Bulletproofing Bayesian particle flow against stiffness

Particle flow is an algorithm that moves a random set of particles to a desired probability density corresponding to the action of Bayes’ rule. The law of motion for our particles is computed as the solution to a smooth linear highly underdetermined PDE. This is similar to optimal transport theory, but it differs in several ways: (1) we do not seek a unique solution of our PDE by minimizing a convex functional; (2) the flow of particles is stochastic, and (3) the real time computational complexity of particle flow is many orders of magnitude less than optimal transport for high dimensional problems. We can apply particle flow to any problem in which Bayes’ rule appears, e.g. Bayesian estimation, Bayesian deep learning and Bayesian reinforcement learning. The best introduction to particle flow is the following video: <https://www.youtube.com/watch?v=vqJGB47XoeY>. **We strongly recommend that you look at this video in order to understand what we are talking about. Reading our papers (or even trying to read the following summary) before viewing the video is generally bad for your health.**

But the flow of particles can be extremely stiff, and hence we would like to mitigate such stiffness for a given flow or else invent new flows that are less stiff. There are many possible methods to attempt to do this, including: (1) implicit numerical integration; (2) adaptive integration step size; (3) non-adaptive but highly non-uniform integration step size; (4) use principal coordinates or approximately principal coordinates; (5) Tychonov regularization or shrinkage of the Hessian of log p, in which p is the conditional probability density; (6) derive a new flow that has much less stiffness (e.g., as in [4]); (7) combinations of the above methods. We have tried all of these methods; see the references for details. Our PDE is not stiff (as far as we know), but rather the flow of particles resulting from the solution of the PDE is stiff. Hence, this is a question of stiffness of a stochastic ODE. We expect that method #6 will be the most fruitful and interesting.

What is particle flow and why do we like it? The very best quick answer to such questions is in the video. But we shall give an all-to-brief summary here. Particle flow reduces the computational complexity of Bayes’ rule by many orders of magnitude for high dimensional problems, and it also is embarrassingly parallelizable, in contrast with standard methods. It fixes the problem of particle degeneracy, which means that the random set of particles might be great to represent a probability density, but after the application of Bayes’ rule it becomes useless because the particles are in the wrong place. See the video! We fix this problem by designing a law of motion to move the particles to the correct place in d-dimensional space to represent the conditional density accurately. This is like physics, but we are picking the law of motion rather than Nature. This law of motion is given by an Itô stochastic differential equation:

(1)

in which f(x, λ) is the d-dimensional drift, x(λ) is the d-dimensional state vector of the dynamical system of interest, and w(λ) is a random process (Brownian motion) with the covariance matrix:

(2)

It turns out that we can compute the set of allowable functions, f and Q, that implements Bayes’ rule, by solving the following PDE:

in which p is the conditional probability density of the d-dimensional state vector (x), h is the likelihood for the measurement, where both p and h are known functions of x, and the two unknown functions are the d-dimensional drift (f) and the d x d covariance matrix (Q) of the Brownian motion (w) in our flow of particles. Where did this PDE come from? Watch the video! But in a nutshell, we construct a simple algebraic formula to tell the mathematics that we want to evolve the probability density from the prior to the posteriori under the action of Bayes’ rule (as λ continuously evolves from 0 to 1). We then plug this condition into the Fokker-Planck equation, which describes the evolution of the conditional probability density of x as λ goes from 0 to 1, continuously. Finally, we want to kill the normalization term (because it is nothing but trouble), and hence we take the derivative of our PDE with respect to x. To understand why we hate the normalization of the conditional probability density, watch the video! This results in the PDE (3). This is a stochastic transport equation. It is much simpler if Q = 0, but not nearly as good. To understand why, watch the video!

The PDE (3) is highly underdetermined, and hence it has many solutions (f, Q); just count the number of unknowns and equations (although this is not always a reliable method to decide such questions; see page XIII in Werner Seiler’s lovely book “Involution” Springer-Verlag 2010). Some of these solutions are stiff and others are not. For example, one of our favorite solutions is:

However, this flow can be extremely stiff. That is, the condition number of the Jacobian of f can be huge for many important real world applications. In particular, if p is Gaussian, then minus the inverse of the Hessian is the error covariance matrix of the state vector, whose largest eigenvalue is about 18 orders of magnitude bigger than the smallest eigenvalue for such applications; hence the condition number of the Jacobian is enormous; see [1], [2] and [4] for details. On the other hand, there is no reason to suppose that the PDE (3) is stiff. In particular, some other solutions of this same PDE are not stiff (e.g., see [4]). Moreover, there is no theorem that says that stiff ODEs must arise from the solutions of stiff PDEs. As far as we know, our PDE is not stiff. In any case, we do not solve our PDE numerically, because the computational complexity of doing so would be counterproductive for high dimensional problems. But rather we solve our PDE as an exact formula, e.g., (4) and (5), and thus we do not need to resort to numerical solutions.

**The following table is dramatic numerical evidence that the flow of particles defined by (4) and (5) can be extremely stiff for certain problems**. That is, the stochastic ODE (1) can be extremely stiff. This table summarizes a comparison of 18 different methods to numerically integrate our ODE (1). The numbers in this table are filter errors. In particular, the explicit method (i.e., Euler integration) with a constant step size in λ is a complete disaster; the errors for Euler integration are 6 orders of magnitude worse than the best methods, which use implicit numerical integration and/or adaptive step size in λ and/or highly non-uniform but non-adaptive step size in λ (i.e., exponentially increasing step size). More details on this table are in [2].



The results in this table suggest that our ODE can be extremely stiff. One common definition of “stiffness” for ODEs is that explicit methods don’t work, whereas implicit methods are much better. We do not completely agree with this definition (see [1] for details), but it is widely used in the literature on stiffness. We can also conclude from this table that we can greatly improve the accuracy of the flow by using better methods, at the cost of increased computer throughput. We emphasize that this is only one of 7 methods to attempt to mitigate stiffness [1]. In particular, by designing a better particle flow that is less stiff we can dramatically improve the situation without any increase in computational complexity. For example, the blue curves in the following plots gives the stiffness and one-sigma estimation errors for one of our favorite particle flows, defined by (4) and (5), whereas the other curves are for particle flows designed specifically to reduce

Chart

Description automatically generated

stiffness (see [4] for details). Apparently, the new theory in [4] has made a significant improvement in filter accuracy by reducing the stiffness of the flow. But the theory in [4] is not good enough for other important applications. We want you to improve upon the theory in [4] by inventing new better stochastic particle flows that avoid the limitations of the methods in [4]. That is, start with our PDE (3), but design flows (f and Q) that are less stiff. We want you to bulletproof Bayesian particle flow against stiffness. **We strongly recommend that you thoroughly study [4] in order to understand what is known and what are open research questions. This avoids reinventing the wheel, and it might actually help guide your research.**

**We are not interested in the following**: (a) numerical solutions to our PDE; (b) optimal transport; (c) regularization of our PDE to obtain a unique solution; (d) theorems about existence or uniqueness or regularity of our PDE; (e) asymptotic error bounds (but we would be delighted to see tight non-asymptotic error bounds with explicit numerical values for all constants); (f) numerical methods to integrate stiff ODEs (we already studied this in depth, e.g. see [2]); (g) Stein’s method, because deterministic Stein’s method suffers from particle collapse [5], and the computational complexity of stochastic Stein’s method does not scale well with dimension [5]; (h) transport or stochastic transport methods that assume log-concave densities (because this rules out multi-modal conditional probability densities); (i) any method that resamples particles, because resampling is the kiss of death for parallelization using GPUs or other parallel computers (our particle flow never resamples particles, because we flow the particles to the correct region of d-dimensional space to represent the conditional probability densities efficiently, and hence we do not need to resample).

There are many **applications of Bayesian particle flow**, as explained in [4] and our video, including: signal processing, Bayesian estimation, Bayesian deep learning [6], Bayesian reinforcement learning, communication systems, navigation systems, control systems, robotics, prognostics, weather forecasting, climate prediction, financial engineering, quantum computing [7], quantum communication [7], quantum navigation [7] and quantum radar [7].

[1] Fred Daum and Jim Huang, “Seven dubious methods to mitigate stiffness in particle flow,” Proceedings of SPIE Conference on Signal Processing, Baltimore June 2014.

[2] David Crouse, “Particle flow filters: biases and bias avoidance,” IEEE International Information FUSION Conference, Ottawa Canada, July 2019.

[3] Fred Daum, Jim Huang and Arjang Noushin, “new theory and numerical experiments for Gromov’s method,” IEEE Information FUSION Conference, Cambridge England, July 2018.

[4] Liyi Dai and Fred Daum, “On the Design of Stochastic Particle Flow Filters,” IEEE Transactions on Aerospace & Electronic Systems, (early on-line access) October 2022.

[5] Jianyi Zhang, Ruiyi Zhang, Lawrence Carin and Changyou Chen, “Stochastic Particle-Optimization Sampling and the Non-Asymptotic Convergence Theory,” Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics (AISTATS) 2020, Palermo, Italy.

[6] Fred Daum, Jim Huang, Arjang Noushin, "Extremely deep Bayesian learning with Gromov's method," VIDEO and paper, Proceedings of SPIE Conference on Signal Processing, Baltimore Maryland, 7 May 2019.

[7] Fred Daum, “Bayesian quantum mechanics,” VIDEO of Distinguished Lecture for IEEE Aerospace and Electronic Systems Society, 13 July 2022.