Automatic Differentiation to Accelerate Inference for Partially Observed Stochastic Processes

Kevin Tan, Giles J. Hooker, Edward L. Ionides

Abstract

Automatic differentiation (AD) has driven recent advances in machine learning, including deep neural networks and Hamiltonian Markov Chain Monte Carlo methods. Partially observed nonlinear stochastic dynamic systems have proved resistant to AD techniques due to (1) the requirement for simulation-based inference that does not require access to the system's transition probabilities and (2) the issue that widely used particle filter algorithms yield a likelihood function that is discontinuous as a function of the model parameters. We propose a new representation that embeds existing methods in a theoretical framework that readily permits extension to a new class of algorithms, providing opportunities for optimizing a bias/variance tradeoff. Further, we develop optimization algorithms suited to the Monte Carlo properties of the derivative estimate, including a hybrid algorithm that requires only a differentiable simulator for maximum likelihood estimation. Promising numerical results indicate that a hybrid algorithm that uses AD to refine a coarse solution from iterated filtering can beat current state-of-the-art methods on a challenging scientific benchmark problem.

1 Introduction

Maximum-likelihood estimation for inference in partially observed stochastic processes, also known as, continuous-state continuous-time hidden Markov models (HMMs), partially-observed Markov processes (POMPs), or partially-observed stochastic nonlinear dynamical systems, is a challenging problem that faces several challenges:

1. Intractable likelihood functions, bypassed with the approaches of simulated likelihood and likelihood-free inference.
2. The desire for simulation-based inference without access to the transition density of the latent Markov process, with various methods created by **?????**.
3. Accurate parameter estimation in the presence of significant Monte-Carlo noise in the likelihood estimate.

Many approaches to inference in these scenarios (such as the EM algorithm, the various Kalman filter variants, and MCMC) either struggle with intractable likelihood functions when the models are complex, or assume access to the probability density of next states given the current state. This is a problem in some critical applications, such as disease modeling, where the models are complex enough such that even obtaining the transition density is an intractable problem. However, the particle filter with the bootstrap proposal, a popular method for solving the filtering problem in partially-observed dynamical systems, provides an unbiased estimate of the likelihood (**?**) without requiring evaluation of the transition density of the latent Markov process, enabling an arbitrary model simulator to be plugged into the algorithm.

This is the backbone behind the only simulation-based full-information maximum likelihood method available in the literature thus far, the improved iterated filtering algorithm of **?**. **?** utilize the bootstrap filter for perform maximum likelihood estimation with an iterated perturbed Bayes map, successfully tackling challenging problems in epidemiology that available Bayesian methodology fails to solve.

Still, maximum likelihood parameter estimation can be challenging, especially when the Monte Carlo variance of the evaluation is high and the number of parameters is not small. For example, though in practice the algorithm of **?** converges quickly to a neighborhood of the MLE, it often struggles to successfully optimize the last few-units of the log-likelihood. **?** attempt to address the third problem, albeit with a computationally-expensive approach.

1.1 Automatic Differentiation for Particle Filters

Recent advances in automatic differentiation (AD) for particle filters (**?????**) have drawn attention to AD as an alternative tool for inference in partially observed stochastic processes with the particle filter. However, these existing approaches either are not yet compatible with the desire for simulation-based inference, or are computationally expensive.

? and **?** drop high-variance resampling terms from the gradient, leading to bias that does not vanish asymptotically according to **?**. Though the approach of **?** is promising, it is computationally expensive. **?** provide another promising fixed-lag smoothing-based approach that is similar in spirit to our method (MOP-α), but requires either (1) access to the process model transition densities or (2) the special case where the transition density factors into a policy that selects actions (in the reinforcement learning sense) and a deterministic model that maps states and actions to next-timestep states.

Finally, **?** demonstrate that the estimators derived in **?** can be attained with AD, albeit with a simple tweak to the particle filter that does not modify its likelihood estimates. Though their method is not fully compatible with simulation-based inference, we show that another simple tweak enables this compatibility and corresponds to a special case of our MOP-α algorithm.

1.2 Our Contributions

Previous research on AD for particle filters has struggled with various issues:

1. Bias, if the discontinuous nature of particle resampling is ignored.
2. High Monte Carlo variance, if continuity corrections lead to numerical instability.
3. High computational cost, for algorithms which involve pairwise interactions between particles, marginalization, smoothing or optimal transport.
4. Reduced applicability for algorithms which lose or fail to take advantage of the simulation-based capabilities of the bootstrap filter.

We develop a new approach to statistical inference via AD for particle filters which addresses these concerns. We develop a new theoretical framework that encompasses existing methods, and addresses the seemingly incompatible paradox of differentiating through a Monte-Carlo algorithm with discontinuous resampling.

This is done through the construction of a smooth extension to the particle filter, as well as a **Measurement Off-Policy-α** (MOP-α) family of algorithms that (informally) encompasses the special case of AD of a vanilla PF (recovering the gradient estimator of **?**) when $\alpha = 0$, and the DPF algorithm from **?** (that recovers the gradient estimator from **?**) when $\alpha = 1$. When $\alpha \in (0, 1)$, this provides opportunities for optimizing a bias/variance tradeoff.

We show these off-policy particle filters are properly weighted, recover desirable gradient estimates previously encountered in the literature, and possess other desirable theoretical properties. To cap off the theoretical results, we also derive a linear convergence rate for gradient methods on the particle filter in the presence of strong convexity, providing support for numerical optimization with AD for the particle filter.

In practice, we propose a hybrid algorithm that we call **Iterated Filtering with Automatic Differentiation (IFAD)** that warm-starts gradient ascent (or a similar iterative first or second-order gradient method) with a coarse solution obtained from a few rounds of iterated filtering. While this requires a differentiable simulator compatible with the reparametrization trick (as previous literature such as **?** also require), future work will deal with workarounds inspired by our framework that use likelihood ratios of transition densities in place of this. Promising numerical results indicate that IFAD beats IF2 (and by the transitive property, IF1, the Liu-West filter, and particle MCMC) on a challenging scientific benchmark problem.

While the Monte-Carlo Adjusted Profile (MCAP) from **?** provides a profile likelihood confidence interval that takes into account both Monte Carlo profile error and statistical uncertainty in the likelihood, it is computationally inefficient. [KT:TODO] We provide a method for incorporating the MCAP methodology in our IFAD algorithm that provides a more computationally efficient profile likelihood confidence interval.

Finally, we make the first steps towards developing a Python counterpart to the popular `pomp` R package by **??**. The software we provide includes welcome features for scientific computing such as GPU acceleration, parallel computing for single-program-multiple-data programs, just-in-time compilation, and automatic differentiation enabled by the `jax` Python package from **?**.

1.3 Significance Statement

Many scientific models involve highly nonlinear stochastic dynamical systems which can be observed only via noisy and incomplete measurements. Under the Markov assumption on system dynamics, previous work has provided methods of performing inference for these models. In particular, prior to this work, iterated filtering algorithms were the only class of algorithms for maximum likelihood estimation that did not require access to the system's transition probabilities, instead needing only a simulator of the system dynamics. We leverage recent advances in automatic differentiation to propose a hybrid algorithm that requires only a differentiable simulator for maximum likelihood estimation. Our new method outperforms previous approaches on a challenging problem in epidemiology. This is implemented in a precursor to a Python counterpart to the popular `pomp` package by **??** that includes welcome features for scientific computing such as GPU acceleration, parallel computing for single-program-multiple-data programs, just-in-time compilation, and automatic differentiation.

[KT:TODO: (1) Fix beta smoothness to d(n) at no more than O(n), (2) add story to say that no one spotted that it could have been plug-and-play because of e.g. reasons why they would stumble]

2 Problem Setup

Consider an unobserved Markov process $\{X_t, t \geq t_0\}$, and observations $Y_1, ..., Y_N$ at timesteps $t_1, ..., t_N$. The process is parameterized by an unknown parameter $\theta \in \Theta$, where $X \in \mathcal{X}, Y \in \mathcal{Y}$, and finally $\mathcal{X}, \mathcal{Y}, \Theta \subseteq \mathbb{R}^n$. By a similar decomposition to that in **?**, we find that the joint density of $X_{0:N}, Y_{1:N}$ can be factored as

$$f_{X_{0:N},Y_{1:N}}\left(x_{0:N}, y_{1:N}; \theta\right) = f_{X_0}\left(x_0; \theta\right) \prod_{n=1}^{N} f_{X_n|X_{n-1}}\left(x_n \mid x_{n-1}; \theta\right) f_{Y_n|X_n}\left(y_n \mid x_n; \theta\right).$$

We call $f_{X_n|X_{n-1}}\left(x_n \mid x_{n-1}; \theta\right)$ the process model, writing $\operatorname{process}(x_n, \theta)$ for the simulator corresponding to the above process model, $f_{Y_n|X_n}\left(y_n \mid x_n, \theta\right)$ the measurement model, and will write y_n^* for the actual values of the observations that were observed.

The above are all defined on a single probability space $(\Omega, \Sigma, \mathbb{P})$, where $\omega \in \Omega$ denotes an element of the sample space that can be thought of as a random seed. For notational simplicity, we omit the dependence of the likelihood and other suitable calculations on X_0.

3 A Smooth Extension of the Particle Filter

Intuition: Our theoretical framework for automatic differentiation for the particle filter hinges on the following observation. Instead of seeking to differentiate $\ell(\theta)$ directly, we instead focus on differentiating through a suitable fixed-seed reweighting of a bootstrap filter. The intuition is as follows. Consider a particle filter likelihood estimate where the system (i.e. **both the process and measurement models**) evolves under a parameter $\phi \in \Theta$ and the seed, or sample space element, is fixed at $\omega \in \Omega$. Conditional on ω and ϕ, one can reweight the likelihood estimate to instead evaluate the likelihood at another (sufficiently nearby) $\theta \in \Theta$.

Differentiating this reweighted likelihood estimate with respect to θ then amounts to differentiating through the reweighting scheme, which is differentiable if the densities used for the reweighting are differentiable. We call this simulation-and-reweighting scheme an **off-policy** likelihood estimate, as it is similar in spirit to the off-policy methods found in the reinforcement learning literature.

The above intuition is formalized below.

3.1 Doubly Off-Policy Particle Filters

Definition 1 (Seed-Fixing)**.** *Consider a probability space* (Ω, Σ, μ) *governing the particles and observations. Noting that the particle filter is unbiased for the likelihood when computing the expectation of the normalizing factor under* *the posterior* $f_{X_{0:N}|Y_{1:N}}\left(x_{0:N} \mid y_{1:N}; \theta\right)$*, we can write the likelihood as*

$$\begin{aligned}
\mathcal{L}(\theta) &= \int_\Omega \mathcal{L}(\theta, \omega, J) d\mu(\omega) \\
&= \int_\Omega \left(\prod_{n=1}^{N} \frac{1}{J} \sum_{j=1}^{J} f_{Y_n|X_n}\left(y_n^* \mid x_{n,j}^P(\omega), \theta\right) \right) f_{X_{0:N},Y_{1:N}}\left(x_{0:N}^P(\omega) \mid y_{1:N}^*, \theta\right) d\mu(\omega) \\
&= \int_\Omega \left(\prod_{n=1}^{N} \frac{1}{J} \sum_{j=1}^{J} f_{Y_n|X_n}\left(y_n^* \mid x_{n,j}^P(\omega), \theta\right) f_{X_{0:N},Y_{1:N}}\left(x_{n,j}^P(\omega) \mid y_n^*, \theta\right) \right) d\mu(\omega)
\end{aligned}$$

where $\mathcal{L}(\theta, \omega, J)$ *is the Monte-Carlo estimate of the likelihood given the realization of* J *particles corresponding to* $\omega \in \Omega$*. For simplicity, we resample every timestep. We call* ω *a seed, and the deterministic quantity* $\mathcal{L}(\theta, \omega, J)$ *the seed-fixed estimate.*

Armed with this seed-fixed likelihood estimate, we modify a particle filter likelihood estimate where the system evolves under a parameter $\phi \in \Theta$ and seed $\omega \in \Omega$ to instead evaluate the likelihood at another (sufficiently nearby) $\theta \in \Theta$.

Definition 2 (Doubly Off-Policy Particle Filter Likelihood)**.** *Let* $\mathcal{L}(\theta, \phi, \omega, J)$ *denote a Monte-Carlo estimate of the likelihood evaluated at* $\theta \in \Theta$ *with* J *particles, but where the system evolves according to another* $\phi \in \Theta$ *and conditional on a fixed seed* $\omega \in \Omega$*. This can be computed with a suitable reweighting of the particles, which we provide the derivation for later. We then have*

$$\mathcal{L}(\theta) = \mathbb{E}[\mathcal{L}(\theta, \phi, \Omega, J)] = \int_\Omega \mathcal{L}(\theta, \phi, \omega, J)\, d\mu(\omega), \tag{1}$$

where we define

$$\mathcal{L}(\theta, \phi, \omega, J) = \prod_{n=1}^{N} \frac{1}{J} \sum_{j=1}^{J} \Big(f_{Y_n|X_n}(y_n^* | x_{n,j}^{P,\phi}(\omega), \phi) f_{X_n|Y_n}(x_{n,j}^{P,\phi}(\omega) | y_n^*, \phi) \tag{2}$$

$$\cdot \frac{f_{Y_n|X_n}(y_n^* | x_{n,j}^{P,\phi}(\omega), \theta)}{f_{Y_n|X_n}(y_n^* | x_{n,j}^{P,\phi}(\omega), \phi)} \cdot \frac{f_{X_n|X_{n-1}}(x_{n,j}^{P,\phi}(\omega) | x_{n-1,j}^{F,\phi}(\omega), \theta)}{f_{X_n|X_{n-1}}(x_{n,j}^{P,\phi}(\omega) | x_{n-1,j}^{F,\phi}(\omega), \phi)} \Big). \tag{3}$$

We want $\phi \in \Theta$ to be a point in parameter space around which $\mathcal{L}(\theta, \phi, \omega, J)$ is differentiable with respect to θ and does not have excessive Monte Carlo variance. This leads us to our first assumption.

Assumption 1 (Smooth Neighborhood)**.** *There exists a neighborhood* $\mathcal{N}(\phi)$ *around any* $\phi \in \Theta$ *where for all* $\theta \in \mathcal{N}(\phi)$ *and almost every* $\omega \in \Omega$*, the Monte Carlo estimate of the likelihood at* $\theta \in \mathcal{N}(\phi)$ *with the system evolving according to* ϕ *conditional on* ω*,* $\mathcal{L}(\theta, \phi, \omega, J)$*, is twice differentiable in* θ*.*

Intuition: We can justify this assumption as follows. For suitably nearby θ and ϕ, the resampling indices for $\mathcal{L}(\theta, \phi, \omega, J)$ and $\mathcal{L}(\phi, \phi, \omega, J)$ are the same, eliminating any discontinuities from resampling, and that for small enough $\mathcal{N}(\phi)$ the likelihood ratios are bounded. The likelihood only changes by a factor of the likelihood ratios, which are bounded and smooth in θ if the densities used in the calculation are.

Lastly, we require the following regularity conditions.

Assumption 2 (Continuity of the Likelihood)**.** $\ell(\theta)$ *proper is continuous in a neighborhood* $\{\theta : \ell(\theta) > \lambda_1\}$ *for some* $\lambda_1 < \sup_\varphi \ell(\varphi)$*.*

Assumption 3 (Bounded Measurement Model)**.** *There is an* $\epsilon > 0$ *with* $\epsilon^{-1} > f_{Y_n|X_n}\left(y_n{}^* \mid x_n; \theta\right) > \epsilon$ *for all* $1 \le n \le N, x_n \in \mathcal{X}$ *and* $\theta \in \Theta$*.*

Assumption 4 (Locally Bounded Derivative)**.** *Let* $\mathcal{M}$ *be an open subset of* Ω*. There exists some function* $G(\theta)$ *and a constant* $G = \sup_{\theta \in \mathcal{N}} G(\theta) < \infty$ *such that*

$$\|\nabla \ell(\theta, \phi, \omega, J)\|_2 < G(\theta) \le G < \infty$$

for every $\phi \in \mathcal{M}, \theta$ *in smooth neighborhood* $\mathcal{N}(\phi)$*, and almost every* $\omega \in \Omega$*.*

We claim that this yields a properly weighted particle filter that recovers the derivative estimate of **??** when θ is evaluated at ϕ. [KT:INSERT RESULTS HERE]

Proposition 1 (Correctness of DOP-α (Algorithm **??**))**.** *When θ is evaluated at ϕ,*

3.2 Measurement Off-Policy

However, the above algorithm requires access to likelihood ratios, and is therefore not compatible with simulation-based inference.

A Doubly Off-Policy Particle Filters

Evaluating the off-policy likelihood is possible with a weighted particle filter by using resampling weight $p(y_n^*|x_n;\phi)$ and making two corrections in the particle weight, through multiplying $w_{t,j}$ by the two likelihood ratios below,

$$s = p(y_n^*|x_n;\theta)/p(y_n^*|x_n;\phi) \tag{4}$$

and

$$r = p(x_n|x_{n-1};\theta)/p(x_n|x_{n-1};\phi). \tag{5}$$

The following result showcases the correctness of corrections 4 and 5.

Proposition 2. *For θ, ϕ that satisfy Assumption 1, the following correction in the particle filter weights,*

$$w_{t,j} := \tilde{w}_{t,j} \cdot p(y^*|x_{t,j}^P, \phi) \cdot r \cdot s \tag{6}$$

successfully recovers $\mathcal{L}(\theta)$.

Proof. Recall $\mathcal{L}(\theta) = \mathbb{E}[\mathcal{L}(\theta, \phi, \Omega, J)] = \int_\Omega \mathcal{L}(\theta, \phi, \omega, J)\, d\mu(\omega)$ and $\mathcal{L}(\theta) = \int_\Omega \mathcal{L}(\theta, \omega, J) d\mu(\omega)$. We want to evaluate

$$\begin{aligned}
\mathcal{L}(\theta, \omega, J) = \mathbb{E}\left[\prod_{t=1}^T \mathcal{L}_t(\theta)\middle|\omega\right] &= \mathbb{E}\left[\prod_{t=1}^T p(y_t^*|y_{1:t-1}^*, \theta)\middle|\omega\right] \\
&= \mathbb{E}\left[\prod_{t=1}^T \frac{1}{J}\sum_{j=1}^J p(y_t^*|x_{t,j}^P, \theta)p(x_{t,j}^P|y_{1:t-1}^*, \theta)\middle|\omega\right] \\
&= \mathbb{E}\left[\prod_{t=1}^T \frac{1}{J}\sum_{j=1}^J p(y_t^*|x_{t,j}^P, \theta)p(x_{t,j}^P|x_{t-1}^F, \theta)\middle|\omega\right]
\end{aligned}$$

where $p(x_t^P|y_{1:t-1}^*, \theta)$ is the prediction distribution of particles, and the filtering distribution of particles is proportional to $p(y_t^*|x_{t,j}^P, \theta)$, the likelihood of particle j at time t.

Consider first the case with no resampling. Let $\mathcal{L}^j(\theta)$, $\mathcal{L}^j(\phi)$ be the likelihood of trajectory j under θ and ϕ respectively. As the prediction and filtering particles are then the same, we use $x_{t,j}$ to refer to both. As such, conditional on ω,

$$\begin{aligned}
\mathcal{L}^j(\theta) &= \prod_{t=1}^T p(y_t^*|x_{t,j}, \theta)p(x_{t,j}|x_{t-1,j}, \theta) \\
&= \prod_{t=1}^T p(y_t^*|x_{t,j}, \phi)\frac{p(y_t^*|x_{t,j}, \theta)}{p(y_t^*|x_{t,j}, \phi)}p(x_{t,j}|x_{t-1,j}, \theta)\frac{p(x_{t,j}|x_{t-1,j}, \theta)}{p(x_{t,j}|x_{t-1,j}, \phi)} \\
&= \prod_{t=1}^T p(y_t^*|x_{t,j}, \phi) \cdot s \cdot p(x_{t,j}|x_{t-1,j}, \phi) \cdot r
\end{aligned}$$

and therefore, if trajectories were proposed with equal probability with particles conditional on ω,

$$\mathcal{L}(\theta, \omega, J) = \frac{1}{J}\sum_{j=1}^{J}\prod_{t=1}^{T} p(y_t^*|x_{t,j}, \phi) \cdot s \cdot p(x_{t,j}|x_{t-1,j}, \phi) \cdot r$$

The correction is therefore sufficient for recovering $\mathcal{L}(\theta)$, with trajectories collected with the process model at ϕ and measurements evaluated with the measurement model at ϕ.

Consider now the case with resampling. Without loss of generality we can resample every timestep. Here, $x_{t,j}^F(\phi) \sim p(y^*|x_{t,j}^P, \phi)$ and $x_{t,j}^P(\phi) \sim p(x_t|x_{t-1}^F, \phi)$, and similarly for θ. We use this notation to emphasize that **both** the measurement model and the previously drawn particles depend on ϕ or θ.

$$\begin{aligned}
\mathcal{L}_t(\theta) &= \mathbb{E}_{x_{t,j}^P \sim p(x_t|x_{t-1}^F, \theta)}\left[\frac{1}{J}\sum_{j=1}^{J} p(y_t^*|x_{t,j}^P, \theta)\right] \\
&= \mathbb{E}_{x_{t,j}^P \sim p(x_t|x_{t-1}^F, \theta)}\left[\frac{1}{J}\sum_{j=1}^{J} p(y_t^*|x_{t,j}^P, \phi)\frac{p(y_t^*|x_{t,j}^P, \theta)}{p(y_t^*|x_{t,j}^P, \phi)}\right] \\
&= \mathbb{E}_{x_{t,j}^P \sim p(x_t|x_{t-1}^F, \phi)}\left[\frac{1}{J}\sum_{j=1}^{J} p(y_t^*|x_{t,j}^P, \phi)\frac{p(y_t^*|x_{t,j}^P, \theta)}{p(y_t^*|x_{t,j}^P, \phi)}\frac{p(x_{t,j}|x_{t-1,j}, \theta)}{p(x_{t,j}|x_{t-1,j}, \phi)}\right] \\
&= \mathbb{E}_{x_{t,j}^P \sim p(x_t|x_{t-1}^F, \phi)}\left[\frac{1}{J}\sum_{j=1}^{J} p(y_t^*|x_{t,j}^P, \phi) \cdot s \cdot r\right] \\
&= \mathbb{E}_{\omega}\mathbb{E}_{x_{t,j}^P \sim \text{process}(x_{t-1}^F, \omega, \phi)}\left[\frac{1}{J}\sum_{j=1}^{J} p(y_t^*|x_{t,j}^P, \phi) \cdot s \cdot r \middle| \omega\right]
\end{aligned}$$

and again the correction is sufficient.

Therefore,

$$\begin{aligned}
\mathcal{L}(\theta) &= \int_{\Omega}\left(\prod_{n=1}^{N}\frac{1}{J}\sum_{j=1}^{J} f(y_n^*|x_{n,j}^P(\omega), \theta)p(x_{n,j}^P(\omega)|x_{n-1,j}^P(\omega), \theta)\right) d\mu(\omega) && (7) \\
&= \int_{\Omega}\left(\prod_{n=1}^{N}\frac{1}{J}\sum_{j=1}^{J} f(y_n^*|x_{n,j}^{P,\theta}(\omega), \theta)p(x_{n,j}^{P,\theta}(\omega)|x_{n,j}^P(\omega), \theta)\right) d\mu(\omega) && (8) \\
&= \int_{\Omega}\left(\prod_{n=1}^{N}\frac{1}{J}\sum_{j=1}^{J} f(y_n^*|x_{n,j}^{P,\theta}(\omega), \phi)p(x_{n,j}^{P,\theta}(\omega)|y_n^*, \phi) \cdot s\right) d\mu(\omega) && (9) \\
&= \int_{\Omega}\left(\prod_{n=1}^{N}\frac{1}{J}\sum_{j=1}^{J} f(y_n^*|x_{n,j}^{P,\phi}(\omega), \phi)p(x_{n,j}^{P,\phi}(\omega)|y_n^*, \phi) \cdot r \cdot s\right) d\mu(\omega) && (10) \\
&= \int_{\Omega}\left(\prod_{n=1}^{N}\frac{1}{J}\sum_{j=1}^{J} f(y_n^*|x_{n,j}^{P,\phi}(\omega), \phi)p(x_{n,j}^{P,\phi}(\omega)|y_n^*, \phi) \cdot r \cdot s\right) d\mu(\omega) && (11) \\
&= \int_{\Omega}\left(\prod_{n=1}^{N}\frac{1}{J}\sum_{j=1}^{J} f(y_n^*|x_{n,j}^{P,\phi}(\omega), \phi) \cdot r \cdot s\right) p(x_{1:N}|y_{1:N}^*, \phi)\, d\mu(\omega) && (12) \\
&=: \int_{\Omega} \mathcal{L}(\theta, \phi, \omega, J)\, d\mu(\omega). && (13)
\end{aligned}$$

□

A.1 Off-Policy Weighted Differentiable Particle Filter

We are finally ready to introduce the off-policy weighted differentiable particle filter. In the event where a seed is fixed, all expressions in Algorithm **??** can be considered to be conditional on $\omega \in \Omega$.

Algorithm 1 Doubly Off-Policy-α

```
1: Input: Number of particles $J$, timesteps $N$, measurement model $f_{Y_n|X_n}(y_n^*|x_n,\theta)$, simulator process$(x_{n+1}|x_n,\theta)$, evaluation parameter $\theta$, behavior parameter $\phi$, seed $\omega$.
2: Initialize filter particles $X_{0,j}^{F,\phi} \sim f_{X_0}(\,\cdot\,;\phi)$, relative weights $w_{0,j}^{F,\theta} = 1$ for $j$ in $1\!:\!J$. Fix $\omega$.
3: for $n = 1, ..., N$ do
4:     Prediction weights with discounting: $w_{n,j}^{P,\theta} = \big(w_{n-1,j}^{F,\theta}\big)^{\alpha}$ for $j$ in $1\!:\!J$
5:     Simulate for prediction: $X_{n,j}^{P,\phi} \sim f_{X_n|X_{n-1}}\big(\,\cdot\,|X_{n-1,j}^{F};\phi\big)$ for $j$ in $1\!:\!J$
6:     Adjust weights: $w_{n,j}^{P,\theta} = w_{n,j}^{P,\theta} \times \dfrac{f_{X_n|X_{n-1}}\big(X_{n,j}^{P,\phi}|X_{n-1,j}^{F};\theta\big)}{f_{X_n|X_{n-1}}\big(X_{n,j}^{P,\phi}|X_{n-1,j}^{F};\phi\big)}$ for $j$ in $1 : J$
7:     Evaluate measurement density: $g_{n,j}^{\theta} = f_{Y_n|X_n}(y_n^*|X_{n,j}^{P,\phi};\theta)$ for $j$ in $1\!:\!J$
8:     Before-resampling conditional likelihood: $L_n^{B,\theta,\alpha} = \dfrac{\sum_{j=1}^{J} g_{n,j}^{\theta} w_{n,j}^{P,\theta}}{\sum_{j=1}^{J} w_{n,j}^{P,\theta}}$
9:     Conditional likelihood under $\phi$: $L_n^{\phi} = \frac{1}{J}\sum_{m=1}^{J} g_{n,m}^{\phi}$
10:    Normalize weights: $\tilde{g}_{n,j}^{\phi} = \dfrac{g_{n,j}^{\phi}}{JL_n^{\phi}}$ for $j$ in $1\!:\!J$
11:    Apply systematic resampling to select indices $k_{1:J}$ with $\mathbb{P}\big(k_j = m\big) = \tilde{g}_{n,m}^{\phi}$
12:    Resample particles: $X_{n,j}^{F,\phi} = X_{n,k_j}^{P,\phi}$
13:    Filter weights corrected for resampling: $w_{n,j}^{FC,\theta} = w_{n,j}^{P,\theta} \times \dfrac{g_{n,j}^{\theta}}{g_{n,j}^{\phi}}$ for $j$ in $1\!:\!J$
14:    Resample filter weights: $w_{n,j}^{F,\theta} = w_{n,k_j}^{FC,\theta}$ for $j$ in $1\!:\!J$
15:    After-resampling conditional likelihood: $L_n^{A,\theta,\alpha} = L_n^{\phi}\,\dfrac{\sum_{j=1}^{J} w_{n,j}^{F,\theta}}{\sum_{j=1}^{J} w_{n,j}^{P,\theta}}$
16: end for
17: return likelihood estimate $\hat{\mathcal{L}}(\theta) = \mathcal{L}(\theta,\phi,\omega,J) := \prod_{n=1}^{N} L_n^{A,\theta,\alpha}$, filtering distribution $\{(x_{N,j}^{P}, w_{N,j}^{F,\theta})\}$.
```

Here, we consider a doubly off policy particle filter, meaning that both `rprocess` and `rmeasure` are computed at ϕ and particles are reweighted to correspond to θ.