Langevin-based strategies for nested particle filters

Sara Pérez-Vieites

Aalto University
Finnish Center for Artificial Intelligence (FCAI)

International Conference on Monte Carlo Methods and Applications (MCM)

31 July 2025

Joint work with **Nicola Branchini**, **Víctor Elvira** (University of Edinburgh) and **Joaquín Míguez** (Universidad Carlos III de Madrid)

Index

Introduction

Nested particle filters
- Nested filtering
- Nested particle filter (NPF)

Gradient-based exploration of the parameter space

State-space model

We are interested in systems can be represented by **Markov state-space dynamical models**:

$$
\begin{aligned}
\textbf{(state)} \quad \boldsymbol{x}_t &= \boldsymbol{f}(\boldsymbol{x}_{t-1}, \boldsymbol{\theta}) + \boldsymbol{v}_t, \\
\textbf{(observation)} \quad \boldsymbol{y}_t &= \boldsymbol{g}(\boldsymbol{x}_t, \boldsymbol{\theta}) + \boldsymbol{r}_t,
\end{aligned}
$$

In terms of a set of **relevant probability density functions (pdfs)**:

$$\boldsymbol{\theta} \sim p(\boldsymbol{\theta}) \quad \text{and} \quad \boldsymbol{x}_0 \sim p(\boldsymbol{x}_0) \tag{1}$$

$$\boldsymbol{x}_t \sim p(\boldsymbol{x}_t | \boldsymbol{x}_{t-1}, \boldsymbol{\theta}) \tag{2}$$

$$\boldsymbol{y}_t \sim p(\boldsymbol{y}_t | \boldsymbol{x}_t, \boldsymbol{\theta}) \tag{3}$$

State-space model

We are interested in systems can be represented by **Markov state-space dynamical models**:

$$
\begin{aligned}
\textbf{(state)} \quad \boldsymbol{x}_t &= \boldsymbol{f}(\boldsymbol{x}_{t-1}, \boldsymbol{\theta}) + \boldsymbol{v}_t, \\
\textbf{(observation)} \quad \boldsymbol{y}_t &= \boldsymbol{g}(\boldsymbol{x}_t, \boldsymbol{\theta}) + \boldsymbol{r}_t,
\end{aligned}
$$

In terms of a set of **relevant probability density functions (pdfs)**:

$$
\boldsymbol{\theta} \sim p(\boldsymbol{\theta}) \quad \text{and} \quad \boldsymbol{x}_0 \sim p(\boldsymbol{x}_0) \tag{1}
$$

$$
\boldsymbol{x}_t \sim p(\boldsymbol{x}_t | \boldsymbol{x}_{t-1}, \boldsymbol{\theta}) \tag{2}
$$

$$
\boldsymbol{y}_t \sim p(\boldsymbol{y}_t | \boldsymbol{x}_t, \boldsymbol{\theta}) \tag{3}
$$

Goal

$\longrightarrow$ We want to **approximate the joint posterior distribution of** θ **and** $\boldsymbol{x}_t$, i.e., $p(\boldsymbol{x}_t, \theta | \boldsymbol{y}_{1:t})$.

$\longrightarrow$ For a long sequence of observations, i.e., **online**.

State-of-the-art methods

Methods for Bayesian inference of both $\boldsymbol{\theta}$ and $\boldsymbol{x}_t$:

- **particle Markov chain Monte Carlo (PMCMC)**[1]
- **sequential Monte Carlo square (SMC^2)**[2]
- **nested particle filters (NPFs)**[3]

⟶ They can quantify the uncertainty or estimation error.
⟶ They can be applied to a broad class of models.
⟶ They provide theoretical guarantees.
⟶ Both PMCMC and SMC^2 are batch techniques, while the NPF is a recursive method.

[1] Andrieu, Doucet, and Holenstein 2010.
[2] Chopin, Jacob, and Papaspiliopoulos 2013.
[3] Crisan and Míguez 2018.

State-of-the-art methods

Methods for Bayesian inference of both θ and x_t:

- **particle Markov chain Monte Carlo (PMCMC)**[1]
- **sequential Monte Carlo square (SMC^2)**[2]
- **nested particle filters (NPFs)**[3]

⟶ They can quantify the uncertainty or estimation error.
⟶ They can be applied to a broad class of models.
⟶ They provide theoretical guarantees.
⟶ Both PMCMC and SMC^2 are batch techniques, while the NPF is a recursive method.

[1] Andrieu, Doucet, and Holenstein 2010.
[2] Chopin, Jacob, and Papaspiliopoulos 2013.
[3] Crisan and Míguez 2018.

State-of-the-art methods

Methods for Bayesian inference of both $\boldsymbol{\theta}$ and $\boldsymbol{x}_t$:

- **particle Markov chain Monte Carlo (PMCMC)**[1]
- **sequential Monte Carlo square (SMC^2)**[2]
- **nested particle filters (NPFs)**[3]

$\longrightarrow$ They can quantify the uncertainty or estimation error.
$\longrightarrow$ They can be applied to a broad class of models.
$\longrightarrow$ They provide theoretical guarantees.
$\longrightarrow$ Both PMCMC and SMC^2 are batch techniques, while the NPF is a recursive method.

[1] Andrieu, Doucet, and Holenstein 2010.
[2] Chopin, Jacob, and Papaspiliopoulos 2013.
[3] Crisan and Míguez 2018.

Index

Introduction

Nested particle filters
- Nested filtering
- Nested particle filter (NPF)

Gradient-based exploration of the parameter space

Nested filtering

We aim at computing the joint posterior pdf $p(\boldsymbol{\theta}, \boldsymbol{x}_t | \boldsymbol{y}_{1:t})$, as

$$p(\boldsymbol{x}_t, \boldsymbol{\theta} | \boldsymbol{y}_{1:t}) = \underbrace{p(\boldsymbol{x}_t | \boldsymbol{\theta}, \boldsymbol{y}_{1:t})}_{2^{nd} \text{ layer}} \underbrace{p(\boldsymbol{\theta} | \boldsymbol{y}_{1:t})}_{1^{st} \text{ layer}}$$

Nested filtering

At every time step t:

1st layer

$$\underbrace{p(\boldsymbol{\theta}|\boldsymbol{y}_{1:t-1})}_{\text{Pred. pdf of } \boldsymbol{\theta}}$$

2nd layer

$$p(\boldsymbol{y}_t|\boldsymbol{\theta}, \boldsymbol{y}_{1:t-1})$$

$$\underbrace{p(\boldsymbol{\theta}|\boldsymbol{y}_{1:t})}_{\text{Post. pdf of } \boldsymbol{\theta}} \propto p(\boldsymbol{y}_t|\boldsymbol{\theta}, \boldsymbol{y}_{1:t-1})\, p(\boldsymbol{\theta}|\boldsymbol{y}_{1:t-1})$$

Nested filtering

At every time step t:

1st layer

$$\underbrace{p(\boldsymbol{\theta}|\boldsymbol{y}_{1:t-1})}_{\text{Pred. pdf of } \boldsymbol{\theta}}$$

2nd layer

$$p(\boldsymbol{y}_t|\boldsymbol{\theta}, \boldsymbol{y}_{1:t-1}) = \int p(\boldsymbol{y}_t|\boldsymbol{x}_t, \boldsymbol{\theta}) p(\boldsymbol{x}_t|\boldsymbol{\theta}, \boldsymbol{y}_{1:t-1}) d\boldsymbol{x}_t$$

$$\underbrace{p(\boldsymbol{\theta}|\boldsymbol{y}_{1:t})}_{\text{Post. pdf of } \boldsymbol{\theta}} \propto p(\boldsymbol{y}_t|\boldsymbol{\theta}, \boldsymbol{y}_{1:t-1}) p(\boldsymbol{\theta}|\boldsymbol{y}_{1:t-1})$$

Nested filtering

At every time step t:

1^{st} layer

$\underbrace{p(\boldsymbol{\theta}|\boldsymbol{y}_{1:t-1})}_{\text{Pred. pdf of }\boldsymbol{\theta}}$

Classical filtering problem (given $\boldsymbol{\theta}$)

Predictive pdf of $\boldsymbol{x}$: $p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t-1},\boldsymbol{\theta})$

Likelihood: $p(\boldsymbol{y}_t|\boldsymbol{x}_t,\boldsymbol{\theta})$

Posterior pdf of $\boldsymbol{x}$: $p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t},\boldsymbol{\theta})$

2^{nd} layer

$$p(\boldsymbol{y}_t|\boldsymbol{\theta},\boldsymbol{y}_{1:t-1}) = \int p(\boldsymbol{y}_t|\boldsymbol{x}_t,\boldsymbol{\theta})p(\boldsymbol{x}_t|\boldsymbol{\theta},\boldsymbol{y}_{1:t-1})d\boldsymbol{x}_t$$

$$\underbrace{p(\boldsymbol{\theta}|\boldsymbol{y}_{1:t})}_{\text{Post. pdf of }\boldsymbol{\theta}} \propto p(\boldsymbol{y}_t|\boldsymbol{\theta},\boldsymbol{y}_{1:t-1})p(\boldsymbol{\theta}|\boldsymbol{y}_{1:t-1})$$

Nested filtering

At every time step t:

1^{st} layer

$\underbrace{p(\boldsymbol{\theta}|\boldsymbol{y}_{1:t-1})}_{\text{Pred. pdf of } \boldsymbol{\theta}}$

Filtering (given $\boldsymbol{\theta}$)

Predictive pdf of $\boldsymbol{x}$: $p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t-1}, \boldsymbol{\theta})$

Likelihood: $p(\boldsymbol{y}_t|\boldsymbol{x}_t, \boldsymbol{\theta})$

Posterior pdf of $\boldsymbol{x}$: $p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t}, \boldsymbol{\theta})$

2^{nd} layer

$$p(\boldsymbol{y}_t|\boldsymbol{\theta}, \boldsymbol{y}_{1:t-1}) = \int p(\boldsymbol{y}_t|\boldsymbol{x}_t, \boldsymbol{\theta})p(\boldsymbol{x}_t|\boldsymbol{\theta}, \boldsymbol{y}_{1:t-1})d\boldsymbol{x}_t$$

$$\underbrace{p(\boldsymbol{\theta}|\boldsymbol{y}_{1:t})}_{\text{Post. pdf of } \boldsymbol{\theta}} \propto p(\boldsymbol{y}_t|\boldsymbol{\theta}, \boldsymbol{y}_{1:t-1})p(\boldsymbol{\theta}|\boldsymbol{y}_{1:t-1})$$

Nested particle filter (NPF)[4]

SMC (N samples) to approximate $p(\boldsymbol{\theta}|\boldsymbol{y}_{1:t})$

For $i = 1, \ldots, N$:

- Jittering: Draw $\bar{\boldsymbol{\theta}}_t^i \sim \kappa_N(d\boldsymbol{\theta}|\boldsymbol{\theta}_{t-1}^i)$

  SMC (M samples) to approximate $p(\boldsymbol{y}_t|\boldsymbol{y}_{1:t-1}, \bar{\boldsymbol{\theta}}_t^i)$

  Given $\bar{\boldsymbol{\theta}}_t^i$, for $j = 1, \ldots, M$:

  - Draw $\bar{\boldsymbol{x}}_t^{i,j} \sim p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}^{i,j}, \bar{\boldsymbol{\theta}}_t^i)$
  - Weights: $\tilde{u}_t^{i,j} \propto p(\boldsymbol{y}_t|\bar{\boldsymbol{x}}_t^{i,j}, \bar{\boldsymbol{\theta}}_t^i)$
  - Resampling: for $m = 1, \ldots, M$, $\tilde{\boldsymbol{x}}_t^{i,j} = \bar{\boldsymbol{x}}_t^{i,m}$ with probability $u_t^{i,m} = \frac{\tilde{u}_t^{i,m}}{\sum_{j=1}^M \tilde{u}_t^{i,j}}$

- Weights of $\bar{\boldsymbol{\theta}}_t^i$: $\tilde{w}_t^i = \frac{1}{M}\sum_{j=1}^M \tilde{u}_t^{i,j}$
- Resampling: for $l = 1, \ldots, N$, $\{\boldsymbol{\theta}_t^i, \{\boldsymbol{x}_t^{i,j}\}_{1\leq j\leq M}\} = \{\bar{\boldsymbol{\theta}}_t^l, \{\tilde{\boldsymbol{x}}_t^{l,j}\}_{1\leq j\leq M}\}$ with prob. w_t^l, so that $p(\boldsymbol{\theta}|\boldsymbol{y}_{1:t}) = \frac{1}{N}\sum_{i=1}^N \delta_{\boldsymbol{\theta}_t^i}(d\boldsymbol{\theta})$

[4] Crisan and Miguez 2017.

Jittering

- NPF $\longrightarrow$ jittering: $\bar{\boldsymbol{\theta}}_t^i \sim \kappa_N(d\boldsymbol{\theta}|\boldsymbol{\theta}')$, where

$$\kappa_N(d\boldsymbol{\theta}|\boldsymbol{\theta}') = (1-\epsilon_N)\delta_{\boldsymbol{\theta}'}(\boldsymbol{\theta}) + \epsilon_N\kappa(d\boldsymbol{\theta}|\boldsymbol{\theta}')$$

  - $0 < \epsilon_N \leq \frac{1}{\sqrt{N}}$
  - $\kappa(d\boldsymbol{\theta}|\boldsymbol{\theta}')$ is an arbitrary Markov kernel with mean $\boldsymbol{\theta}'$ and finite variance, e.g., $\kappa(d\boldsymbol{\theta}|\boldsymbol{\theta}') = \mathcal{N}(\boldsymbol{\theta}|\boldsymbol{\theta}', \tilde{\sigma}^2\boldsymbol{I})$, with $\tilde{\sigma}^2 < \infty$.

- Guarantees convergence to the true posterior when $N \longrightarrow \infty$

Take-aways

Advantages:

- Only framework that is *online* and *Bayesian* on $\boldsymbol{\theta}$
- Applicable to general parametric state-space models
- Asymptotic convergence guarantees

Limitations:

- Covergence speed might be slow (depends on the jittering (hyper)parameters)
- This problem gets worse as the dimension of θ increases

Take-aways

Advantages:

- Only framework that is *online* and *Bayesian* on $\boldsymbol{\theta}$
- Applicable to general parametric state-space models
- Asymptotic convergence guarantees

Limitations:

- Covergence speed might be slow (depends on the jittering (hyper)parameters)
- This problem gets worse as the dimension of $\boldsymbol{\theta}$ increases

Index

Introduction

Nested particle filters
- Nested filtering
- Nested particle filter (NPF)

Gradient-based exploration of the parameter space

Langevin nested particle filter (LNPF)

Iteratively, move the parameters towards areas of higher probability with the Unadjusted Langevin algorithm (ULA):

$$\boldsymbol{\theta}_{t,k+1}^{l} = \boldsymbol{\theta}_{t,k}^{l} + \gamma_k \cdot \nabla_{\boldsymbol{\theta}} \log p(\boldsymbol{\theta} \mid \boldsymbol{y}_{1:t})\Big|_{\boldsymbol{\theta}=\boldsymbol{\theta}_{t,k}^{l}} + \sqrt{2\gamma_k}\boldsymbol{v}_k, \quad (4)$$

where $\boldsymbol{v}_k \sim \mathcal{N}(0, \boldsymbol{I}_{d_\theta})$, $\gamma_k > 0$ is a step size sequence.

We replace jittering by ULA:

$$\text{(jittering)} \quad \bar{\theta}_t^i \sim \kappa_{NPF}(d\theta|\theta') = (1-\epsilon_N)\delta_{\theta'}(\theta) + \epsilon_N \, \kappa_{jitter}(d\theta|\theta') \quad (5)$$

$$\text{(ULA)} \quad \bar{\theta}_t^i \sim \kappa_{LNPF}(d\theta|\theta') = (1-\epsilon_N)\delta_{\theta'}(\theta) + \epsilon_N \, \kappa_{ULA}(d\theta|\theta') \quad (6)$$

*(ULA needs to adjust number of iterations and step size)

Langevin nested particle filter (LNPF)

Iteratively, move the parameters towards areas of higher probability with the Unadjusted Langevin algorithm (ULA):

$$\boldsymbol{\theta}^{l}_{t,k+1} = \boldsymbol{\theta}^{l}_{t,k} + \gamma_k \cdot \nabla_{\boldsymbol{\theta}} \log p(\boldsymbol{\theta} \mid \boldsymbol{y}_{1:t})\Big|_{\boldsymbol{\theta}=\boldsymbol{\theta}^{l}_{t,k}} + \sqrt{2\gamma_k}\boldsymbol{v}_k, \quad (4)$$

where $\boldsymbol{v}_k \sim \mathcal{N}(0, \boldsymbol{I}_{d_\theta})$, $\gamma_k > 0$ is a step size sequence.

We replace jittering by ULA:

$$\text{(jittering)} \quad \bar{\boldsymbol{\theta}}^{i}_{t} \sim \kappa_{NPF}(d\boldsymbol{\theta}|\boldsymbol{\theta}') = (1-\epsilon_N)\delta_{\boldsymbol{\theta}'}(\boldsymbol{\theta}) + \epsilon_N \, \kappa_{jitter}(d\boldsymbol{\theta}|\boldsymbol{\theta}') \quad (5)$$

$$\text{(ULA)} \quad \bar{\boldsymbol{\theta}}^{i}_{t} \sim \kappa_{LNPF}(d\boldsymbol{\theta}|\boldsymbol{\theta}') = (1-\epsilon_N)\delta_{\boldsymbol{\theta}'}(\boldsymbol{\theta}) + \epsilon_N \, \kappa_{ULA}(d\boldsymbol{\theta}|\boldsymbol{\theta}') \quad (6)$$

*(ULA needs to adjust number of iterations and step size)

Langevin nested particle filter (LNPF)

Iteratively, move the parameters towards areas of higher probability with the Unadjusted Langevin algorithm (ULA):

$$\boldsymbol{\theta}^l_{t,k+1} = \boldsymbol{\theta}^l_{t,k} + \gamma_k \cdot \nabla_{\boldsymbol{\theta}} \log p(\boldsymbol{\theta} \mid \boldsymbol{y}_{1:t})\Big|_{\boldsymbol{\theta}=\boldsymbol{\theta}^l_{t,k}} + \sqrt{2\gamma_k}\boldsymbol{v}_k, \quad (4)$$

where $\boldsymbol{v}_k \sim \mathcal{N}(0, \boldsymbol{I}_{d_\theta})$, $\gamma_k > 0$ is a step size sequence.

We replace jittering by ULA:

$$\text{(jittering)} \quad \bar{\boldsymbol{\theta}}^i_t \sim \kappa_{NPF}(d\boldsymbol{\theta}|\boldsymbol{\theta}') = (1-\epsilon_N)\delta_{\boldsymbol{\theta}'}(\boldsymbol{\theta}) + \epsilon_N \, \kappa_{jitter}(d\boldsymbol{\theta}|\boldsymbol{\theta}') \quad (5)$$

$$\text{(ULA)} \quad \bar{\boldsymbol{\theta}}^i_t \sim \kappa_{LNPF}(d\boldsymbol{\theta}|\boldsymbol{\theta}') = (1-\epsilon_N)\delta_{\boldsymbol{\theta}'}(\boldsymbol{\theta}) + \epsilon_N \, \kappa_{ULA}(d\boldsymbol{\theta}|\boldsymbol{\theta}') \quad (6)$$

*(ULA needs to adjust number of iterations and step size)

Challenges: intractable score

The gradient of interest includes the score, that is intractable and its computation is not recursive.

$$\nabla_{\boldsymbol{\theta}} \log p(\boldsymbol{\theta} \mid \boldsymbol{y}_{1:t}) = \nabla_{\boldsymbol{\theta}} \log p(\boldsymbol{y}_{1:t}|\boldsymbol{\theta}) + \nabla_{\boldsymbol{\theta}} \log p(\boldsymbol{\theta}) \quad (7)$$

$\longrightarrow$ Using Fisher's identity we have two approximations[5]

$\mathcal{O}(N)$: $\nabla_\theta \log p(\boldsymbol{y}_{1:t} \mid \theta) = \int \nabla_\theta \log p(\boldsymbol{y}_{1:t}, \boldsymbol{x}_{1:t} \mid \theta) p(\boldsymbol{x}_{1:t} \mid \boldsymbol{y}_{1:t}, \theta) d\boldsymbol{x}_{1:t}$

$\mathcal{O}(N^2)$: $\nabla_\theta \log p(\boldsymbol{y}_{1:t} \mid \theta) = \int \nabla_\theta \log p(\boldsymbol{y}_{1:t}, \boldsymbol{x}_t \mid \theta) p(\boldsymbol{x}_t \mid \boldsymbol{y}_{1:t}, \theta) d\boldsymbol{x}_t$

*(This is described for a fixed θ)

[5] Poyiadjis, Doucet, and Singh 2011.

Challenges: intractable score

The gradient of interest includes the score, that is intractable and its computation is not recursive.

$$\nabla_{\boldsymbol{\theta}} \log p(\boldsymbol{\theta} \mid \boldsymbol{y}_{1:t}) = \nabla_{\boldsymbol{\theta}} \log p(\boldsymbol{y}_{1:t}|\boldsymbol{\theta}) + \nabla_{\boldsymbol{\theta}} \log p(\boldsymbol{\theta}) \quad (7)$$

$\longrightarrow$ Using Fisher's identity we have two approximations[5]

$\mathcal{O}(N)$: $$\nabla_{\boldsymbol{\theta}} \log p(\boldsymbol{y}_{1:t} \mid \boldsymbol{\theta}) = \int \nabla_{\boldsymbol{\theta}} \log p(\boldsymbol{y}_{1:t}, \boldsymbol{x}_{1:t} \mid \boldsymbol{\theta}) p(\boldsymbol{x}_{1:t} \mid \boldsymbol{y}_{1:t}, \boldsymbol{\theta}) d\boldsymbol{x}_{1:t}$$

$\mathcal{O}(N^2)$: $$\nabla_{\boldsymbol{\theta}} \log p(\boldsymbol{y}_{1:t} \mid \boldsymbol{\theta}) = \int \nabla_{\boldsymbol{\theta}} \log p(\boldsymbol{y}_{1:t}, \boldsymbol{x}_{t} \mid \boldsymbol{\theta}) p(\boldsymbol{x}_{t} \mid \boldsymbol{y}_{1:t}, \boldsymbol{\theta}) d\boldsymbol{x}_{t}$$

*(This is described for a fixed θ)

[5] Poyiadjis, Doucet, and Singh 2011.

Challenges: intractable score

The gradient of interest includes the score, that is intractable and its computation is not recursive.

$$\nabla_{\boldsymbol{\theta}} \log p(\boldsymbol{\theta} \mid \boldsymbol{y}_{1:t}) = \nabla_{\boldsymbol{\theta}} \log p(\boldsymbol{y}_{1:t}|\boldsymbol{\theta}) + \nabla_{\boldsymbol{\theta}} \log p(\boldsymbol{\theta}) \quad (7)$$

$\longrightarrow$ Using Fisher's identity we have two approximations[5]

$\mathcal{O}(N)$: $$\nabla_{\boldsymbol{\theta}} \log p(\boldsymbol{y}_{1:t} \mid \boldsymbol{\theta}) = \int \nabla_{\boldsymbol{\theta}} \log p(\boldsymbol{y}_{1:t}, \boldsymbol{x}_{1:t} \mid \boldsymbol{\theta}) p(\boldsymbol{x}_{1:t} \mid \boldsymbol{y}_{1:t}, \boldsymbol{\theta}) d\boldsymbol{x}_{1:t}$$

$\mathcal{O}(N^2)$: $$\nabla_{\boldsymbol{\theta}} \log p(\boldsymbol{y}_{1:t} \mid \boldsymbol{\theta}) = \int \nabla_{\boldsymbol{\theta}} \log p(\boldsymbol{y}_{1:t}, \boldsymbol{x}_{t} \mid \boldsymbol{\theta}) p(\boldsymbol{x}_{t} \mid \boldsymbol{y}_{1:t}, \boldsymbol{\theta}) d\boldsymbol{x}_{t}$$

*(This is described for a fixed $\boldsymbol{\theta}$)

[5] Poyiadjis, Doucet, and Singh 2011.

Challenges: recursive approximation

$\longrightarrow$ Approximation used in the recursive maximum likelihood literature[6], such that

$$\nabla_{\boldsymbol{\theta}} \log p(\boldsymbol{y}_{1:t}|\boldsymbol{\theta}) \text{ is replaced by } \nabla_{\boldsymbol{\theta}} \log p(\boldsymbol{y}_t \mid \boldsymbol{y}_{1:t-1}, \boldsymbol{\theta}),$$

and it can be computed as

$$\nabla_\theta \log p(\boldsymbol{y}_t \mid \boldsymbol{y}_{1:t-1}, \theta)\Big|_{\theta=\theta_t} = \nabla_\theta \log p(\boldsymbol{y}_{1:t}|\theta)\Big|_{\theta=\boldsymbol{\theta}_{1:t}} - \nabla_\theta \log p(\boldsymbol{y}_{1:t-1}|\theta)\Big|_{\theta=\boldsymbol{\theta}_{1:t-1}}$$

*(We are assuming that process $\boldsymbol{y}_t$ is ergodic)

[6] Chopin, Papaspiliopoulos, et al. 2020.

Challenges: recursive approximation

⟶ Approximation used in the recursive maximum likelihood literature[6], such that

$$\nabla_{\boldsymbol{\theta}} \log p(\boldsymbol{y}_{1:t}|\boldsymbol{\theta}) \quad \text{is replaced by} \quad \nabla_{\boldsymbol{\theta}} \log p(\boldsymbol{y}_t \mid \boldsymbol{y}_{1:t-1}, \boldsymbol{\theta}),$$

and it can be computed as

$$\nabla_{\boldsymbol{\theta}} \log p(\boldsymbol{y}_t \mid \boldsymbol{y}_{1:t-1}, \boldsymbol{\theta})\Big|_{\boldsymbol{\theta}=\boldsymbol{\theta}_t} = \nabla_{\boldsymbol{\theta}} \log p(\boldsymbol{y}_{1:t}|\boldsymbol{\theta})\Big|_{\boldsymbol{\theta}=\boldsymbol{\theta}_{1:t}} - \nabla_{\boldsymbol{\theta}} \log p(\boldsymbol{y}_{1:t-1}|\boldsymbol{\theta})\Big|_{\boldsymbol{\theta}=\boldsymbol{\theta}_{1:t-1}}$$

*(We are assuming that process $\boldsymbol{y}_t$ is ergodic)

[6] Chopin, Papaspiliopoulos, et al. 2020.

Challenges: recursive approximation

$\longrightarrow$ Approximation used in the recursive maximum likelihood literature[6], such that

$$\nabla_{\boldsymbol{\theta}} \log p(\boldsymbol{y}_{1:t}|\boldsymbol{\theta}) \text{ is replaced by } \nabla_{\boldsymbol{\theta}} \log p(\boldsymbol{y}_t \mid \boldsymbol{y}_{1:t-1}, \boldsymbol{\theta}),$$

and it can be computed as

$$\nabla_{\boldsymbol{\theta}} \log p(\boldsymbol{y}_t \mid \boldsymbol{y}_{1:t-1}, \boldsymbol{\theta})\Big|_{\boldsymbol{\theta}=\boldsymbol{\theta}_t} = \nabla_{\boldsymbol{\theta}} \log p(\boldsymbol{y}_{1:t}|\boldsymbol{\theta})\Big|_{\boldsymbol{\theta}=\boldsymbol{\theta}_{1:t}} - \nabla_{\boldsymbol{\theta}} \log p(\boldsymbol{y}_{1:t-1}|\boldsymbol{\theta})\Big|_{\boldsymbol{\theta}=\boldsymbol{\theta}_{1:t-1}}$$

*(We are assuming that process $\boldsymbol{y}_t$ is ergodic)

[6] Chopin, Papaspiliopoulos, et al. 2020.

Conclusions

- Jittering can be very inefficient, especially with larger d_{θ}
- We proposed ULA updates for smarter exploration of θ space
- Challenge is approximating the score online and accurately

References

- Crisan & Míguez (2018), *Nested particle filters for online parameter estimation in discrete-time state-space Markov models*. Bernoulli, vol. 24, no. 4A, pp. 3039–3086.
- Chopin, N., Jacob, P. E., & Papaspiliopoulos, O. (2013). *SMC^2: an efficient algorithm for sequential analysis of state space models.* Journal of the Royal Statistical Society Series B: Statistical Methodology, 75(3), 397-426.
- Andrieu, Christophe, Arnaud Doucet, & Roman Holenstein (2010), *Particle Markov chain Monte Carlo methods.* Journal of the Royal Statistical Society Series B: Statistical Methodology 72.3 : 269-342.
- Poyiadjis, G., Doucet, A., & Singh, S. S. (2011). *Particle approximations of the score and observed information matrix in state space models with application to parameter estimation.* Biometrika, 98(1), 65-80.
- Chopin, N., & Papaspiliopoulos, O. (2020). *An introduction to sequential Monte Carlo* (Vol. 4). Cham, Switzerland: Springer.

Thank you!

https://sarapv.github.io/