AN IMPLEMENTATION OF OPTIMIZED WEIGHTED ENSEMBLE

FELIX JONES[1], GIDEON SIMPSON[1], DAVID ARISTOFF[2], JEREMY COPPERMAN[3], ROBERT J. WEBBER[4], AND DANIEL M. ZUCKERMAN[3]

1. Desciption of the method

1.1. **Introduction.** Weighted ensemble is an importance sampling interacting particle method that simulates the evolution of a generic Markov chain. The method consists of a fixed number of weighted particles, evolving via repeated *selection* and *mutation* steps.

The mutation step is simply evolution by the underlying Markov chain kernel. The selection step corresponds to resampling from the weighted particles. The resampling is based on *binning* the particles, and it may be designed to keep more particles in important regions of space to reduce the variance of computations of the quantity of interest (QOI). Using a genealogical analogy, we refer to particles before selection as *parents*, and just after selection as *children*.

This implementation of weighted ensemble focuses on optimizing the parameter choices, namely the bins and the number of children to put in each bin.

1.2. **Outline of the algorithm.** In the selection step, each parent is either killed, or copied to produce one or more children. The children are initially identical to their parents, but they evolve forward in time independently. The childrens' weights are adjusted so that the weighted ensemble is not biased. This is implemented as follows.

- The parents are grouped into *bins*, and a target number of children is defined for each bin. We refer to these targets as the *particle allocation*. Both the bins and the particle allocation are user-chosen parameters.
- In each bin, the children are resampled from parents according to their weights. For instance, the children could be obtained by sampling, with replacement, the target number of parents, according to the weight distribution of the parents in the bin.
- In each bin, the children are all given the same weight, which is chosen so that the total weight in the bin remains the same before and after the selection step.

In the mutation step, the children evolve independently according to the law of the underlying Markov chain, becoming the next parents. Their weights remain the same during this step.

1.3. **Algorithm details.** Let K be the underlying Markov kernel, N the total number of particles, and $t \geq 0$ the time, which counts the number of combined selection and mutation steps. Before selection, the particles and their weights are written, respectively,

$$\xi_t^1, \ldots, \xi_t^N \text{ and } \omega_t^1, \ldots, \omega_t^N,$$

After resampling, the particles and weights are denoted

$$\hat{\xi}_t^1, \ldots, \hat{\xi}_t^N \text{ and } \hat{\omega}_t^1, \ldots, \hat{\omega}_t^N.$$

[1]Department of Mathematics, Drexel University
[2]Department of Mathematics, Colorado State University
[3]Department of Biomedical Engineering, Oregon Health & Science University
[4]Courant Institute, New York University
E-mail addresses: fgj23@drexel.edu, grs53@drexel.edu, aristoff@rams.colostate.edu, copperma@ohsu.edu, rw2515@nyu.edu, zuckermd@ohsu.edu.

Weighted ensemble evolves from time t to $t+1$ as follows:

$$\{\xi_t^i\}^{i=1,\dots,N} \xrightarrow{\text{selection}} \{\hat{\xi}_t^i\}^{i=1,\dots,N} \xrightarrow{\text{mutation}} \{\xi_{t+1}^i\}^{i=1,\dots,N},$$
$$\{\omega_t^i\}^{i=1,\dots,N} \xrightarrow{\text{selection}} \{\hat{\omega}_t^i\}^{i=1,\dots,N} \xrightarrow{\text{mutation}} \{\omega_{t+1}^i\}^{i=1,\dots,N}.$$

The initial weights must be strictly positive and sum to one, $\omega_0^1 + \ldots + \omega_0^N = 1$.

The mutation step consists of the particles evolving independently according to K. This means that ξ_{t+1}^j is distributed as $K(\hat{\xi}_t^j, \cdot)$, and the ξ_{t+1}^j are independent over j, conditional on $\hat{\xi}_t^1, \dots, \hat{\xi}_t^N$. The weights do not change during the mutation step,

$$\omega_{t+1}^j = \hat{\omega}_t^j.$$

The selection step requires user-chosen *bins* and *particle allocation*. The bins are labeled by $\mathcal{B}$, and the particle allocation is defined by $N_t(u)$, the number of children in each bin $u \in \mathcal{B}$ at time t. The children in bin $u \in \mathcal{B}$ are obtained by sampling $N_t(u)$ times, with replacement, from the weight distribution of the parents in bin u:

$$Pr(\text{sample } \xi_t^j \text{ in bin } u) = \frac{\omega_t^j}{\omega_t(u)}, \qquad \omega_t(u) := \sum_{i:\text{bin}(\xi_t^i)=u} \omega_t^i.$$

Here $\text{bin}(\xi_t^i) = u$ means ξ_t^i is assigned to bin u. All of the children in any bin have the same weight,

$$\hat{\omega}_t^j = \frac{\omega_t(u)}{N_t(u)}, \qquad \text{if bin}(\hat{\xi}_t^j) = u.$$

Each bin with a parent must have at least one child after selection, and if a bin contains no parents, it can have no children. Thus, $N_t(u) \geq 1$ if and only if $\omega_t(u) > 0$. The total number of particles is always N, so $\sum_{u\in\mathcal{B}} N_t(u) = N$. Total particle weight is also preserved in time.

2. Optimizing the parameters

2.1. **Quantities of interest.** Let (ξ_t) be a Markov chain with kernel K, and let f be a function on its state space. The bin optimization and optimal particle allocation are targeted for the computation of a particular QOI. We consider two types:

Finite time: The QOI is $\mathbb{E}[f(\xi_T)]$, where T is a deterministic time, which is estimated by

$$\mathbb{E}[f(\xi_T)] \approx \sum_{i=1}^N \omega_T^i f(\xi_T^i).$$

Steady state: The QOI is $\int f\, d\mu$, where μ is the stationary distribution of (ξ_t), which we assume is ergodic. Given a user-chosen final time T, weighted ensemble estimates this by

$$\int f\, d\mu \approx \frac{1}{T} \sum_{t=0}^{T-1} \sum_{i=1}^N \omega_t^i f(\xi_t^i).$$

2.2. **Bin optimization.** The bins can be optimized by using a *selection discrepancy function* $H_t = H_t(\xi)$ defined on the particles ξ. The bins are defined so that the values of H_t on the particles in each bin does not vary too much. More precisely:

- The bins are chosen to minimize, over choices of bins, the cost function

$$\sum_{u\in\mathcal{B}} \text{Var}(H_t(\xi_t^i)^{i:\text{bin}(\xi_t^i)=u}),$$

where $H_t(\xi_t^i)^{i:\text{bin}(\xi_t^i)=u}$ is the vector of values of H_t on the particles in bin u.

We implement this by applying k-means to the vector $H_t(\xi_t^i)^{i=1,\dots,N}$ of the values of H_t on all the particles. The k-means clusters are the bins, and k is the user-chosen number of bins.

2.3. **Optimal allocation.** The particle allocation can be optimized via a *selection value function* $V_t = V_t(u)$ defined on the bins u. The particle allocation is defined so that

- The number of children in bin u satisfies

$$N_t(u) \approx \frac{N\omega_t(u)V_t(u)}{\sum_{u\in\mathcal{B}}\omega_t(u)V_t(u)} \qquad \text{and} \qquad N_t(u) \geq 1 \text{ whenever } \omega_t(u) > 0.$$

We achieve this by assigning one child to each bin with a parent, and then allocating the remainder of the children according to the distribution $\omega_t(u)V_t(u)/\sum_{u\in\mathcal{B}}\omega_t(u)V_t(u)$. If the denominator here is zero, we skip the selection step and proceed directly to mutation.

3. Formulas for optimization

3.1. **Feedback formulas.** The selection discrepancy and selection value functions that minimize the variance can be written in terms of the QOI in a feedback form. The feedback expressions we need are as follows.

$$\begin{aligned}
&\textit{Finite time}: h_t(\xi) = \mathbb{E}[f(\xi_T)|\xi_t = \xi] \equiv K^{T-t}f(\xi),\\
&\textit{Steady state}: h(\xi) = \sum_{s=0}^{\infty}\mathbb{E}\left[f(\xi_s) - \int f\,d\mu \,\middle|\, \xi_0 = \xi\right] \equiv \sum_{s=0}^{\infty}\left(K^s f(\xi) - \int f\,d\mu\right).
\end{aligned}$$

Note that h is the solution to the Poisson equation $(\mathrm{Id} - K)h = f - \int f\,d\mu$. Since we will also require variances of the quantities above, we introduce the notation

$$\mathrm{Var}_{K(\xi,\cdot)}g = \mathrm{Var}[g(\xi_{t+1})|\xi_t = \xi] \equiv Kg^2(\xi) - (Kg(\xi))^2,$$

where we will have either $g = h_{t+1}$ or $g = h$.

Because of their feedback form, these functions are not exactly computable in practice. In the code, the expressions above are estimated by using a *coarse model* for K and f.

3.2. **The selection discrepancy and selection value functions.** Exact formulas for the selection discrepancy function $H_t = H_t(\xi)$ that minimizes variance are:

$$\begin{aligned}
&\textit{Finite time}: H_t(\xi) = Kh_{t+1}(\xi) = h_t(\xi),\\
&\textit{Steady state}: H_t(\xi) = Kh(\xi).
\end{aligned}$$

Exact formulas for the selection value function $V_t = V_t(u) > 0$ that minimizes variance are:

$$\begin{aligned}
&\textit{Finite time}: V_t(u)^2 = \sum_{i:\mathrm{bin}(\xi_t^i)=u}\frac{\omega_t^i}{\omega_t(u)}\mathrm{Var}_{K(\xi_t^i,\cdot)}h_{t+1},\\
&\textit{Steady state}: V_t(u)^2 = \sum_{i:\mathrm{bin}(\xi_t^i)=u}\frac{\omega_t^i}{\omega_t(u)}\mathrm{Var}_{K(\xi_t^i,\cdot)}h.
\end{aligned}$$

4. Traditional implementation

This code also includes the traditional weighted ensemble implementation, in which bins are usually defined as Voronoi regions, and the allocation is *uniform*, meaning roughly the same number of children are in each bin, that is, $N_t(u)$ is nearly constant in $u \in \mathcal{B}$.

5. Reference guide

See [1, 3] for details on some of the optimizations described above. See [10] for the original weighted ensemble paper, and [1, 2, 3, 4, 7, 8, 15, 16] for analyses of weighted ensemble. See `https://westpa.github.io/westpa/` for a complete list of weighted ensemble papers. See [17, 18] for the main weighted ensemble code and Github page. See [11, 12, 14] and references therein for a related sequential Monte Carlo method and [6, 14] for optimization in that context. See [5] for details more details on combining weighted ensemble with a coarse model to improve efficiency and accuracy. For a comparison of different resampling techniques, see [9, 13].

References

[1] D. Aristoff. Analysis and optimization of weighted ensemble sampling. *ESAIM: Mathematical Modelling and Numerical Analysis*, 52(2018), 1219–1238, 2018.

[2] D. Aristoff. An ergodic theorem for weighted ensemble. *arXiv preprint arXiv:1906.00856v4*.

[3] D. Aristoff and D.M. Zuckerman, Optimizing weighted ensemble sampling of steady states, *arXiv preprint arXiv:1806.00860v4*, 2019.

[4] D. Bhatt, B. W. Zhang, and D. M. Zuckerman. Steady-state simulations using weighted ensemble path sampling. *The Journal of chemical physics*, 133(1):014110, 2010.

[5] J. Copperman and D.M. Zuckerman. Accelerated estimation of long-timescale kinetics by combining weighted ensemble simulation with Markov model "microstates" using non-Markovian theory. *arXiv preprint arXiv:1903.04673*.

[6] H. Chraibi, A. Dutfoy, T. Galtier, and J. Garnier. Optimal input potential functions in the interacting particle system method. *arXiv preprint arXiv:1811.10450*, 2018.

[7] R. Costaouec, H. Feng, J. Izaguirre, and E. Darve. Analysis of the accelerated weighted ensemble methodology. *Discrete and Continuous Dynamical Systems*, pages 171–181, 2013.

[8] E. Darve and E. Ryu. Computing reaction rates in bio-molecular systems using discrete macro-states. *arXiv preprint arXiv:1307.0763*, 2013.

[9] R. Douc and O. Cappé. Comparison of resampling schemes for particle filtering. In *Image and Signal Processing and Analysis, 2005. ISPA 2005. Proceedings of the 4th International Symposium on*, pages 64–69. IEEE, 2005.

[10] G. A. Huber and S. Kim. Weighted-ensemble brownian dynamics simulations for protein association reactions. *Biophysical journal*, 70(1):97–110, 1996.

[11] P. Del Moral. Feynman-kac formulae: genealogical and interacting particle approximations. *Probability and Its Applications, Springer*, 2004.

[12] P. Del Moral, J. Garnier, et al. Genealogical particle analysis of rare events. *The Annals of Applied Probability*, 15(4):2496–2534, 2005.

[13] R.J. Webber. Unifying Sequential Monte Carlo with Resampling Matrices. *arXiv preprint arXiv:1903.12583*, 2019.

[14] R.J. Webber, D.A. Plotkin, M.E. O'Neill, D.S. Abbot, and J. Weare. Practical rare event sampling for extreme mesoscale weather. *arXiv preprint arXiv:1904.03464*, 2019.

[15] B. W. Zhang, D. Jasnow, and D. M. Zuckerman. The "weighted ensemble" path sampling method is statistically exact for a broad class of stochastic processes and binning procedures. *The Journal of chemical physics*, 132(5):054107, 2010.

[16] D.M. Zuckerman and L.T. Chong. Weighted Ensemble Simulation: Review of Methodology, Applications, and Software. Annu. Rev. Biophys. 46: 43–57, 2017.

[17] M.C. Zwier, J. L. Adelman, J. W. Kaus, A. J. Pratt, K. F. Wong, N. S. Rego, E. Suárez, S. Lettieri, D. W. Wang, M. Grabe, D. M. Zuckerman, and L. T. Chong. Westpa github pack. *https://westpa.github.io/westpa/*, 2015–.

[18] M. C. Zwier, J. L. Adelman, J. W. Kaus, A. J. Pratt, K. F. Wong, N. B. Rego, E. Suárez, S. Lettieri, D. W. Wang, M. Grabe, D.M. Zuckerman, andL.T. Chong. Westpa: An interoperable, highly scalable software package for weighted ensemble simulation and analysis. *Journal of chemical theory and computation*, 11(2):800–809, 2015.