Towards Heuristic Weights for Sequential Monte Carlo by Future Likelihood Estimates

Adam Jozefiak

April 28, 2021

Sequential Monte Carlo

- For our setting, $X_{1:N}$ are the latent random variables and $Y_{1:N}$ are the observed random variables.
- We recall the Sequential Monte Carlo (SMC) Algorithm.
- In SMC we target the sequence of distributions $\{p(X_{1:n}|Y_{1:n})\}_{n=1}^{N}$, obtaining unbiased estimates of the distributions $p(X_{1:n}|Y_{1:n})$ by a collection of particles $\{X_{(1:n),j}\}_{j=1}^{M}$.
- At each time step, when going from $\hat{p}(X_{1:n-1}|Y_{1:n-1})$ to $\hat{p}(X_{1:n}|Y_{1:n})$ we resample particles according to likelihood weights

$$\frac{p(Y_n|X_{1:n}, Y_{1:n-1})p(X_n|X_{1:n-1}, Y_{1:n-1})}{q_\phi(X_n|X_{1:n-1}, Y_{1:N})}$$

- Where $q_\phi(X_n|X_{1:n-1}, Y_{1:N})$ is a proposal distribution.

Motivation

- We hypothesize that there may be future dependencies where the importance weights

$$\frac{p(Y_n|X_{1:n}, Y_{1:n-1})p(X_n|X_{1:n-1}, Y_{1:n-1})}{q_\phi(X_n|X_{1:n-1}, Y_{1:N})}$$

  may undervalue "good" particles in the situation where $p(Y_n|X_{1:n}, Y_{1:n-1})p(X_n|X_{1:n-1}, Y_{1:n-1})$ is extremely small but $q_\phi(X_n|X_{1:n-1}, Y_{1:N})$ is large.
- This motivates us to ask the question, could we have access to, target, or approximate the sequence of distributions $\{p(X_{1:n}|Y_{1:N})\}_{n=1}^N$? Thereby obtaining approximations to the likelihood weights

$$\frac{p(X_n|X_{1:n-1}, Y_{1:N})}{q_\phi(X_n|X_{1:n-1}, Y_{1:N})}$$

  which no longer collapse for particles with high future likelihoods but low likelihood under the prior.

Overview

- We begin by first unpacking the following ratio $\frac{p(X_n|X_{1:n-1}, Y_{1:N})}{q_\phi(X_n|X_{1:n-1}, Y_{1:N})}$
  1. Obtaining terms that can be computed exactly given $X_{1:n}$ and $Y_{1:n}$.
  2. Obtaining the terms $p(Y_{n+1:N}|X_{1:n}, Y_{1:n})$ and $p(Y_{n:N}|X_{1:n-1}, Y_{1:n-1})$ which cannot be computed given $X_{1:n}$ and $Y_{1:n}$ and must therefore be approximated.
- We then frame the sequence of future likelihoods $\{p(Y_{n+1:N}|X_{1:n}, Y_{1:n})\}_{n=1}^{N-1} \cup p(Y_{1:N})$ as value functions $\{V_n\}_{n=0}^{N-1}$ that satisfy a recurrence relation.
- We then give an objective for learning approximate value functions and show that the value functions can be trained by stochastic gradient descent and nested Monte Carlo.
- We then investigate whether SMC can break down with outliers present in $Y_{1:N}$, and if so, does the addition of the value functions to the "standard" SMC likelihood weights, as a heuristic, improve the performance.

Derivation of Long-Term Weights

- For each $n \in \{0, 1, 2, \ldots, N-1\}$ we let $W_n = \frac{p(X_n|X_{1:n-1}, Y_{1:N})}{q_\phi(X_n|X_{1:n-1}, Y_{1:N})}$.
- We then obtain that:
  - $$W_1 = \frac{p(Y_{2:N}|X_1, Y_1)p(Y_1|X_1)p(X_1)}{p(Y_{1:N})q_\phi(X_1|Y_{1:N})}$$
  - $\forall n \in \{2, 3, \ldots, N-2\}$,

$$W_n = \frac{p(Y_{n+1:N}|X_{1:n}, Y_{1:n})p(Y_{1:n}|X_{1:n})p(X_n|X_{1:n-1})}{p(Y_{n:N}|X_{1:n-1}, Y_{1:n-1})p(Y_{1:n-1}|X_{1:n-1})q_\phi(X_{1:n}|X_{1:n-1}, Y_{1:N})}$$

  - $$W_N = \frac{p(Y_{1:N}|X_{1:N})p(X_N|X_{1:N-1})}{p(Y_N|X_{1:N-1}, Y_{1:N-1})p(Y_{1:N-1}|X_{1:N-1})q_\phi(X_N|X_{1:N-1}, Y_{1:N})}$$

Derivation of Long Term Weights

- We remark that the likelihood weights are similar to those of the "standard" SMC, except, for the terms of the form $p(Y_{n+1:N}|X_{1:n}, Y_{1:n})$. Below we give a derivation of the general time-step, omitting the base and terminal cases as they follow by similar arguments.

$$
\begin{aligned}
W_n &= \frac{p(X_n|X_{1:n-1}Y_{1:N})}{q_\phi(X_{1:n}|X_{1:n-1}, Y_{1:N})} \\
&= \frac{p(X_{1:n}, Y_{1:N})}{p(X_{1:n-1}, Y_{1:N})q_\phi(X_{1:n}|X_{1:n-1}, Y_{1:N})} \\
&= \frac{p(Y_{1:N}|X_{1:n})p(X_{1:n})}{p(Y_{1:N}|X_{1:n-1})p(X_{1:n-1})q_\phi(X_{1:n}|X_{1:n-1}, Y_{1:N})} \\
&= \frac{p(Y_{1:N}|X_{1:n})p(X_n|X_{1:n-1})}{p(Y_{1:N}|X_{1:n-1})q_\phi(X_{1:n}|X_{1:n-1}, Y_{1:N})} \\
&= \frac{p(Y_{n+1:N}|X_{1:n}, Y_{1:n})p(Y_{1:n}|X_{1:n})p(X_n|X_{1:n-1})}{p(Y_{n:N}|X_{1:n-1}, Y_{1:n-1})p(Y_{1:n-1}|X_{1:n-1})q_\phi(X_{1:n}|X_{1:n-1}, Y_{1:N})}
\end{aligned}
$$

Future Likelihoods as Value Functions

- We define $V_0 := p(Y_{1:N})$
- We define
  $\forall n \in \{1, 2, \ldots, N-1\}, V_n(X_{1:n}) := p(Y_{n+1:N} | X_{1:n}, Y_{1:n})$
- Note, $V_n(X_{1:n})$ is really a function of $X_{1:n}$ and $Y_{1:N}$, i.e $V_n(X_{1:n}, Y_{1:N})$, but we drop $Y_{1:N}$ when the observed random variables are apparent and fixed for a given inference task.
- Inspired by the reinforcement learning context, we can think of $\{V_n\}_{n=0}^{N-1}$ as value functions that capture the future likelihoods given the latent and observed random variables seen so far (Sutton, 1998).

Future Likelihoods as Value Functions

- In particular $\{V_n\}_{n=0}^{N-1}$ satisfy the following recurrence relation, which can be thought of as a Bellman equation.
- $\forall n \in \{2, 3, \ldots, N-1\}, \forall X_{1:n-1}, V_{n-1}(X_{1:n-1}) = \mathbb{E}_{X_n \sim p(X_n|X_{1:n-1}, Y_{1:n-1})}[p(Y_n|X_{1:n}, Y_{1:n-1})V_n(X_{1:n})]$
- Base case: $V_0 = \mathbb{E}_{X_1 \sim p(X_1)}[p(Y_1|X_1)V_1(X_1)]$
- Terminal case: $V_{N-1}(X_{1:N-1}) = \mathbb{E}_{X_N \sim p(X_N|X_{1:N-1}, Y_{1:N-1})}[p(Y_N|X_{1:N}, Y_{1:N-1})]$
- If $\{V_n\}_{n=1}^{N-1}$ are a family of functions that satisfy the above reccurence, then scaling $\{V_n\}_{n=1}^{N-1}$ by a scalar α will result in $\{\alpha V_n\}_{n=1}^{N-1}$ satisfying all but the terminal case of the recurrence. Therefore, it is paramount to include the terminal case in order to "fix" the scale of V_{N-1} and ultimately the scale of all of the $\{V_n\}_{n=1}^{N-1}$.

Future Likelihoods as Value Functions

- We provide the proof of the general case of the recurrence, omitting the base and terminal cases as they follow similar arguments.

$$
\begin{aligned}
& V_{n-1}(X_{1:n-1}) \\
= \quad & p(Y_{n:N}|X_{1:n-1}, Y_{1:n-1}) \\
= \quad & \int p(Y_{n:N}|X_{1:n}, Y_{1:n-1}) p(X_n|X_{1:n-1}, Y_{1:n-1}) dX_n \\
= \quad & \int p(Y_{n+1:N}|X_{1:n}, Y_{1:n}) p(Y_n|X_{1:n}, Y_{1:n-1}) p(X_n|X_{1:n-1}, Y_{1:n-1}) dX_n \\
= \quad & \mathbb{E}_{X_n \sim p(X_n|X_{1:n-1}, Y_{1:n-1})}[p(Y_n|X_{1:n}, Y_{1:n-1}) V_n(X_{1:n})]
\end{aligned}
$$

Learning the Value Functions

- A family of estimators $\{\hat{V}_n\}_{n=0}^{N-1}$ that satisfy the above recurrence are a sufficient replacements for $\{V_n\}_{n=0}^{N-1}$ for computing the likelihood weights $W_1, W_2, \ldots, W_N$ of the distributions $\{p(X_{1:n}|Y_{1:N})\}_{n=1}^{N}$.
- Therefore, using the value function recurrence we place an objective on learning value function approximators $\{\hat{V}_{\psi,n}\}_{n=0}^{N-1}\}$.

$$
\begin{aligned}
& \mathcal{L}(\psi) \\
:= \quad & \sum_{n=1}^{N-1} \mathbb{E}[(\hat{V}_{\psi,n-1}(X_{1:n-1}) - \mathbb{E}[p(Y_n|X_{1:n}, Y_{1:n-1})\hat{V}_{\psi,n}(X_{1:n})])^2] \\
+ \quad & \mathbb{E}[(\hat{V}_{\psi,N-1}(X_{1:N-1}) - \mathbb{E}[p(Y_N|X_{1:N}, Y_{1:N-1})])^2]
\end{aligned}
$$

- Where the inner expectation is take with respect to $X_n \sim p(X_n|X_{1:n-1}, Y_{1:N})$ and where the outer expectation can be taken with respect to $p(X_{1:n-1})$ the prior, $p(X_{1:n-1}|Y_{1:n})$, or $p(X_{1:n-1}|Y_{1:N})$.

Learning the Value Functions

- We can learn estimators $\{\hat{V}_{\psi,n}\}_{n=0}^{N-1}$ by performing stochastic gradient descent (SGD) with respect to $\mathcal{L}(\psi)$.

$$
\begin{aligned}
& \nabla_\psi \mathcal{L}(\psi) \\
= \; & \sum_{n=1}^{N-1} \mathbb{E}\big[(\hat{V}_{\psi,n-1}(X_{1:n-1}) - \mathbb{E}[p(Y_n|X_{1:n}, Y_{1:n-1})\hat{V}_{\psi,n}(X_{1:n})]) \\
& \qquad (\nabla_\psi \hat{V}_{\psi,n-1}(X_{1:n-1}) - \mathbb{E}[p(Y_n|X_{1:n}, Y_{1:n-1})\nabla_\psi \hat{V}_{\psi,n}(X_{1:n})]\big) \\
+ \; & \mathbb{E}[(\hat{V}_{\psi,N-1}(X_{1:N-1}) - \mathbb{E}[p(Y_N|X_{1:N}, Y_{1:N-1})])\nabla_\psi \hat{V}_{\psi,N-1}(X_{1:N-1}]
\end{aligned}
$$

- During training we can have stages where we take the outer expectation according to a sequence of distributions, moving away from the prior and towards $p(X_{1:n-1}|Y_{1:n-1})$ and $p(X_{1:n-1}|Y_{1:N})$ as the value function approximators become more accurate.
- Note, we must make nested Monte Carlo approximations for computing approximations to $\nabla_\psi \mathcal{L}$ (Rainforth, 2018).

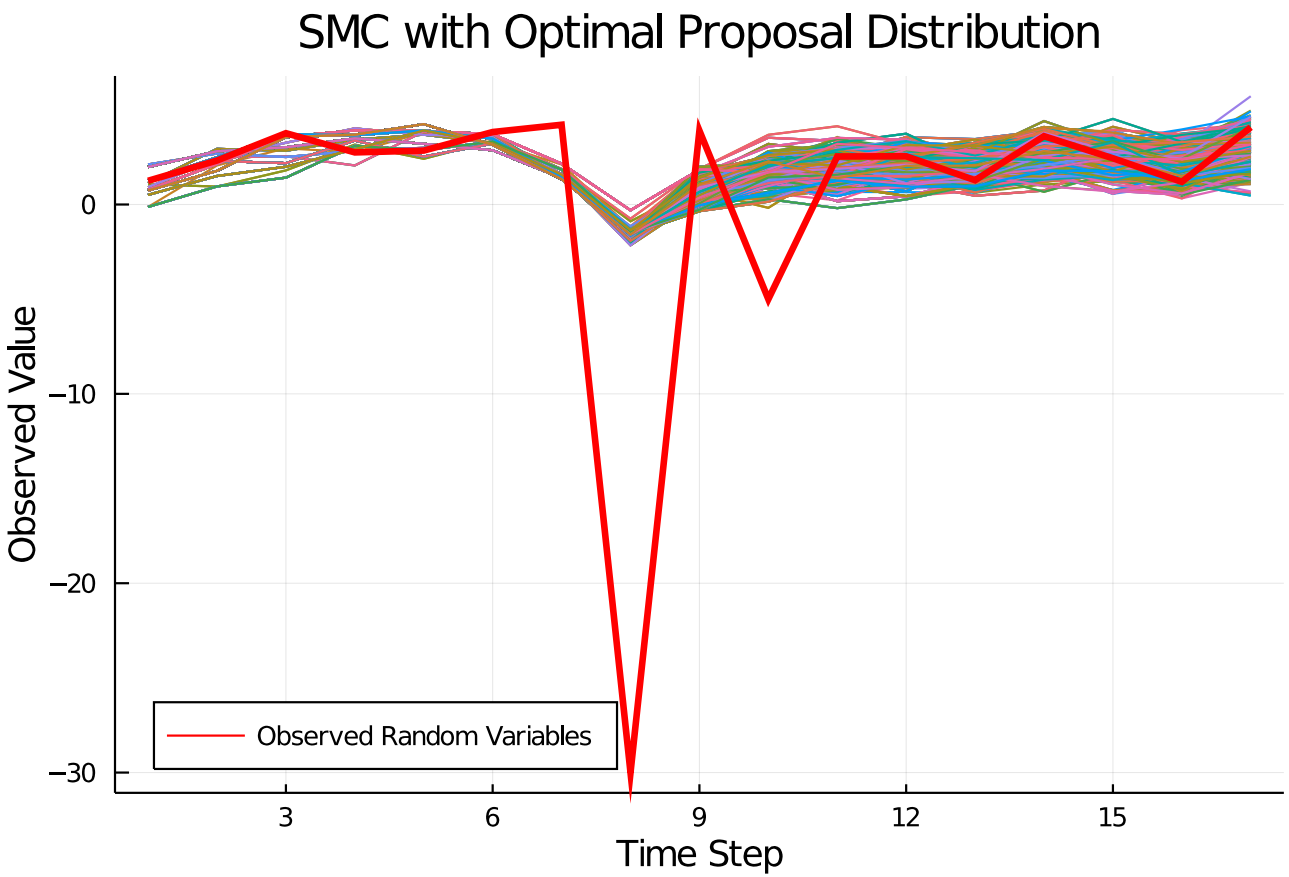
Limitations of the Value Function Approximators

- We do not have guarantees that our trained $\{\hat{V}_{\psi,n}\}_{n=0}^{N-1}$ are unbiased estimators of the value functions. Therefore, we cannot simply plug them into the weights for inferring the long-range distributions $\{p(X_{1:n}|Y_{1:N})\}_{n=1}^{N}$ without forfeiting the unbiasedness of the resulting estimates.
- We have two possible future directions:
  1. Provide a tractable algorithm that learns unbiased estimates of the value functions.
  2. Utilize the value function approximators as heuristic weights within SMC. A heuristic weight is a weight introduced in a program in order to help guide incremental inference algorithms, like SMC, while an equivalent cancelling weight is introduced at a later point of inference in order to keep the program's distribution invariant (Stuhlmller et al., 2015).

Experiment

- The original hypothesis, that $\frac{p(Y_n|X_{1:n}, Y_{1:n-1})p(X_n|X_{1:n-1}, Y_{1:n-1})}{q_\phi(X_n|X_{1:n-1}, Y_{1:N})}$, could result in poor weights and cause the SMC algorithm to break down was tested.
- A Gaussian linear dynamical system (LDS) formed the prior. The observed random variables were chosen so that all observations but a single outlier are a sensible output of the prior distribution.
- Due to the model being a Gaussian LDS, the optimal proposal distribution, $q_\phi(X_n|X_{n-1}, Y_{1:N})$, i.e. the posterior of the model, was computed. Furthermore, value function approximators were trained along with computing exact value functions, again, possible by the system being a LDS (Bishop, 2006).
- Conclusion: SMC did not break down with the optimal proposal distribution.

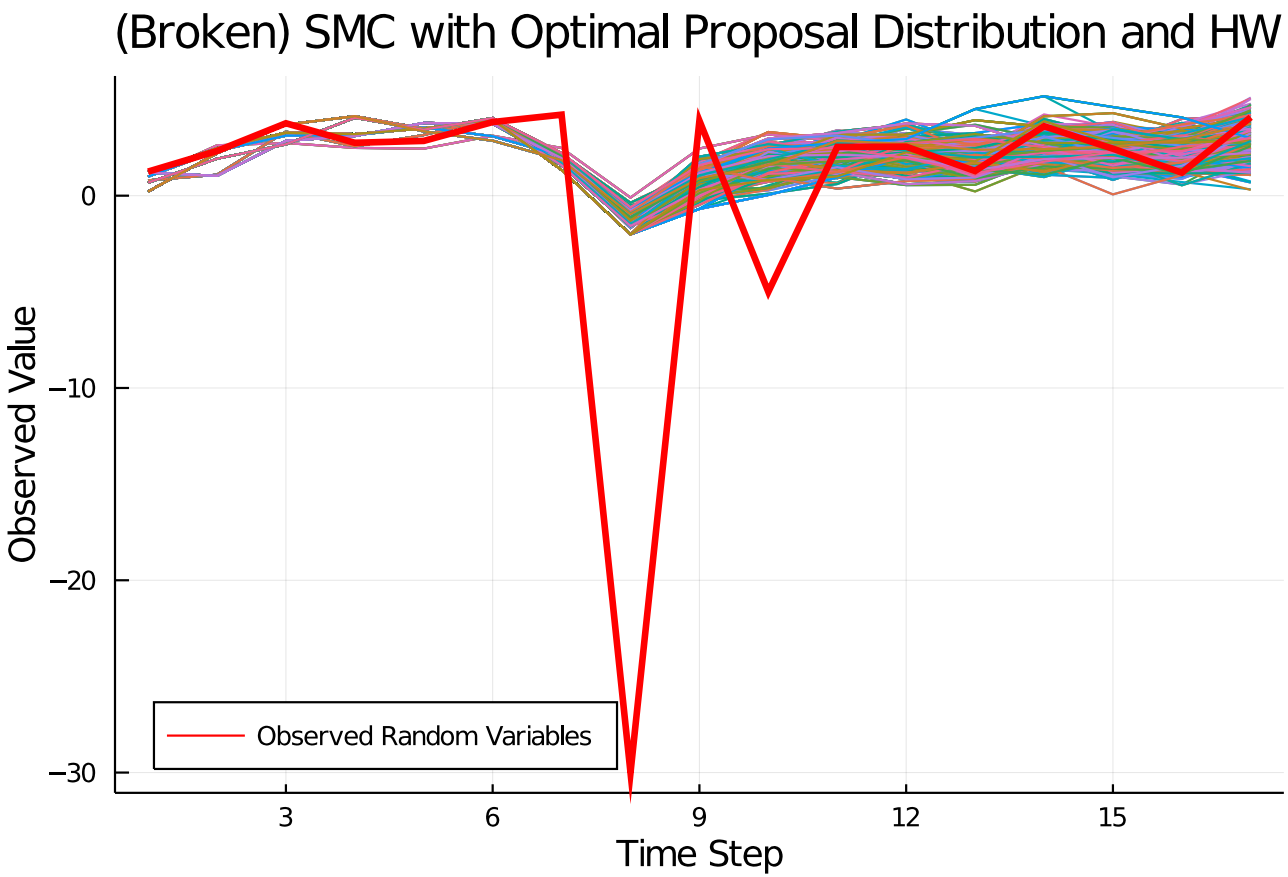
Experiment


SMC with Optimal Proposal Distribution
Observed Value
0
−10
−20
−30
3
6
9
12
15
Time Step
Observed Random Variables

Experiment

- Furthermore, we attempted to see if using the value functions (approximations and exact) as heursitic weights, without correctly backing out of the altered target distributions would result in any benefit. However, this did not seem to improve the performance in any way.

Experiment


(Broken) SMC with Optimal Proposal Distribution and HW
Observed Value
0
−10
−20
−30
3
6
9
12
15
Time Step
Observed Random Variables

Conclusion

1. We showed that the weights of the distributions $\{p(X_{1:n}|Y_{1:N})\}_{n=1}^{N}$ differ from the weights of $\{p(X_{1:n}|Y_{1:n})\}_{n=1}^{N}$ by a ratio of value functions $\frac{V_n(X_{1:n})}{V_{n-1}(X_{1:n-1})}$.
2. We derived a technique for learning approximate value functions, although we do not have guarantees that these trained estimators are unbiased. Hence we cannot compute unbiased weights for $p(X_{1:n}|Y_{1:N})$. Remains to either give an algorithm that learns unbiased value function estimators or fruitfully utilize the value function approximators as heuristic weights.
3. In the limited experiments performed, the scenario where we hoped to see a potential for the value functions to correct a failing SMC algorithm, we did not in fact see SMC break down. Further investigation is necessary to see if the value functions form any useful heuristics, before even examing their correct usage (in terms asymptotic convergence guarantees).

References

1. Bishop, Christopher M. Pattern recognition and machine learning. springer, 2006.
2. Kim, Geon-Hyeong, et al. "Variational Inference for Sequential Data with Future Likelihood Estimates." International Conference on Machine Learning. PMLR, 2020.
3. Rainforth, Tom. "Nesting probabilistic programs." arXiv preprint arXiv:1803.06328 (2018).
4. Stuhlmller, Andreas, et al. "Coarse-to-fine sequential monte carlo for probabilistic programs." arXiv preprint arXiv:1509.02962 (2015).
5. Sutton, Richard S., and Andrew G. Barto. Introduction to reinforcement learning. Vol. 135. Cambridge: MIT press, 1998.