Differentiable SMC Samplers for Fast Bayesian Deep Learning

Vincent Beraud, Supervised by Simon Maskell, Vassil Alexandrov (Hartree Centre), James (GCHQ)

EPSRC Centre for Doctoral Training in Distributed Algorithms, University of Liverpool, Liverpool, UK

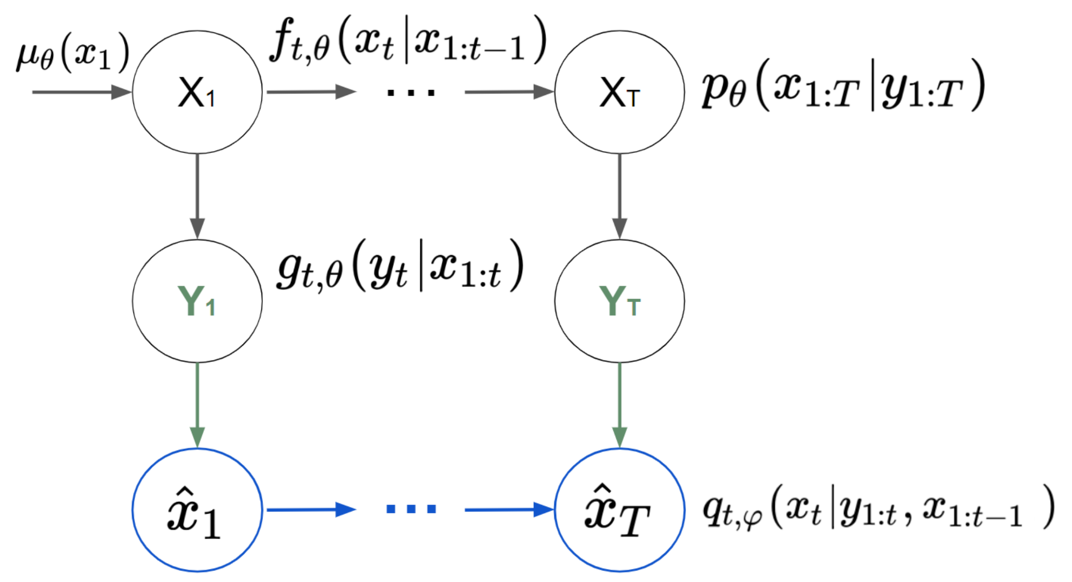
Motivation and Summary

- Provide a **Bayesian solution** to neural networks to obtain uncertainty.
- Build a Bayesian Neural network that can deal with **stationary distributions** with a **scalable convergence**.
- Make a clear differentiation of SMC samplers.
- SMC samplers combine benefits from MCMC methods and Particle Filters.

Background

We want to infer a target distribution. We use a proposal distribution to generate samples and use **Importance sampling** to weight them proportionally to our target. Using these weighted samples we can infer our target.

- **MCMC methods** have offered state-of-the-art sampling methods to infer stationary distributions.
- **Particle filters** are sequential Bayesian samplers that are used in wide range of fields involving state-space models, they offer unbiased Bayesian estimates and are parallelisable.



Algorithm 1 Bootstrap Particle Filter

> Sample initial N particles $x_1^{(i)} \sim q_{1,\phi}(y_1)$
> Compute the weights $\widetilde{\mathbf{w}}_1^{(i)} = \frac{p_{1,\theta}(x_1^{(i)}, y_1)}{q_{1,\phi}(x_1^{(i)}|y_1)} = \frac{\mu_\theta(x_t^{(i)}) g_{1,}(y_1|x_t^{(i)})}{q_{1,\phi}(x_1^{(i)}|y_1)}$
> Set $t = 2$
> **for** $k \leq$ iter **do**
>   Normalise weights, $\mathbf{w}_t^{(i)} = \frac{\widetilde{\mathbf{w}}_k^{(i)}}{\sum_j (\widetilde{\mathbf{w}}_t^{(j)})}$
>   Calculate $N_{eff} = \frac{1}{\sum_i (\mathbf{w}_t^{(i)})^2}$
>   **if** $N_{eff} < \frac{N}{2}$ **then**
>   Resample the particles $\widetilde{x}_t^{(i)} \sim \sum_i^N \mathbf{w}_t^{(i)} \delta_{x_t^{(i)}}$
>   Set $\mathbf{w}_t^{(i)} = \frac{1}{N}$
>   **end if**
>   t = t+1
>   Propagate the particles $x_t^{(i)} \sim q_\phi(x_t|x_{t-1}^{(i)})$
>   Calculate $\widetilde{\mathbf{w}}_t^{(i)} = \mathbf{w}_{t-1}^{(i)} \frac{p_{t,\theta}\left(x_k^{(i)}, y_t\right)}{q_{t,\phi}\left(x_t^{(i)}|x_{k-1}^{(i)}, y_t\right)} = \mathbf{w}_{t-1}^{(i)} \frac{f_{t,\theta}(x_t^{(i)}|x_{1:t1}) g_{t,\theta}(y_t|x_{1:t})}{qt,\phi\left(x_k^{(i)}|x_{k-1}^{(i)}, y_t\right)}$
> **end for**

We just give our filer/sampler the measurements generated by the target and likelihood dist.

Here we aim to infer the **posterior distribution:**

$p_\theta(x_{1:T}|y_{1:T})$

θ, ϕ are fixed , the weights give us the likelihood based on the measurements.

Differentiable Particle Filter:

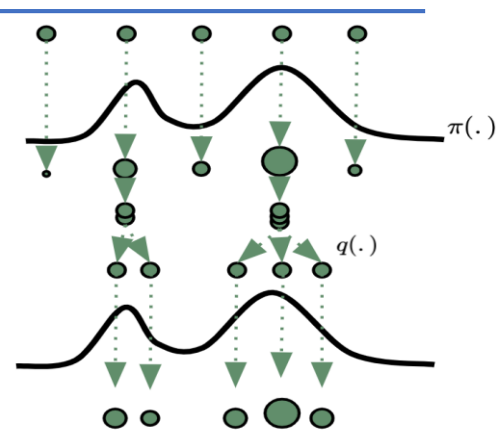
We can differentiate the PF and **learn the parameters** of all the **distributions** by maximising the normalising constant, therefore minimising: $-\frac{1}{T}\sum_{t=1}^{T} \nabla_{\theta,\phi} log(\sum_{i=1}^{N} \widetilde{w}_t^i)$. However:

- **Differentiable Resampling**: Resampling avoids weight degeneracy and low variance estimates. But it can't be differentiated as it's multinomial. Therefore several solutions have been proposed, [1] [2] [3] [4] , it seems that it exists a trade-off between biased-likelihood and biased-gradient.
- **Confusion in the literature:** What loss do we use ? Belief ? Semi supervised ? Optimising the normalising constant yields to a minimisation of the ELBO, therefore, optimisation of all the distributions/functions: the transition, likelihood, proposals and prior.

SMC Samplers

An SMC sampler targets a distribution $\pi(x)$ over T iterations using N samples. Importance sampling assigns each sample a weight, like in MCMC, at iteration t which is used to estimate $\pi(x)$.

$$\widetilde{\mathbf{w}}_t^{(i)} = \mathbf{w}_{t-1}^{(i)} \frac{p_{t,\theta}\left(x_t^{(i)}, y_t\right) \mathcal{L}_{t,\psi}\left(x_{t-1}^{(i)}|x_t^{(i)}, y_t\right)}{p_{t,\theta}\left(x_{t-1}^{(i)}, y_t\right) q_{t,\phi}\left(x_t^{(i)}|x_{t-1}^{(i)}, y_t\right)}$$



- **Backward Kernel (L-kernel):** The L-Kernel, $\mathcal{L}$, is commonly underexploited. The literature uses the proposal kernel as the backward kernel but it has been proven that it doesn't necessarily has to be the case [5]. A good choice of a backward kernel can improve the exploration of the parameter space and then **speed up convergence**.
- **Parallelisable:** SMC samplers can benefit from all the parallelisation techniques that are applied in the PF literature, even the resampling step [6].
- **Other Benefits**: **No burn-in steps**, handle multimodality, recycling.

Prior Proposals

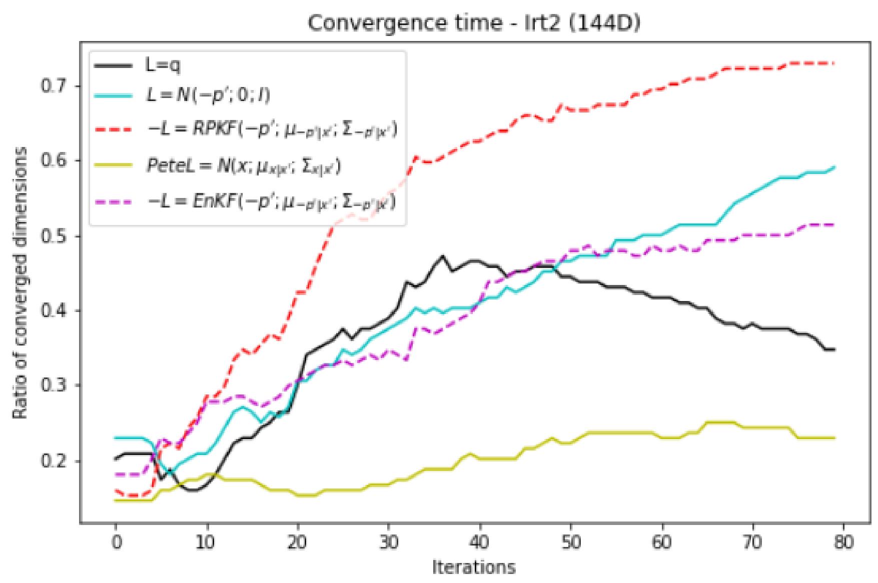
- **Proposal Kernel:** Even though a significant portion of the literature focuses on the impact of the resampling gradient bias. It has been shown in a submitted paper that the **choice of the proposal as a bigger impact** on the quality of the parameter estimation.

$$\widetilde{\mathbf{w}}_t^{(i)} = \mathbf{w}_{t-1}^{(i)} \frac{p_{t,\theta}\left(x_t^{(i)}, y_t\right) \mathcal{L}_{t,\psi}\left(-p_t^{(i)}|x_t^{(i)}, y_t\right)}{p_{t,\theta}\left(x_{t-1}^{(i)}, y_t\right) q_{t,\phi}\left(p_{t-1}^{(i)}|x_{t-1}^{(i)}, y_t\right)}$$

Weight update using **NUTS** [7] as a proposal. Where p is the momentum in the leapfrog algorithm.

- **L-Kernel:** It is possible to directly compute an optimal L-Kernel [5]. However its computational complexity remains $O(N^2)$ which might be a non negligible bottleneck.
  - **Near optimal-l Kernel using Kalman Filtering** [5]. Therefore we can choose to use a Ensemble Kalman Filter (to tackle non-linearity and high dimensions) as a decent backward kernel instead of learning it through optimisation.

We proposed a new high dimensional Kalman filter (RPKF). We show here a comparison of convergence depending on the choice of L-Kernel on a 144 dimensional model (irt2) [8]. We compare against L=q (forward kernel), a suboptimal L-kernel, RPKF, regular KF (Pete), EnKF.



Future Work & References

- **VAE:** We will try to compete against the current state of the art methods for the inference of latent space.
- **Fully Parallelised SMC Network:** We will try to exploit the inherent parallelisation of the sampler to speed up inference.

[1] Corenflos, A., Thornton, J., Deligiannidis, G., Doucet, A. Differentiable Particle Filtering via Entropy-Regularized Optimal Transport. Proceedings of the 38th International Conference on Machine Learning, PMLR 139:2100-2111, 2021.
[2] Le, T., Igl, M., Rainforth, T., Jin, T., & Wood, F. (2019). Auto-encoding sequential Monte Carlo. International Conference on Learning Representations (ICLR).
[3] Jonschkowski, R., Rastogi, D., Brock, O.: Differentiable Particle Filters: End-to-End Learning with Algorithmic Priors. Robotics: Science and Systems 2018
[4] Ma, X., Karkus, P., Hsu, D. (2020). Particle Filter Recurrent Neural Networks. Proceedings of the AAAI Conference on Artificial Intelligence. 34. 5101-5108. 10.1609/aaai.v34i04.5952.
[5] Wu, J., Wen, L., Green, P.L. et al. Ensemble Kalman filter based sequential Monte Carlo sampler for sequential Bayesian inference. Stat Comput 32, 20 (2022). doi.org/10.1007/s11222-021-10075-x
[6] Varsi, A., Maskell, S., & Spirakis, P. G. (n.d.). An O(log2N) Fully-Balanced Resampling Algorithm for Particle Filters on Distributed Memory Architectures. Algorithms, 14(12), 342. doi:10.3390/a14120342
[7] Devlin, L., Horridge, P., Green L, P., Maskell, S., (2021), The No-U-Turn Sampler as a Proposal Distribution in a Sequential Monte Carlo Sampler with a Near-Optimal L-Kernel. arXiv:2108.02498
[8] Stan User's Guide https://mc-stan.org/docs/2_29/stan-users-guide/item-response-models.html

Engineering and Physical Sciences Research Council
UNIVERSITY OF LIVERPOOL
GCHQ