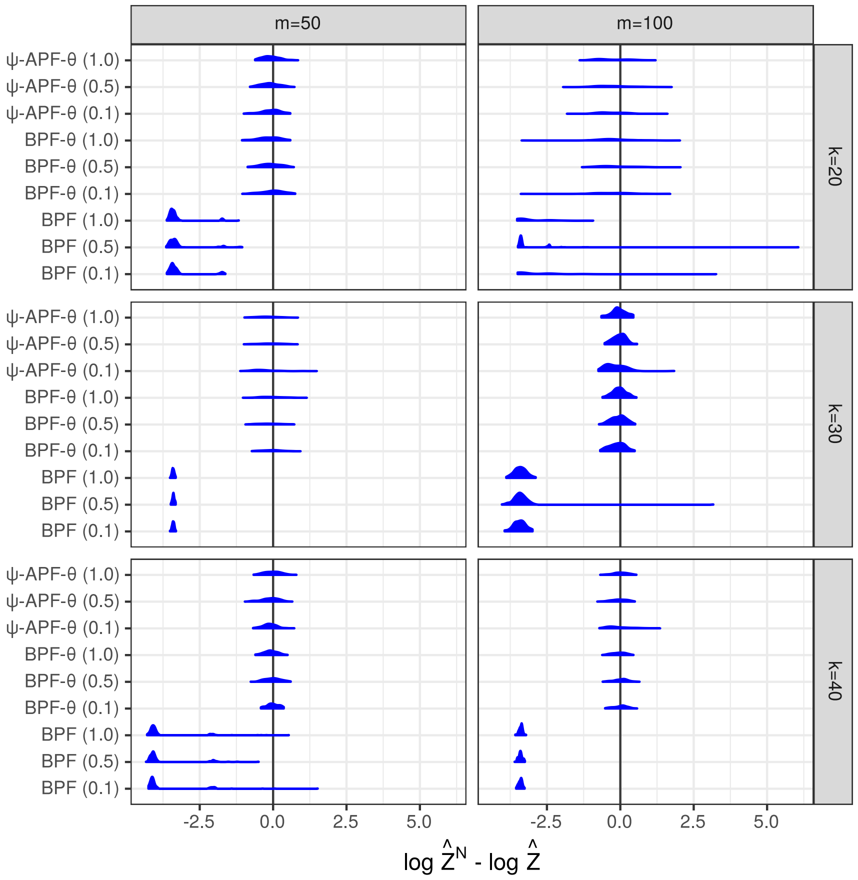
		LI-MCMC			BPF								
					$\kappa = 0.1$			$\kappa = 0.5$			$\kappa = 1.0$		
m	k	Mean	10%	90%	Mean	10%	90%	Mean	10%	90%	Mean	10%	90%
50	20	0.13	0.09	0.17	0.06	0.04	0.08	**0.05**	0.04	0.08	0.06	0.04	0.08
	30	20.0	0.25	100	**0.13**	0.08	0.17	0.13	0.09	0.18	0.13	0.09	0.17
	40	90.0	90.0	100	0.23	0.17	0.32	0.23	0.16	0.31	**0.22**	0.14	0.29
100	20	0.07	0.05	0.09	0.03	0.02	0.04	**0.03**	0.02	0.04	0.03	0.02	0.04
	30	0.16	0.12	0.20	0.07	0.05	0.10	**0.07**	0.05	0.09	0.07	0.05	0.10
	40	93.0	100	100	0.18	0.13	0.24	**0.17**	0.12	0.21	0.18	0.13	0.22

Using twisting ideas

Idea: Use definition of optimal twisting functions to choose temperatures θ_p.

- Novel tuning for Feynman-Kac with "free" parameters
- Recursion to select θ_p to make optimal twisting functions close to 1

Results



m=50
m=100
ψ-APF-θ (1.0)
ψ-APF-θ (0.5)
ψ-APF-θ (0.1)
BPF-θ (1.0)
BPF-θ (0.5)
BPF-θ (0.1)
BPF (1.0)
BPF (0.5)
BPF (0.1)
k=20
k=30
k=40
-2.5
0.0
2.5
5.0
log $\hat{Z}^N$ - log $\hat{Z}$

Future work and opportunities

Rejection-based twisting

- Other types of exact samplers
    - Use hybrid between rejection and analytical integration for better performance
- Apply rejection-based twisting ideas to new statistical models
- Relate average acceptance rate to asymptotic variance to choose target acceptance rate
    - Identify models method will be appropriate for

Future work and opportunities

Sparse binary state-spaces

- Extend to mixed discrete-continuous settings
- Select temperatures using twisting for other models

Thank you for listening!