Literature review: Implementing variance reduced stochastic gradient Langevin dynamics

Ayoub El Hanchi

November 6, 2019

1 Introduction and Motivation

The ultimate goal of machine learning and statistics is to perform optimal decision making under uncertainty. For problems that can be solved exactly given enough data, uncertainty quantification is usually overlooked under the rational that perfect performance alleviates the need of such quantification. Unfortunately, many interesting problems do not fit in that category, and some applications require transparent uncertainty estimation (medical diagnosis, self driving cars, and others). Probabilistic machine learning provides a way of quantifying this uncertainty, but it has not been as popular recently due to its inability to scale well to very large datasets.

The growth of the size of the datasets, as well as the need for fast algorithms, led to the use of variational methods as a replacement to the more traditional Markov chain Monte Carlo (MCMC) algorithms. The problem, however, is that the error incurred by the use of the variational approximations cannot be quantified, making inferences made from these approximations unreliable in critical applications. On the other hand, traditional MCMC methods require full data computations at each iteration, making them unusable for large problems. This situation is what motivates the need for algorithms that have both a sub-linear time complexity, and have provable guarantees on the error.

In [30, 22], a new framework is proposed for constructing approximate MCMC methods. In addition to the argument above, the authors offer the following appealing statistical argument. Traditional MCMC estimators are (asymptotically) unbiased, but given their high computational cost, only a few samples (if any) can be collected, so that the estimators suffer from high variance. Presented this way, one can easily recognize the classical bias variance trade-off, and it immediately becomes obvious that one can do better if we construct approximate MCMC methods that are not necessarily unbiased.

The authors propose two algorithms that fit in this framework. The first is a stochastic version of the Metropolis-Hastings algorithm where the accept/reject step is treated as a statistical decision problem, and dealt with using hypothesis testing. The second is stochastic gradient Langevin dynamics (SGLD), a direct analog of stochastic gradient descent (SGD) for sampling, and which will be the focus of our discussion. SGLD scales well to large datasets, and the user of the algorithm can directly control the bias/variance of the resulting estimator by adjusting the step size.

While it satisfies the requirements of the framework presented above, SGLD suffers from a strong bias when used with computationally feasible step sizes [6, 24]. Many variants of SGLD were proposed since then, but most do not directly address this issue [9, 15, 2]. Only recently was this problem directly addressed using variance reduction methods originally developed in optimization. We give an account of the theoretical developments in the next section. In the remaining section we discuss computational aspects of the resulting variance reduced algorithms, and survey some of the software platforms that are relevant.

2 Theoretical developments and open problems

Much like SGD is a stochastic approximation of gradient descent (GD), which is itself the discretization of gradient flow, SGLD is a stochastic approximation of the discretization of the Langevin dynamics (LD), which can be interpreted as the analog of gradient flow in the space of probability measures [21, 31]. Furthermore, given the strong resemblance of the update equation of SGLD with that of SGD, optimization ideas became suddenly relevant to the subsequent development of SGLD. We quickly review some results in optimization and draw the parallel with similar results in sampling. We restrict our discussion here to the smooth and strongly convex case, although more general results exist in the literature.

In optimization, it is well known that SGD has a rate of convergence of $O(\frac{1}{k})$, which is much worse than the linear convergence rate of GD. Variance reduction methods such as SAG [28], SAGA [14], and SVRG [20] succeed in recovering the linear rate of GD, but at the cost of worst constants. On the other hand, Nesterov acceleration [25], which has the same computational cost as gradient descent, is able to achieve a better rate than GD. Variance reduction methods for the accelerated case also exist and achieve the same accelerated rate [3, 32, 13].

On the sampling side, [11] was the first to prove a non-asymptotic convergence rate for discretized Langevin dynamics, and to give a precise quantification of its bias. [12] generalized this result to the case where only noisy estimates of the gradient are available, which covers SGLD as a special case. [16, 8] analyzed variance reduced SGLD using SAGA

and SVRG. [23, 10] study underdamped Langevin dynamics, which can be viewed as the analog of Nesterov acceleration for sampling, and prove a faster convergence rate.

Many problems in this area are still open, such as the choice of optimal batch size, the optimal choice of estimator, and ways to reduce the bias of LD. Only recently in optimization have such results being published [18, 17], and extensions to the sampling case should follow soon.

3 Computational aspect and current software

The only package that implements SGLD and a few of its variants, to my knowledge, is sgmcmc [4], which is written in R, and is not updated with the recent advances in the field. Since it is built on raw R, it does not have support for automatic differentiation, nor does it support the use of GPUs which makes it inadequate for large datasets.

Among the most popular software frameworks for probabilistic machine learning, we find Stan [7], Edward [29], Pyro [5], and PyMC3 [27]. Stan uses the NUTS sampler [19] by default, and has an implementation of mean field variational Bayes, but has no support for data subsampling. Edward on the other hand entirely relies on variational inference, while Pyro and PyMC3 offer both the NUTS sampler and the variational approximations. All of the last three support data subsampling for variational inference, but not for MCMC. Among the four frameworks, only Pyro and Edward offer support for GPUs, using Tensorflow [1] as a backend for Edward and PyTorch [26] for Pyro. Tensorflow has an implementation of SGLD, but only for optimization. None of the other major frameworks have an implementation of SGLD or any of its variance reduced version. In fact, none of them has an implementation for variance reduced optimization algorithms either.

The reason for the absence of these implementations is that none of these major software platforms is well suited for the implementation of variance reduced optimization and sampling algorithms. Almost all of these variance reduced algorithms require the evaluation of the gradients per example, whereas automatic differentiation engines do not support this functionality, and only return the gradient with respect to the whole mini-batch. Requesting gradients individually is very wasteful and scales very poorly. Also, many of these variance reduced algorithms require the storage of the gradients per training example. This is not feasible in any reasonably large problem, at least if done in the naive way.

The way I plan to overcome the above problems is as follows. First, it is well known that in linear models, it is enough to store the derivative of the activation function for each example. The gradient for each example can then be recovered by multiplying this derivative with the corresponding feature vector. This idea has a straightforward generalization

when we have multiple linear layers instead of just one. Similarly, convolutional layers can be viewed as linear layers, and the same trick can be used. This solves the memory problem. As for recovering the per example gradients, for linear and convolutional layers, one can show that it is enough to intercept the gradients coming from the loss at the activations of each layer. The individual gradients can then be reconstructed by multiplying these gradients with the inputs of the layers. I will present the full idea in the final report, and elaborate on it properly.

References

[1] Martin Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, Manjunath Kudlur, Josh Levenberg, Rajat Monga, Sherry Moore, Derek G. Murray, Benoit Steiner, Paul Tucker, Vijay Vasudevan, Pete Warden, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: A system for large-scale machine learning, 2016.

[2] Sungjin Ahn, Anoop Korattikara, and Max Welling. Bayesian Posterior Sampling via Stochastic Gradient Fisher Scoring. Technical report.

[3] Zeyuan Allen-Zhu. Katyusha: The first direct acceleration of stochastic gradient methods. *Journal of Machine Learning Research*, 18:1–51, jun 2018.

[4] Jack Baker, Paul Fearnhead, Emily B. Fox, and Christopher Nemeth. sgmcmc: An R Package for Stochastic Gradient Markov Chain Monte Carlo. oct 2017.

[5] Eli Bingham, Jonathan P. Chen, Martin Jankowiak, Fritz Obermeyer, Neeraj Pradhan, Theofanis Karaletsos, Rohit Singh, Paul Szerlip, Paul Horsfall, and Noah D. Goodman. Pyro: Deep Universal Probabilistic Programming. oct 2018.

[6] Nicolas Brosse, Alain Durmus, and Eric Moulines. The promises and pitfalls of Stochastic Gradient Langevin Dynamics. nov 2018.

[7] Bob Carpenter, Daniel Lee, Marcus A Brubaker, Allen Riddell, Andrew Gelman, Ben Goodrich, Jiqiang Guo, Matt Hoffman, Michael Betancourt, and Peter Li. Journal of Statistical Software Stan: A Probabilistic Programming Language. Technical report.

[8] Niladri S. Chatterji, Nicolas Flammarion, Yi-An Ma, Peter L. Bartlett, and Michael I. Jordan. On the Theory of Variance Reduction for Stochastic Gradient Monte Carlo. feb 2018.

[9] Tianqi Chen, Emily B Fox, and Carlos Guestrin. Stochastic Gradient Hamiltonian Monte Carlo. Technical report.

[10] Xiang Cheng, Niladri S. Chatterji, Peter L. Bartlett, and Michael I. Jordan. Underdamped Langevin MCMC: A non-asymptotic analysis. jul 2017.

[11] Arnak S Dalalyan. Theoretical guarantees for approximate sampling from smooth and log-concave densities. Technical report.

[12] Arnak S. Dalalyan and Avetik G. Karagulyan. User-friendly guarantees for the Langevin Monte Carlo with inaccurate gradient. sep 2017.

[13] Aaron Defazio Ambiata and Sydney Australia. A Simple Practical Accelerated Method for Finite Sums. Technical report.

[14] Aaron Defazio Ambiata, Francis Bach, and Simon Lacoste-Julien. SAGA: A Fast Incremental Gradient Method With Support for Non-Strongly Convex Composite Objectives. Technical report.

[15] Nan Ding, Youhan Fang, Ryan Babbush, Changyou Chen, Robert D Skeel, and Hartmut Neven. Bayesian Sampling Using Stochastic Gradient Thermostats. Technical report.

[16] Avinava Dubey, Sashank J Reddi, Barnabás Póczos, Alexander J Smola, Eric P Xing, and Sinead A Williamson. Variance Reduction in Stochastic Gradient Langevin Dynamics. Technical report.

[17] Nidham Gazagnadou, Robert M. Gower, and Joseph Salmon. Optimal mini-batch and step sizes for SAGA. jan 2019.

[18] Robert M. Gower, Peter Richtárik, and Francis Bach. Stochastic Quasi-Gradient Methods: Variance Reduction via Jacobian Sketching. may 2018.

[19] Matthew D Hoffman and Andrew Gelman. The No-U-Turn Sampler: Adaptively Setting Path Lengths in Hamiltonian Monte Carlo. Technical report, 2014.

[20] Rie Johnson and Tong Zhang. Accelerating Stochastic Gradient Descent using Predictive Variance Reduction. Technical report.

[21] Richard Jordan, David Kinderlehrer, Felix Otto, and Siam J M A T H An A L. THE VARIATIONAL FORMULATION OF THE FOKKER-PLANCK EQUATION * In memory of Richard Duffin. Technical Report 1, 1998.

[22] Anoop Korattikara, Yutian Chen, and Max Welling. Austerity in MCMC land: Cutting the metropolis-hastings budget. In *31st International Conference on Machine Learning, ICML 2014*, volume 1, pages 321–336. International Machine Learning Society (IMLS), 2014.

[23] Yi-An Ma, Niladri Chatterji, Xiang Cheng, Nicolas Flammarion, Peter Bartlett, and Michael I. Jordan. Is There an Analog of Nesterov Acceleration for MCMC? feb 2019.

[24] Tigran Nagapetyan, Andrew B. Duncan, Leonard Hasenclever, Sebastian J. Vollmer, Lukasz Szpruch, and Konstantinos Zygalakis. The True Cost of Stochastic Gradient Langevin Dynamics. jun 2017.

[25] Yu Nesterov. Introductory Lectures on Convex Programming Volume I: Basic course. Technical report, 1998.

[26] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary Devito Facebook, A I Research, Zeming Lin, Alban Desmaison, Luca Antiga, Orobix Srl, and Adam Lerer. Automatic differentiation in PyTorch. Technical report.

[27] John Salvatier, Thomas V. Wiecki, and Christopher Fonnesbeck. Probabilistic programming in Python using PyMC3. *PeerJ Computer Science*, 2016(4), 2016.

[28] Mark Schmidt, Nicolas Le Roux, and Francis Bach. Minimizing Finite Sums with the Stochastic Average Gradient. sep 2013.

[29] Dustin Tran, Matthew D. Hoffman, Rif A. Saurous, Eugene Brevdo, Kevin Murphy, and David M. Blei. Deep Probabilistic Programming. jan 2017.

[30] Max Welling, D Bren, and Yee Whye Teh. Bayesian Learning via Stochastic Gradient Langevin Dynamics. Technical report, 2010.

[31] Andre Wibisono. Sampling as optimization in the space of measures: The Langevin dynamics as a composite optimization problem. feb 2018.

[32] Kaiwen Zhou. Direct Acceleration of SAGA using Sampled Negative Momentum. jun 2018.