Second-order stochastic variational inference

Jeffrey Regier
jeff@stat.berkeley.edu

Jon McAuliffe
jon@stat.berkeley.edu

Variational inference finds an approximation to the posterior distribution among a class of distributions through numerical optimization [1]. The variational objective function $\mathcal{L}$ is an expectation with respect to latent variables z that follow an approximating distribution q:

$$\mathcal{L} := \mathbb{E}_q \left\{ \log p(x, z) - \log q(z) \right\}. \tag{1}$$

Candidate approximating distributions $q_\omega := q$ are parameterized by a real-valued vector ω. If the expectation has an analytic form, $\mathcal{L}$ may be maximized by deterministic optimization methods, such as coordinate ascent and Newton's method. Realistic Bayesian models, however, not selected primarily for computational convenience, seldom have variational objective functions with analytic forms.

Stochastic optimization is an alternative. For many common classes of approximating distributions, $q_\omega = h_\omega(\epsilon)$, in law, for a base distribution ϵ that does not depend on ω and a deterministic function h_ω [2, 3, 4, 5, 6]. At each iteration of an optimization procedure, ω is updated based on an unbiased Monte Carlo approximation to the objective function:

$$\hat{\mathcal{L}}(\omega; e_1, \ldots, e_N) := \frac{1}{N} \sum_{i=1}^{N} \left\{ \log p(x, h_\omega(e_i)) - \log q_\omega(h_\omega(e_i)) \right\}. \tag{2}$$

for $e_1, \ldots, e_N$ sampled from the base distribution ϵ.

Stochastic gradient descent (SGD), the workhorse of stochastic optimization, is slow in theory (sub-linear convergence) and in practice (thousands of iterations), intuitively for two reasons: 1) Its learning rate schedule is fixed a priori and decays rapidly enough to 0 that is square-summable. This learning rate schedule limits the step size and hence the rate of convergence for a Lipschitz objective function. 2) It fails to account for the objective's curvature. Extensions to SGD like AdaGrad [7] and Adam [8] adjust for the relative scales of the parameters but not the curvature in general.

In deterministic settings, second-order methods converge quadratically, typically reaching machine precision after tens of iterations. They do not follow fixed learning rate schedules and they adjust for curvature.

Second-order methods are less studied in stochastic settings. Most existing second-order stochastic methods are not directly applicable to Monte Carlo approximations of the variational objective. Some methods require that the objective is a finite sum [9, 10]; the variational objective is an expectation over continuous values. Other methods allow stochastic estimates of the Hessian, but require exact gradients [11, 12, 9]. Other methods apply exclusively to convex problems [9, 13]; variational objectives are non-convex in general. Still other second-order stochastic methods limit their rates of convergence by adhering to a fixed square-summable learning rate schedule [14], thus reproducing a primary shortcoming of SGD While this does cause the algorithm to converge despite the noise, it recreate a major source of slow convergence of SGD. Others lack convergence guarantees [15]. Others base their estimates of the Hessian on the difference between gradients during subsequent iterations, rather than exploiting current second-order information [13]. In practice, variational objectives' second derivatives can change dramatically between iterations, making Hessian estimates based on history misleading. Fortunately, the Hessian often can be computed efficiently each iteration by reusing expensive intermediate computations from the gradient. Furthermore, most optimization algorithms based on BFGS-style Hessian approximations only apply to convex optimization.

We adapt the STORM framework for stochastic optimization [16] to variational inference. We call the resulting algorithm Second-order Stochastic Variational Inference (SOS-VI). SOS-VI (Algorithm 1)

Algorithm 1 Second-order Stochastic Variational Inference

Require: ω is the initial vector of variational parameters; $\delta \in (\delta_{\min}, \delta_{\max})$ is the initial trust-region radius; $\gamma > 1$ is the trust region expansion factor; and $\eta_1 \in (0, 1)$ and $\eta_2 > 0$ are constants.

```
for $i \leftarrow 1$ to $M$ do
    Sample $e_1, \ldots, e_N$ iid from base distribution $\epsilon$.
    $g \leftarrow \nabla_\nu \hat{\mathcal{L}}(\nu; e_1, \ldots, e_N)|_\omega$
    $H \leftarrow \nabla^2_\nu \hat{\mathcal{L}}(\nu; e_1, \ldots, e_N)|_\omega$
    $\omega' \leftarrow \arg\max_\nu \{g^\intercal \nu + \nu^\intercal H \nu : \|\nu\| \leq \delta\}$    ▷ non-convex quadratic optimization
    $\beta \leftarrow g^\intercal \omega' + \omega'^\intercal H \omega'$    ▷ the expected improvement
    Sample $e'_1, \ldots, e'_N$ iid from base distribution $\epsilon$.
    $\alpha \leftarrow \hat{\mathcal{L}}(\omega'; e'_1, \ldots, e'_N) - \hat{\mathcal{L}}(\omega; e'_1, \ldots, e'_N)$    ▷ the observed improvement
    if $\alpha/\beta > \eta_1$ and $\|g\| \geq \eta_2 \delta$ then
        $\omega \leftarrow \omega'$
        $\delta \leftarrow \max(\gamma\delta, \delta_{max})$
    else
        $\delta \leftarrow \delta/\gamma$
    if $\delta < \delta_{\min}$ or $i = M$ then return $\omega$
```

inherits convergence guarantees from STORM. Update steps for SOS-VI are restricted to a trust region that grows or shrinks based on an assessment of model quality during previous steps. In SOS-VI, as opposed to STORM, the assessment of a model's quality is based on a match pairs experiment: $e'_1, \ldots, e'_N$ are used to evaluate $\mathcal{L}$ at both ω and ω'. This dramatically improves rate of convergence as q approaches the optimizer, and randomness from ϵ comes to dominate the observed improvement. The Hessian is approximated by sampling, not by finite differencing gradients, or by building a derivative-free model, all of which STORM allows.

Experiments

We compare SOS-VI to Automatic Differentiation Variational Inference (ADVI) [6] on 107 different Bayesian models and datasets. ADVI is based on SGD. We use the creator's implementation of ADVI, provided in STAN, and implement SOS-VI within the STAN framework as well. We average $N = 100$ samples from the variational distribution to approximate the objective function and its derivatives. Both methods were always run well past the point of making any meaningful progress at maximizing the objective function.

For 76 of the 107 models, on a set of 10 runs, the means of $\hat{\mathcal{L}}$ at the optimizers found by ADVI and SOS-VI did not differ at the 95% confidence level, using unpooled estimates of standard error. For 27 models, SOS-VI converged to significantly better optimizers than ADVI. For 4 models, ADVI finds better local optima, perhaps due to greater exploration of the parameter space. When ADVI outperforms SOS-VI, it was always by at most 4 standard deviations. On some models, SOS-VI outperforms ADVI by as many as 12 standard deviations.

SOS-VI iterations are computational more expensive, but not always by much. SOS-VI approximates the Hessian, but often the most expensive computation needed to compute the Hessian, such as exponentiation, is already calculated to approximate the gradient. Each SOS-VI iteration also evaluates the objective function twice. Algorithm 1 requires non-convex optimization as a subroutine, but it involves no additional evaluations of the objective function, which typically dominates runtime.

To compare runtimes, we measure the number of iterations until each method's iterations are consistently within 2 standard errors of the maxima ultimately found by the worse method. For 7 models, ADVI alone failed to remain within 2 standard errors of the optima it found after 10,000 for iterations, for any sequence of 20 consecutive iterations. We exclude these models from subsequent comparison. For the remaining 100 models, we average the number of iterations for each method to terminate. This unweighted average iterations counts effectively weights models where both ADVI and SOS-VI took many iterations to converge: difficult models matter more. ADVI takes 1131 iterations on average to converge, whereas SOS-VI takes 19 iterations, a 68-fold improvement.

References

[1] David M Blei, Alp Kucukelbir, and Jon D McAuliffe. Variational inference: A review for statisticians. *arXiv preprint arXiv:1601.00670*, 2016.

[2] James C. Spall. *Introduction to Stochastic Search and Optimization: Estimation, Simulation, and Control*. 2005.

[3] Diederik Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[4] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. *arXiv preprint arXiv:1401.4082*, 2014.

[5] Michalis Titsias and Miguel Lázaro-Gredilla. Doubly stochastic variational bayes for non-conjugate inference. In *Proceedings of the 31st International Conference on Machine Learning*, 2014.

[6] Alp Kucukelbir, Rajesh Ranganath, Andrew Gelman, and David Blei. Automatic variational inference in stan. In *Advances in neural information processing systems*, 2015.

[7] John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 2011.

[8] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[9] Naman Agarwal, Brian Bullins, and Elad Hazan. Second order stochastic optimization in linear time. *arXiv preprint arXiv:1602.03943*, 2016.

[10] Jascha Sohl-Dickstein, Ben Poole, and Surya Ganguli. Fast large-scale optimization by unifying stochastic gradient and quasi-newton methods. In *Proceedings of the 31st International Conference on Machine Learning*, 2014.

[11] Murat A Erdogdu and Andrea Montanari. Convergence rates of sub-sampled newton methods. In *Advances in Neural Information Processing Systems*, 2015.

[12] Richard H Byrd, Gillian M Chin, Will Neveitt, and Jorge Nocedal. On the use of stochastic hessian information in optimization methods for machine learning. *SIAM Journal on Optimization*, 2011.

[13] Nicol N Schraudolph, Jin Yu, Simon Günter, et al. A stochastic quasi-newton method for online convex optimization. In *AISTATS*, 2007.

[14] Richard H Byrd, SL Hansen, Jorge Nocedal, and Yoram Singer. A stochastic quasi-newton method for large-scale optimization. *SIAM Journal on Optimization*, 2016.

[15] James Martens. Deep learning via hessian-free optimization. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, 2010.

[16] Ruobing Chen, Matt Menickelly, and Katya Scheinberg. Stochastic optimization using a trust-region method and random models. *arXiv preprint arXiv:1504.04231*, 2015.