Non-reversible samplers for mixture models

Filippo Ascolani

Duke University

January 14, 2026

Co-authors

Paolo Manildo
(University of Padova)

Giacomo Zanella
(Bocconi University)

Bayesian mixture models

$$\boxed{Y_i \mid \boldsymbol{\theta}, \boldsymbol{w} \overset{\text{i.i.d.}}{\sim} \sum_{k=1}^{K} w_k f_{\theta_k}(\cdot) \qquad i = 1, \ldots, n} \qquad\qquad \theta_k \overset{\text{i.i.d.}}{\sim} p_\theta, \quad \boldsymbol{w} = (w_1, \ldots, w_K) \sim p_{\boldsymbol{w}}.$$

$f_\theta =$ **parametric family**, $\quad K =$ **number of components** (finite or infinite)

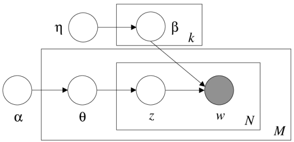
[1] Blei et al., 2004; Marin et al., 2005, McLachlan et al., 2019

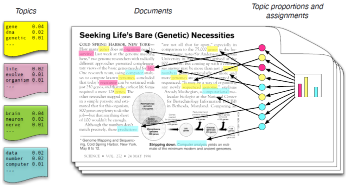
Bayesian mixture models

$$\boxed{Y_i \mid \theta, \boldsymbol{w} \overset{\text{i.i.d.}}{\sim} \sum_{k=1}^{K} w_k f_{\theta_k}(\cdot) \qquad i = 1, \dots, n} \qquad\qquad \theta_k \overset{\text{i.i.d.}}{\sim} p_\theta, \quad \boldsymbol{w} = (w_1, \dots, w_K) \sim p_{\boldsymbol{w}}.$$

f_θ = **parametric family**, K = **number of components** (finite or infinite)

- Very classical models. Applied in various fields in many variants[1]
- Building block of larger probabilistic models (e.g. hierarchical, temporal, . . .)
- Computationally challenging (i.e. algorithms slow for large n!)



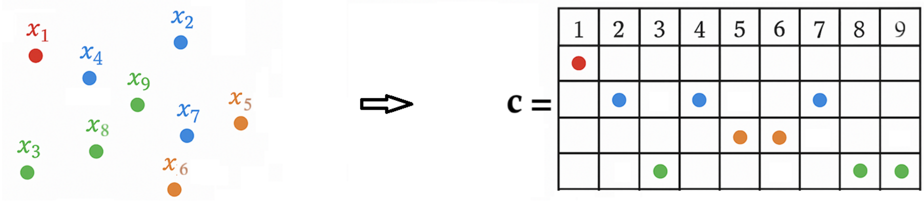


[1] Blei et al., 2004; Marin et al., 2005, McLachlan et al., 2019

Formulation with allocation variables

Introduce **allocation variables** $\boxed{c = (c_1, \ldots, c_n) \in [K]^n}$. Assume K **fixed** for now!

$$\boxed{Y_i \mid c, \boldsymbol{\theta}, \boldsymbol{w} \overset{\text{i.i.d.}}{\sim} f_{\theta_{c_i}}(y), \quad c_i \mid \boldsymbol{\theta}, \boldsymbol{w} \overset{\text{i.i.d.}}{\sim} \mathrm{Mult}(\boldsymbol{w}),} \qquad \theta_k \overset{\text{i.i.d.}}{\sim} p_0, \quad \boldsymbol{w} \sim p_{\boldsymbol{w}}$$



Classical MCMC for (finite) mixture models

If $\boxed{p_{\boldsymbol{w}} = \text{Dir}(\boldsymbol{\alpha})}$, with $\boldsymbol{\alpha} = (\alpha, \dots, \alpha)$:

- **Conditional sampler**: updates $\boxed{c \sim \pi(c|\boldsymbol{\theta}, \boldsymbol{w})}$ and $\boxed{(\boldsymbol{\theta}, \boldsymbol{w}) \sim \pi(\boldsymbol{\theta}, \boldsymbol{w}|c)}$ with target

$$\pi(c, \boldsymbol{\theta}, \boldsymbol{w}) \propto \prod_{k=1}^{K} w_k^{n_k(c)+\alpha-1} \prod_{i\,:\,c_i=k} f_{\theta_k}(Y_i) p_\theta(\theta_k) \qquad \boxed{n_k(c) = \sum_{i=1}^{n} \mathbb{1}(c_i = k)}$$

Classical MCMC for (finite) mixture models

If $\boxed{p_{\boldsymbol{w}} = \mathrm{Dir}(\boldsymbol{\alpha})}$, with $\boldsymbol{\alpha} = (\alpha, \dots, \alpha)$:

- **Conditional sampler**: updates $\boxed{c \sim \pi(c|\boldsymbol{\theta}, \boldsymbol{w})}$ and $\boxed{(\boldsymbol{\theta}, \boldsymbol{w}) \sim \pi(\boldsymbol{\theta}, \boldsymbol{w}|c)}$ with target

$$\pi(c, \boldsymbol{\theta}, \boldsymbol{w}) \propto \prod_{k=1}^{K} w_k^{n_k(c)+\alpha-1} \prod_{i\,:\,c_i=k} f_{\theta_k}(Y_i) p_\theta(\theta_k) \qquad \boxed{n_k(c) = \sum_{i=1}^{n} \mathbb{1}(c_i = k)}$$

- **Marginal sampler**: updates $\boxed{c_i \sim \pi(c_i \mid c_{-i}) \text{ for } i \in [n]}$ with target

$$\pi(c) \propto \prod_{k=1}^{K} \Gamma\left(\alpha + n_k(c)\right) \int_\Theta \prod_{i\,:\,c_i=k} f_{\theta_k}(Y_i) p_\theta(\theta_k)\, \mathrm{d}\theta_k$$

Classical MCMC for (finite) mixture models

If $\boxed{p_{\boldsymbol{w}} = \mathrm{Dir}(\boldsymbol{\alpha})}$, with $\boldsymbol{\alpha} = (\alpha, \dots, \alpha)$:

- **Conditional sampler**: updates $\boxed{c \sim \pi(c|\boldsymbol{\theta}, \boldsymbol{w})}$ and $\boxed{(\boldsymbol{\theta}, \boldsymbol{w}) \sim \pi(\boldsymbol{\theta}, \boldsymbol{w}|c)}$ with target

$$\pi(c, \boldsymbol{\theta}, \boldsymbol{w}) \propto \prod_{k=1}^{K} w_k^{n_k(c)+\alpha-1} \prod_{i\,:\,c_i=k} f_{\theta_k}(Y_i) p_\theta(\theta_k) \qquad \boxed{n_k(c) = \sum_{i=1}^{n} \mathbb{1}(c_i = k)}$$

- **Marginal sampler**: updates $\boxed{c_i \sim \pi(c_i \mid c_{-i}) \text{ for } i \in [n]}$ with target

$$\pi(c) \propto \prod_{k=1}^{K} \Gamma\left(\alpha + n_k(c)\right) \int_\Theta \prod_{i\,:\,c_i=k} f_{\theta_k}(Y_i) p_\theta(\theta_k)\, \mathrm{d}\theta_k$$

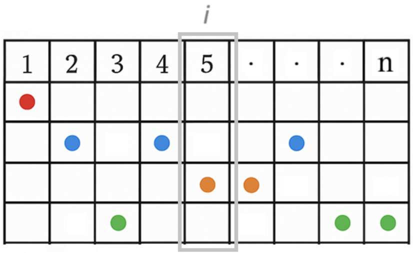
- Similarly happens with $\boxed{K = \infty}$ and $p_{\boldsymbol{w}}$ the GEM distribution $\Rightarrow$ **Dirichlet process**!

Marginal (Gibbs) sampler

$\pi(c)$-reversible Markov kernel P_{MG} on $[K]^n$

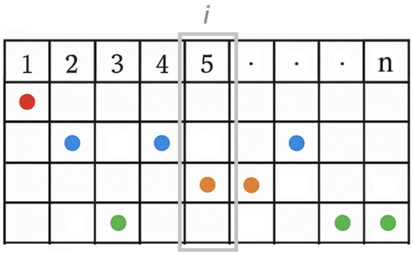
At each iteration:

1. Sample $i \sim \mathrm{Unif}([n])$
2. Update $c_i \sim \pi(c_i \mid c_{-i})$



Marginal (Gibbs) sampler

$\pi(c)$-reversible Markov kernel P_{MG} on $[K]^n$



At each iteration:

1. Sample $i \sim \mathrm{Unif}([n])$
2. Update $c_i \sim \pi(c_i \mid c_{-i})$

- Arguably among the most popular MCMC schemes for mixture models.
- Simple to implement.
- $\mathcal{O}(K)$ cost per iteration.

How does it **scale** with n?

Prior case: slow convergence

Consider the prior case: $\boxed{f_\theta(y) = f(y)}$ $\Leftarrow$ limiting case of **weakly informative** data.

Prior case: slow convergence

Consider the prior case: $\boxed{f_\theta(y) = f(y)}$ $\Leftarrow$ limiting case of **weakly informative** data.

Theorem (variation of Khare and Zou, 2009)

The L^2-relaxation time of P_{MG} is

$$t_{rel} = \frac{n(n + K\alpha - 1)}{K\alpha} \approx n^2$$

- Related to Pólya urns and models in population genetics
- Implication: $\boxed{\mathcal{O}(n^2)}$ required for convergence
- Intuition: **random-walk behaviour** $\Rightarrow$ see later!

Posterior case: slow convergence

Consider data generated as

$$\boxed{Y_i \overset{\text{i.i.d.}}{\sim} 0.9N(y \mid 0.9, 1) + 0.1N(y \mid -0.9, 1)}, \quad i = 1, \dots, n = 2000$$

and consider $K = 2$ and $f_\theta(y) = N(y \mid \theta, 1) \quad \Leftarrow \quad$ **"easy" problem.**

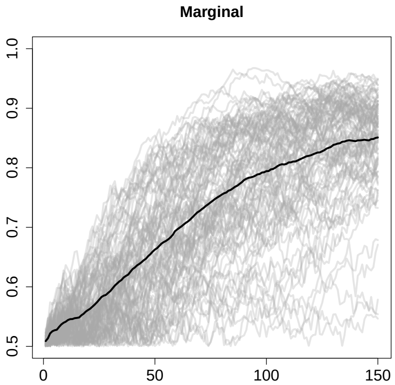
Posterior case: slow convergence

Consider data generated as

$$\boxed{Y_i \overset{\text{i.i.d.}}{\sim} 0.9N(y \mid 0.9, 1) + 0.1N(y \mid -0.9, 1)}, \quad i = 1, \ldots, n = 2000$$

and consider $K = 2$ and $f_\theta(y) = N(y \mid \theta, 1) \quad \Leftarrow \quad$ **“easy” problem.**

- Traceplot of the **size of the largest cluster**.
- Initialized **uniformly at random**.
- Thinning of size $n = 2000$.
- We expect to be close to 0.9 **in stationarity**.



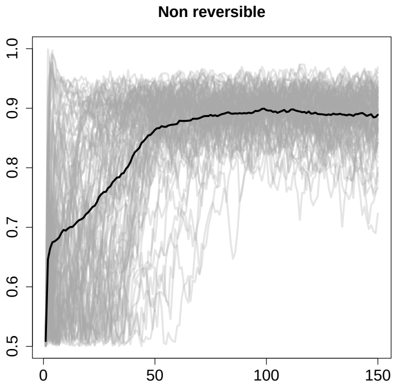
Posterior case: our proposal

Consider data generated as

$$\boxed{Y_i \overset{\text{i.i.d.}}{\sim} 0.9N(y \mid 0.9, 1) + 0.1N(y \mid -0.9, 1)}, \quad i = 1, \dots, n = 2000$$

and consider $K = 2$ and $f_\theta(y) = N(y \mid \theta, 1) \quad \Leftarrow \quad$ **“easy” problem.**

- Traceplot of the **size of the largest cluster**.
- Initialized **uniformly at random**.
- Thinning of size $n = 2000$.
- We expect to be close to 0.9 **in stationarity**.



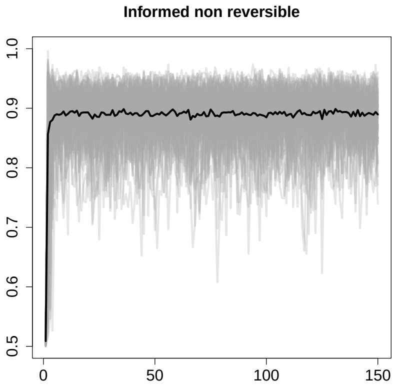
Posterior case: our proposal (advanced)

Consider data generated as

$$\boxed{Y_i \stackrel{\text{i.i.d.}}{\sim} 0.9N(y \mid 0.9, 1) + 0.1N(y \mid -0.9, 1)}, \quad i = 1, \dots, n = 2000$$

and consider $K = 2$ and $f_\theta(y) = N(y \mid \theta, 1) \quad \Leftarrow \quad$ **"easy" problem.**

- Traceplot of the **size of the largest cluster**.
- Initialized **uniformly at random**.
- Thinning of size $n = 2000$.
- We expect to be close to 0.9 **in stationarity**.



Goal of this talk

- The marginal sampler is **provably and empirically** slow under many scenarios of interest.
- Its issues **when n is large** have been known for a long time [2].
- P_{MG} is arguably one of the most popular MCMC schemes for mixture models.

[2] Celeux et al. (2000)

Goal of this talk

- The marginal sampler is **provably and empirically** slow under many scenarios of interest.
- Its issues **when n is large** have been known for a long time [2].
- P_{MG} is arguably one of the most popular MCMC schemes for mixture models.

What's next?

- Understanding why P_{MG} is slow.
- Showing a **random-walk** behaviour of P_{MG} when n is large.
- Exploit recent literature on **non-reversible** sampler to devise a **simple** and **more efficient** MCMC scheme (scaling **linearly** with n in the prior case).

[2] Celeux et al. (2000)

Goal of this talk

- The marginal sampler is **provably and empirically** slow under many scenarios of interest.
- Its issues **when *n* is large** have been known for a long time [2].
- P_{MG} is arguably one of the most popular MCMC schemes for mixture models.

What's next?

- Understanding why P_{MG} is slow.
- Showing a **random-walk** behaviour of P_{MG} when n is large.
- Exploit recent literature on **non-reversible** sampler to devise a **simple** and **more efficient** MCMC scheme (scaling **linearly** with n in the prior case).

A. F. and *Zanella, G.* (2026+) A fast non-reversible sampler for Bayesian finite mixture models. *Under review*.

[2] Celeux et al. (2000)

Insight: scaling limit

$\{c^{(t)}\}_t$ Markov chain on $[K]^n$ with kernel P_{MG}

[3] Deinitializing Markov chain using the terminology of Roberts and Rosenthal (2001)

Insight: scaling limit

$\{c^{(t)}\}_t$ Markov chain on $[K]^n$ with kernel P_{MG}

Consider prior case: $\boxed{f_\theta(y) = f(y)}$. Define

$$\boxed{X_{t,k}(c) = \frac{n_k\left(c^{(t)}\right)}{n}} \quad = \frac{\text{multiplicity of component } k \text{ at iteration } t}{n}$$

By symmetry of $\pi(c)$ across $i \in [n]$, **convergence of** $c^{(t)}$ fully determined[3] by the Markov chain

$$\boldsymbol{X}_t = (X_{t,1}, \ldots, X_{t,K}) \qquad t = 0, 1, 2, \ldots$$

[3] Deinitializing Markov chain using the terminology of Roberts and Rosenthal (2001)

Insight: scaling limit

Expected change **after one iteration**:

$$\mathbb{E}\left[X_{t+1,k} - x_k \mid \boldsymbol{X}_t = \boldsymbol{x}\right] = \frac{1}{n}\left[(1 - x_k)\frac{\alpha + n x_k}{K\alpha + n - 1} - x_k \frac{K\alpha - \alpha + n(1 - x_k)}{K\alpha + n - 1}\right]$$

$$= \frac{2}{n^2}\left[\frac{\alpha}{2} - K\alpha\frac{x_k}{2} + o(1)\right]$$

Insight: scaling limit

Expected change **after one iteration**:

$$\mathbb{E}\left[X_{t+1,k} - x_k \mid \boldsymbol{X}_t = \boldsymbol{x}\right] = \frac{1}{n}\left[(1-x_k)\frac{\alpha + n x_k}{K\alpha + n - 1} - x_k \frac{K\alpha - \alpha + n(1-x_k)}{K\alpha + n - 1}\right]$$

$$= \frac{2}{n^2}\left[\frac{\alpha}{2} - K\alpha\frac{x_k}{2} + o(1)\right]$$

- We expect $\mathcal{O}(n^2)$ iterations are needed for $\mathcal{O}(1)$ distance.
- Intuition: **the two probabilities cancel**!
- This can be made more rigorous.

Insight: scaling limit

Let $\boxed{\boldsymbol{Z}_t^{(n)} = \boldsymbol{X}_{\lceil n^2 t \rceil}}$ $\quad\Leftarrow\quad$ time **acceleration** by $\mathcal{O}(n^2)$.

Insight: scaling limit

Let $\boxed{\boldsymbol{Z}_t^{(n)} = \boldsymbol{X}_{\lceil n^2 t \rceil}}$ $\quad \Leftarrow \quad$ time **acceleration** by $\mathcal{O}(n^2)$.

Theorem

$\{\boldsymbol{Z}_t^{(n)}\}_{t\in\mathbb{R}_+} \to \{\boldsymbol{Z}_t\}_{t\in\mathbb{R}_+}$ weakly as $n \to \infty$, where $\{\boldsymbol{Z}_t\}_{t\in\mathbb{R}_+}$ is a diffusion process with generator

$$Lg(\boldsymbol{x}) = \sum_{k=1}^{K} \alpha(1 - Kx_k)\frac{\partial}{\partial x_k} g(\boldsymbol{x}) + \sum_{k,k'=1}^{K} x_k(\delta_{kk'} - x_{k'})\frac{\partial^2}{\partial x_k \partial x_{k'}} g(\boldsymbol{x}),$$

- Wright-Fisher process = **diffusion** on the unit simplex
- **Diffusive** behaviour **at the level of cluster sizes**.

Insight: scaling limit

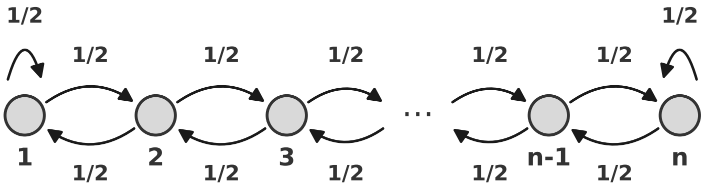
Main reason underlying **diffusive behaviour**:

$$\mathbb{P}\left(X_{t+1,k} - x_k = +\frac{1}{n} \mid \boldsymbol{X}_t = \boldsymbol{x}\right) \approx x_k(1 - x_k) \approx \mathbb{P}\left(X_{t+1,k} - x_k = -\frac{1}{n} \mid \boldsymbol{X}_t = \boldsymbol{x}\right).$$

- Almost **equally likely** to move along the two directions.
- The chain moves back and forth a lot!
- Reasonable that this happens also **a posteriori** (in weakly informative cases).
- How to solve this?

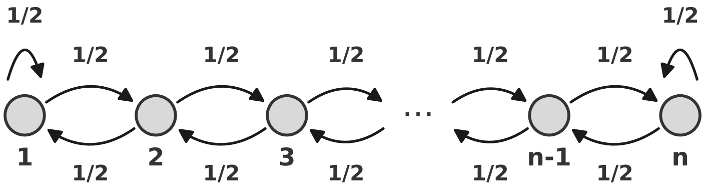
A simple example: problem

Chain in Diaconis et al. (2000).

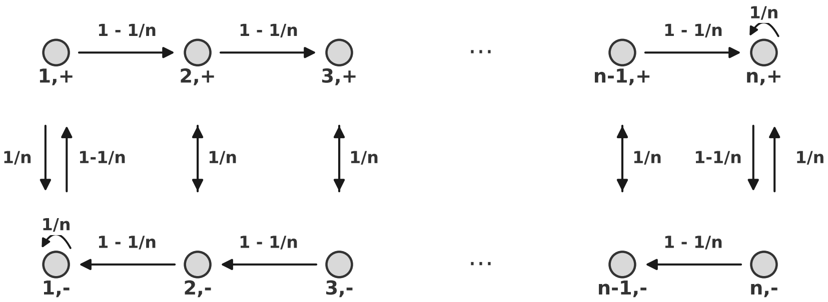


A simple example: problem

Chain in Diaconis et al. (2000).

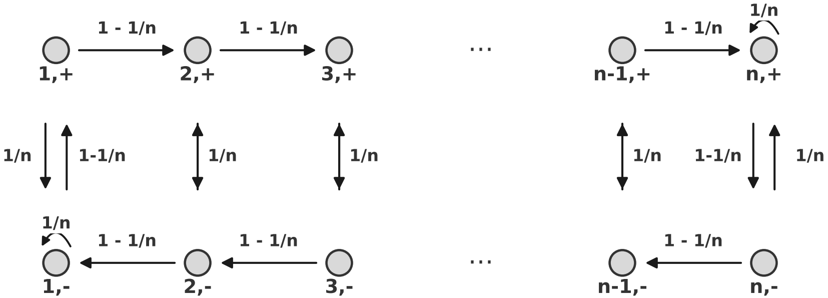


- Reversible chain with n **states**.
- $\mathcal{O}(n^2)$ iterations are needed to converge $\Rightarrow$ similar to our case!



1 - 1/n
1 - 1/n
1 - 1/n
1/n
1,+
2,+
3,+
...
n-1,+
n,+
1/n
1-1/n
1/n
1/n
1/n
1-1/n
1/n
1/n
1 - 1/n
1 - 1/n
1 - 1/n
1,-
2,-
3,-
...
n-1,-
n,-

A simple example: solution



- Non-reversible (**lifted**) chain $\Rightarrow$ we add a **direction**!
- $\mathcal{O}(n)$ iterations are needed to converge $\Rightarrow$ fast!

Non-reversible sampler (informal)

Extended target: $\boxed{\tilde{\pi}(c, v) = \pi(c)\left(\frac{1}{2}\right)^{K(K-1)/2}}$ $\quad c \in [K]^n,\ v = (v_{k,k'})_{k<k'} \in \{-1, +1\}^{K(K-1)/2}$

$\boxed{v_{k,k'} = \textbf{ direction across clusters } k \textbf{ and } k'}$

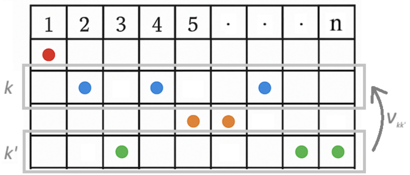
Non-reversible sampler (informal)

Extended target: $\boxed{\tilde{\pi}(c, v) = \pi(c)\left(\frac{1}{2}\right)^{K(K-1)/2}}$ $\quad c \in [K]^n,\ v = (v_{k,k'})_{k<k'} \in \{-1, +1\}^{K(K-1)/2}$

$\boxed{v_{k,k'} = \textbf{ direction across clusters } k \textbf{ and } k'}$

$\tilde{\pi}(c, v)$-invariant **Markov kernel** P_{NR}:

1. Sample a pair of clusters $(k, k') \in [K]^2$.
2. Propose to **move a single observation according to** $v_{kk'}$.
3. Accept with usual Metropolis-Hastings ratio.
4. If rejected, **flip** $v_{kk'}$.



Non-reversible sampler

Extended target: $\boxed{\tilde{\pi}(c, v) = \pi(c)\left(\frac{1}{2}\right)^{K(K-1)/2}}$ $\quad c \in [K]^n,\ v = (v_{k,k'})_{k<k'} \in \{-1, +1\}^{K(K-1)/2}$

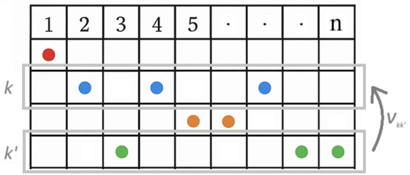
$\boxed{v_{k,k'} = \textbf{ direction across clusters } k \textbf{ and } k'}$

$\tilde{\pi}(c, v)$-invariant **Markov kernel** P_{NR}:

1. Sample $(k, k') \in [K]^2$ with probability $\frac{n_k(c)+n_{k'}(c)}{2(K-1)n}\mathbb{1}(k < k')$
2. Set $(k_-, k_+) = (k, k')\mathbb{1}(v_{k,k'} = +1) + (k', k)\mathbb{1}(v_{k,k'} = -1)$
3. Sample $i \sim \mathrm{Unif}(\{i' : c_{i'} = k_-\})$ and set $c_i = k_+$ with prob.

$$\min\left\{1, \left(\frac{n_{k_-}(c)}{n_{k_+}(c)+1}\right)\frac{\pi(c_i = k_+ \mid c_{-i})}{\pi(c_i = k_- \mid c_{-i})}\right\}.$$

If reject, flip $v_{kk'}$



Remarks

- P_{NR} is a **mixture of lifted kernels** with selection[4] probabilities p_c:

$$P_{\mathrm{NR}}(c, c') = \sum_{k<k'} p_c(k, k') P^{(lift)}_{kk'}(c, c')$$

where $P^{(lift)}_{kk'}$ is the MH-lift with **direction** $v_{kk'}$.

$\rightsquigarrow$ **multiple velocity** components $(v_{kk'})_{k<k'}$. At each rejection, flip only one of them.

[4] Here $p_c(k, k') = \frac{n_k(c)+n_{k'}(c)}{2(K-1)n}$

Remarks

- P_{NR} is a **mixture of lifted kernels** with selection[4] probabilities p_c:

$$P_{\mathrm{NR}}(c, c') = \sum_{k<k'} p_c(k, k') P^{(lift)}_{kk'}(c, c')$$

where $P^{(lift)}_{kk'}$ is the MH-lift with **direction** $v_{kk'}$.

$\rightsquigarrow$ **multiple velocity** components $(v_{kk'})_{k<k'}$. At each rejection, flip only one of them.

- For mixture models, **acceptance probability** becomes

$$\frac{n_{k_-}(c)}{n_{k_+}(c)+1} \frac{\pi(c_i = k_+ \mid c_{-i})}{\pi(c_i = k_- \mid c_{-i})} = \underbrace{\frac{p(Y_i \mid Y_{-i}, c_{-i}, c_i = k_+)}{p(Y_i \mid Y_{-i}, c_{-i}, c_i = k_-)}}_{\text{likelihood ratio}} (1 + O(n^{-1}))$$

$\rightsquigarrow$ proposal **matches the prior** to favour **long excursions**!

[4]Here $p_c(k, k') = \frac{n_k(c)+n_{k'}(c)}{2(K-1)n}$

Scaling limit

Again prior case $\boxed{f_\theta(y) = f(y)}$

$\boldsymbol{Z}_t^{(n)} = (\boldsymbol{X}_{\lceil nt \rceil}, \boldsymbol{V}_{\lceil nt \rceil}) \quad \Leftarrow \quad$ time **acceleration** by $\mathcal{O}(n)$

Theorem

$\{\boldsymbol{Z}_t^{(n)}\}_{t\in\mathbb{R}_+} \to \{\boldsymbol{Z}_t\}_{t\in\mathbb{R}_+}$ *weakly as* $n \to \infty$, *where* $\{\boldsymbol{Z}_t\}_{t\in\mathbb{R}_+}$ *is an (ergodic) piecewise deterministic Markov process*

- **No diffusive** behaviour
- P_{NR} gives $\mathcal{O}(n)$ **speedup** relative to P_{MG} in prior case!

Scaling limit

Again prior case $\boxed{f_\theta(y) = f(y)}$

$\boldsymbol{Z}_t^{(n)} = \left(\boldsymbol{X}_{\lceil nt \rceil}, \boldsymbol{V}_{\lceil nt \rceil}\right) \quad \Leftarrow \quad$ time **acceleration** by $\mathcal{O}(n)$

Theorem

$\{\boldsymbol{Z}_t^{(n)}\}_{t\in\mathbb{R}_+} \to \{\boldsymbol{Z}_t\}_{t\in\mathbb{R}_+}$ *weakly as* $n \to \infty$, *where* $\{\boldsymbol{Z}_t\}_{t\in\mathbb{R}_+}$ *is an (ergodic) piecewise deterministic Markov process*

- **No diffusive** behaviour
- P_{NR} gives $\mathcal{O}(n)$ **speedup** relative to P_{MG} in prior case!

What about more general π?

Asymptotic variance comparisons

$$\mathrm{Var}(g, P) := \lim_{T \to \infty} \mathrm{Var}\left(\frac{1}{\sqrt{T}} \sum_{t=1}^{T} g(X_t) \right) \qquad \text{for } X_0 \sim \pi\,,\ X_{t+1} | X_t \sim P(X_t, \cdot)$$

Asymptotic variance comparisons

$$\mathrm{Var}(g,P) := \lim_{T\to\infty} \mathrm{Var}\left(\frac{1}{\sqrt{T}}\sum_{t=1}^{T} g(X_t)\right) \qquad \text{for } X_0 \sim \pi\,,\ X_{t+1}|X_t \sim P(X_t,\cdot)$$

Theorem

For every π on $[K]^n$ and $g : [K]^n \to \mathbb{R}$ we have

$$\boxed{Var(g, P_{\mathrm{NR}}) \leq 2(K-1)\, Var(g, P_{\mathrm{MG}}) + (2K-3)\, Var_{\pi}(g)}$$

and $\boxed{Var(g, P_{\mathrm{MG}}) \leq Var(g, P_{\mathrm{CD}})}$

P_{MG} = marginal sampler; P_{NR} = non-reversible sampler; P_{CD} = conditional sampler

Asymptotic variance comparisons

Theorem (Approximately)

For every π on $[K]^n$ and $g : [K]^n \to \mathbb{R}$ we have

$$\boxed{Var(g, P_{\mathrm{NR}}) \leq 2(K-1)\, Var(g, P_{\mathrm{MG}})}$$

- $\mathrm{Cost}(P_{\mathrm{MG}}) = \mathcal{O}(K)$ and $\mathrm{Cost}(P_{\mathrm{NR}}) = \mathcal{O}(1)$ $\rightsquigarrow$ **little to loose** by using P_{NR} instead of P_{MG} (typical feature of lifted schemes)

Asymptotic variance comparisons

Theorem (Approximately)

For every π on $[K]^n$ and $g : [K]^n \to \mathbb{R}$ we have

$$\boxed{Var(g, P_{\mathrm{NR}}) \leq 2(K-1)\, Var(g, P_{\mathrm{MG}})}, \qquad \boxed{Var(g, P_{\mathrm{MG}}) \leq Var(g, P_{\mathrm{CD}})}$$

- $\mathrm{Cost}(P_{\mathrm{MG}}) = \mathcal{O}(K)$ and $\mathrm{Cost}(P_{\mathrm{NR}}) = \mathcal{O}(1)$ $\rightsquigarrow$ **little to loose** by using P_{NR} instead of P_{MG} (typical feature of lifted schemes)
- The conditional sampler is always **less efficient** than the marginal one.

Asymptotic variance comparisons

Theorem (Approximately)

For every π on $[K]^n$ and $g : [K]^n \to \mathbb{R}$ we have

$$\boxed{Var(g, P_{\mathrm{NR}}) \leq 2(K-1)\, Var(g, P_{\mathrm{MG}})}, \qquad \boxed{Var(g, P_{\mathrm{MG}}) \leq Var(g, P_{\mathrm{CD}})}$$

- $\mathrm{Cost}(P_{\mathrm{MG}}) = \mathcal{O}(K)$ and $\mathrm{Cost}(P_{\mathrm{NR}}) = \mathcal{O}(1)$ $\rightsquigarrow$ **little to loose** by using P_{NR} instead of P_{MG} (typical feature of lifted schemes)
- The conditional sampler is always **less efficient** than the marginal one.
- Do we gain much when targeting mixture model posteriors?

Bayesian discrete posteriors: does lifting help?

- Data often makes Bayesian posteriors with discrete parameters[5] **sharply concentrated** and non-smooth
$\rightsquigarrow$ large 'discrete gradients' speed-up reversible samplers[6] while reducing excursion lengths for lifted MCMC[7]

[5] e.g. variable selection, stochastic block model, graphical models
[6] Yang et al., 2016; Zhou et al., 2022; Zhou and Chang, 2023, ...
[7] different from, e.g., successful applications of lifting to Statistical Physics models

Bayesian discrete posteriors: does lifting help?

- Data often makes Bayesian posteriors with discrete parameters[5] **sharply concentrated** and non-smooth
  $\rightsquigarrow$ large 'discrete gradients' speed-up reversible samplers[6] while reducing excursion lengths for lifted MCMC[7]
- By contrast, mixture models have statistical features that are well-suited to lifted samplers, e.g.:
  1. **Lack of posterior concentration**
  2. **Flatness in the tails**
  3. **Overfitted regimes**

[5] e.g. variable selection, stochastic block model, graphical models
[6] Yang et al., 2016; Zhou et al., 2022; Zhou and Chang, 2023, ...
[7] different from, e.g., successful applications of lifting to Statistical Physics models

Posterior case: statistical features of mixture model

1. **Lack of posterior concentration for** c: for data $(Y_1, \dots, Y_n)$ generated from mixture with true parameters $(\boldsymbol{\theta}^*, \boldsymbol{w}^*)$:[8]

$$\pi(\boldsymbol{\theta}, \boldsymbol{w}) \to \delta_{(\boldsymbol{\theta}^*, \boldsymbol{w}^*)} \quad \text{as } n \to \infty$$

$$\pi(c) \nrightarrow \delta_{c^*} \quad \text{nor} \quad \pi(c_i) \nrightarrow \delta_{c_i^*} \quad \text{as } n \to \infty$$

Intuition: only one observation per 'parameter' c_i

Contrast with Bayesian variable selection, stochastic block model, graphical models, where concentration in discrete model space occurs.

[8] with convergence to $\delta_{(\boldsymbol{\theta}^*, \boldsymbol{w}^*)}$ in an appropriate sense, see e.g. Nguyen (2013); Guha et al. (2021)

Posterior case: statistical features of mixture model

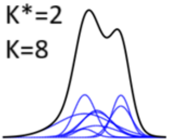
1. **Lack of posterior concentration**
2. **Flatness in the tails**: for random $c \sim \mathrm{Unif}([K]^n)$

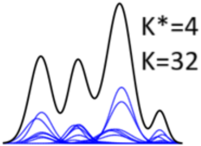
$$\frac{n_k(c)}{n_{k'}(c)+1}\frac{\pi(c_i = k' \mid c_{-i})}{\pi(c_i = k \mid c_{-i})} = 1 + O(n^{-1/2})$$

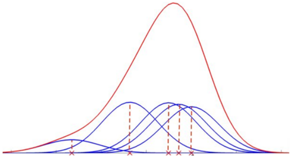
$\rightsquigarrow$ vanishing ‘discrete gradients’ in tails

Posterior case: statistical features of mixture model

1. **Lack of posterior concentration**
2. **Flatness in the tails**
3. **Overfitted or misspecified regimes**: mixture models often used in overfitted (i.e. K^* 'true' components with $K^* < K$; left figure) and misspecified (right figure) regimes







$\rightsquigarrow$ weakly identifiable and strongly overlapping clusters with

$$\frac{p(Y_i \mid Y_{-i}, c_{-i}, c_i = k_+)}{p(Y_i \mid Y_{-i}, c_{-i}, c_i = k_-)} \approx 1$$

Numerics: set-up

- **Parametric family:** 1d Gaussian mixture model

$$f_\theta(y) = N(y \mid \theta, 1), \quad p_0(\theta) = N(\theta \mid 0, 1)$$

- **Data**: generate $n = 1000$ data points from mixture with K^* components. Fit mixture with K components

Numerics: set-up

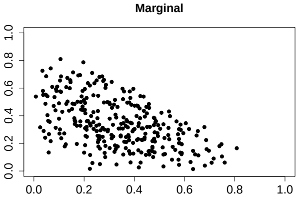
- **Parametric family:** 1d Gaussian mixture model

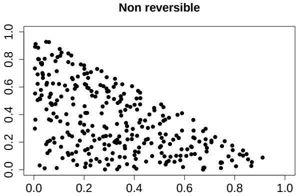
$$f_\theta(y) = N(y \mid \theta, 1), \quad p_0(\theta) = N(\theta \mid 0, 1)$$

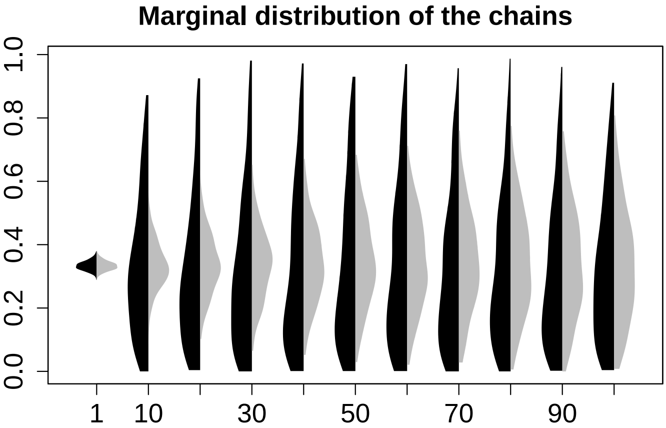
- **Data**: generate $n = 1000$ data points from mixture with K^* components. Fit mixture with K components
- Compare P_{MG} and P_{NR} through **prior-posterior check**:
  - Generate random datasets Y from the model distribution $p(Y)$
  - Sample from posterior $\pi(c) := p(c|Y)$ with MCMC
  - If chains reach convergence we **should recover the prior distribution** $p(c)$

First case: $K = K^* = 3$, $\alpha = 1$

Left: **final proportions** of the first two components after $100 \times n$ iterations $\Rightarrow$ Dirichlet$(1, 1, 1)$
Right: **evolution over time** with thinning of size n. Gray = P_{MG}, **Black** = P_{NR}



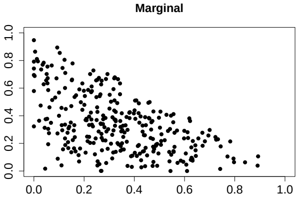


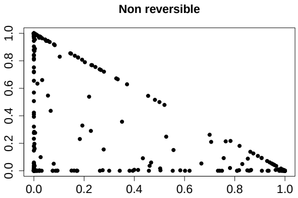


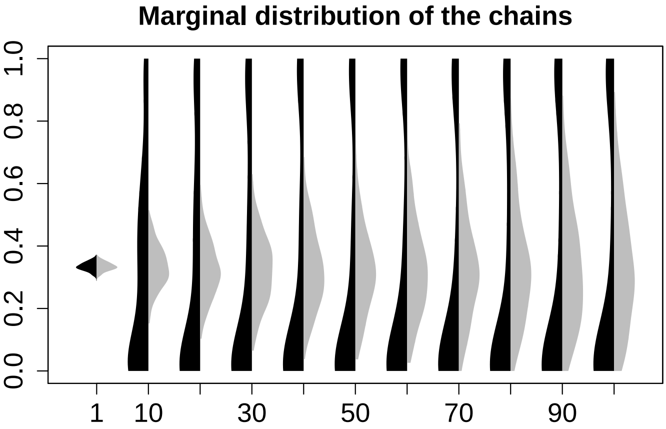
Second case: $K = K^* = 3$, $\alpha = 0.1$

Left: **final proportions** of the first two components after $100 \times n$ iterations $\Rightarrow$ Dirichlet$(0.1, 0.1, 0.1)$

Right: **evolution over time** with thinning of size n. Gray = P_{MG}, **Black** = P_{NR}







Dirichlet process mixtures

$$\boxed{Y_i \mid P \overset{\text{i.i.d.}}{\sim} Pf_\theta(\cdot) \quad i = 1, \ldots, n} \qquad P \sim \mathrm{DP}(\alpha, P_0).$$

$\mathrm{DP}(\alpha, P_0)$ = **Dirichlet process**, α = **concentration** parameter, P_0 = **baseline** distribution.

[9] Ruggiero and Walker (2009)

Dirichlet process mixtures

$$\boxed{Y_i \mid P \overset{\text{i.i.d.}}{\sim} Pf_\theta(\cdot) \quad i = 1, \dots, n} \qquad P \sim \mathrm{DP}(\alpha, P_0).$$

$\mathrm{DP}(\alpha, P_0)$ = **Dirichlet process**, α = **concentration** parameter, P_0 = **baseline** distribution.

Similar situation as before!

- The marginal sampler is even **more popular** (and often slow to converge).
- The prior case still admits a **scaling limit** with $\mathcal{O}(n^2)$ scaling factor[9].
- Wright-Fisher process $\to$ **Fleming-Viot process**.

[9] Ruggiero and Walker (2009)

Dirichlet process mixtures

The **non-reversible sampler** works as before! At each iteration:

1. Select a pair of clusters $(k, k') \leftarrow$ allowing to select a **new** one.
2. Propose a move **according to the direction** $v_{k,k'}$.
3. If rejected, flip $v_{k,k'}$.

Dirichlet process mixtures

The **non-reversible sampler** works as before! At each iteration:

1. Select a pair of clusters $(k, k') \leftarrow$ allowing to select a **new** one.
2. Propose a move **according to the direction** $v_{k,k'}$.
3. If rejected, flip $v_{k,k'}$.

Not discussed in this talk:

- Adjusting the non-reversible sampler to the **space of partitions** is not trivial!
- We need to allow **creation** and **elimination** of clusters.
- The selection probabilities must be chosen to preserve ergodicity.

Numerics: set-up

- **Parametric family:** 1d Gaussian mixture model

$$f_\theta(y) = N(y \mid \theta, 1), \quad P_0(\theta) = N(\theta \mid 0, 1)$$

- **Data**: generate $n = 1000$ data points from the associated Dirichlet Process Mixture model.

Numerics: set-up

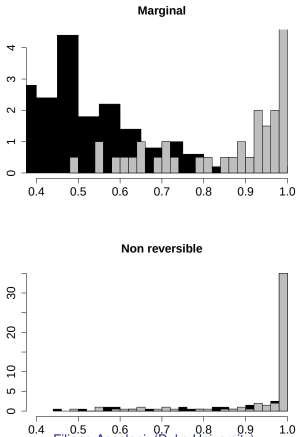
- **Parametric family:** 1d Gaussian mixture model

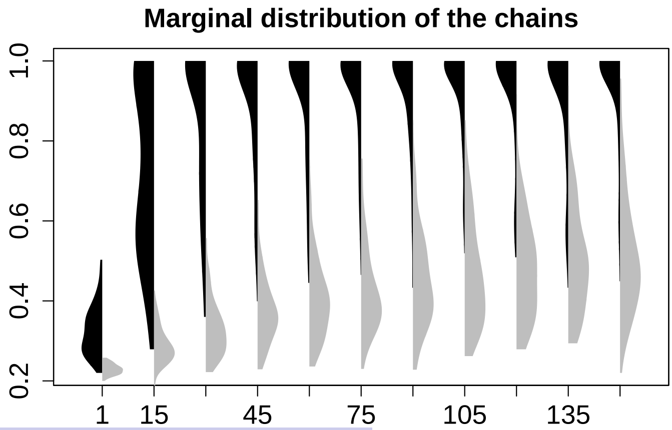
$$f_\theta(y) = N(y \mid \theta, 1), \quad P_0(\theta) = N(\theta \mid 0, 1)$$

- **Data**: generate $n = 1000$ data points from the associated Dirichlet Process Mixture model.
- Compare P_{MG} and P_{NR} through **prior-posterior check**:
  - Generate random datasets Y from the model distribution $p(Y)$
  - Sample from posterior $\pi(c) := p(c|Y)$ with MCMC
  - If chains reach convergence we **should recover the prior distribution.**
  - We focus on the distribution of the **largest cluster.**

Dirichlet process mixtures with $\alpha = 0.1$

Left: histogram of the **proportion of the largest cluster** after $100 \times n$ iterations, compared with the prior distribution.
Right: **evolution over time** with thinning of size n. Gray = P_{MG}, **Black** = P_{NR}





Practical takeaways

When should lifting help for mixture model samplers?

- Components **not well-separated**
- During **convergence phase**
- **Overfitted case** with $K > K^*$

Practical takeaways

When should lifting help for mixture model samplers?

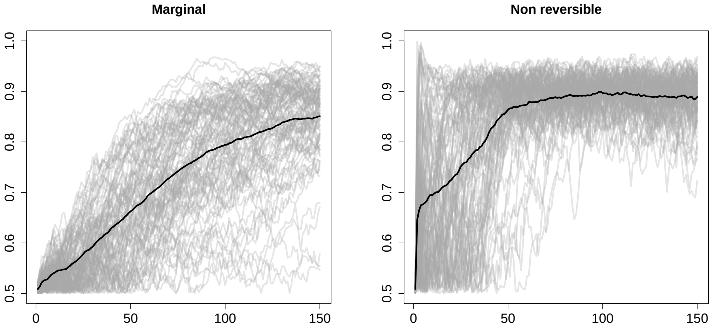
- Components **not well-separated**
- During **convergence phase**
- **Overfitted case** with $K > K^*$

Expect less improvement when

- Components are well-separated, $K = K^*$ and closer to convergence

Limitations (of methodology)

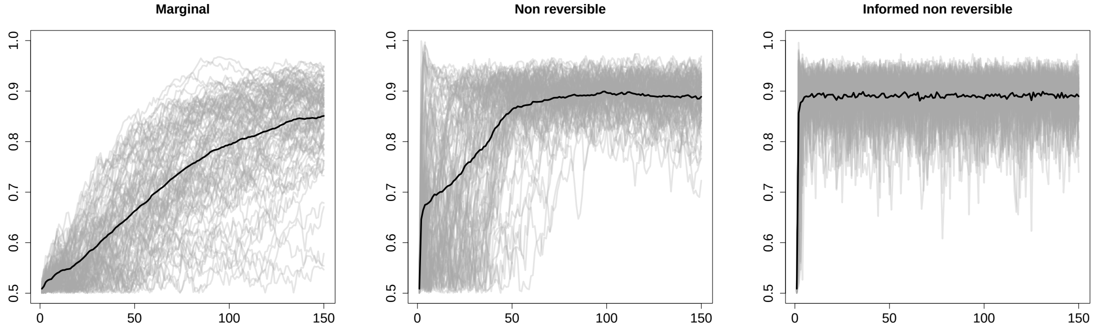
- For lifting to work well, **need to engineer directions with acceptance** ≈ 1
  $\rightsquigarrow$ not easy to do in general!
- Example: if $K = K^* = 2$ and components well-separated obtain



Can we improve?

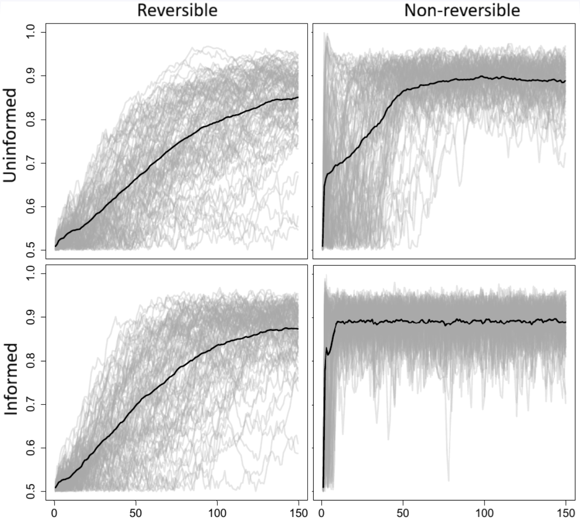
Informed versions

- Combining lifting with **informed proposals**[10] leads to MH acceptance$\approx$1
  $\rightsquigarrow$ allows to preserve momentum!

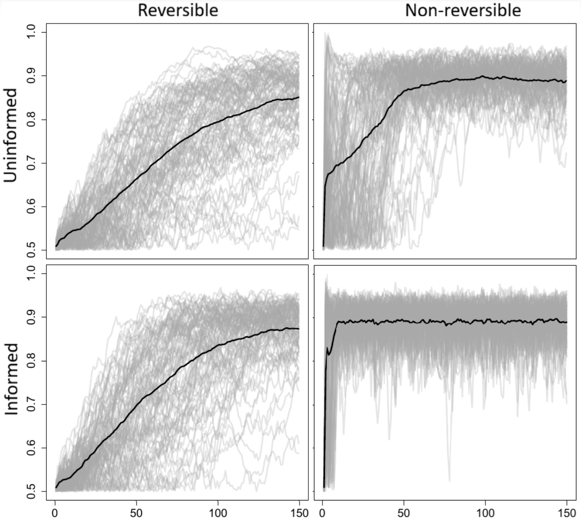


[10] Zanella (2020), Power and Goldman (2019), Gagnon and Maire (2024)

NB: here informed version only needed to increase acceptance and preserve momentum!



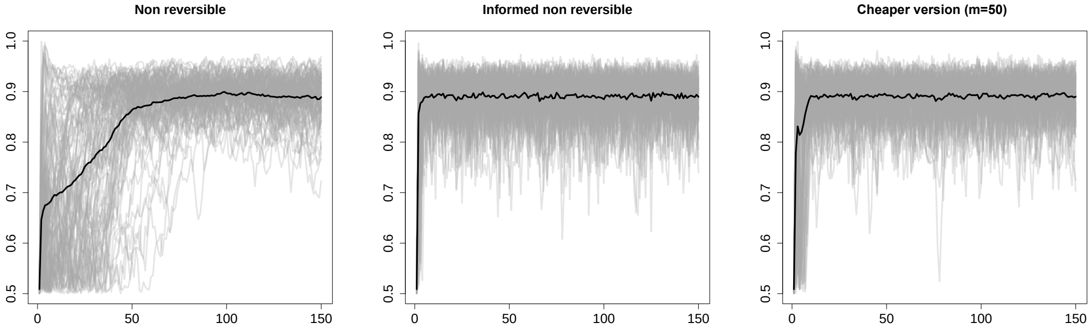
NB: here informed version only needed to increase acceptance and preserve momentum!



Issue: in general $O(n)$ cost per iteration. Can we do something cheaper?

Cheaper informed versions

- Sample **random neighborhood** of size m and use informed proposal therein[11]
- Moderate m (e.g. $m = 50$ with $n = 2000$) can be enough to make $\alpha \approx 1 \rightsquigarrow$ favourable trade-off



[11] similar to random neighborhood approach of Liang et al (2022) or informed multiple-try of Gagnon et al (2023)

Limitations (of theory)

We have

- **non-deterioration results** for any target
- **O(n) speedup** in prior case

We miss

- **quantify speedup in posterior case**?

Limitations (of theory)

We have

- **non-deterioration results** for any target
- **O(n) speedup** in prior case

We miss

- **quantify speedup in posterior case**?

  Current approach: scaling limits with data

  $\rightsquigarrow$ no exchangeability across $i \in [n]$ $\rightsquigarrow$ **measure-valued diffusion limit**

Conclusions

Summary:

- Standard reversible algorithms for mixture models can be **slow**
- We introduce a **non-reversible version** (simple to implement, **no extra cost**)
- Theoretically: **never slower**, $O(n)$ **speed-up in prior case**
- Empirically: **large speed-ups in posterior case**

Conclusions

Summary:

- Standard reversible algorithms for mixture models can be **slow**
- We introduce a **non-reversible version** (simple to implement, **no extra cost**)
- Theoretically: **never slower**, $O(n)$ **speed-up in prior case**
- Empirically: **large speed-ups in posterior case**

Many open problems:

- **Theory for posterior case**?
- **More robust approaches** to preserve momentum in general discrete spaces?
- Comparison with **Split-and-Merge** schemes?

Conclusions

Summary:

- Standard reversible algorithms for mixture models can be **slow**
- We introduce a **non-reversible version** (simple to implement, **no extra cost**)
- Theoretically: **never slower**, $O(n)$ **speed-up in prior case**
- Empirically: **large speed-ups in posterior case**

Many open problems:

- **Theory for posterior case**?
- **More robust approaches** to preserve momentum in general discrete spaces?
- Comparison with **Split-and-Merge** schemes?

Thanks for listening!