Accelerating SVRG via second-order information

Ritesh Kolte
Department of Electrical Engineering
Stanford University, CA 94305 USA
rkolte@stanford.edu

Murat Erdogdu
Department of Statistics
Stanford University, CA 94305 USA
erdogdu@stanford.edu

Ayfer Özgür
Department of Electrical Engineering
Stanford University, CA 94305 USA
aozgur@stanford.edu

Abstract

We consider the problem of minimizing an objective function that is a sum of convex functions. For large sums, batch methods suffer from a prohibitive per-iteration complexity, and are outperformed by incremental methods such as the recent variance-reduced stochastic gradient methods (e.g. SVRG). In this paper, we propose to improve the performance of SVRG by incorporating approximate curvature information while maintaining a per-iteration complexity that is linear in the dimension. An option which we find to perform remarkably, is to combine SVRG with LBFGS updates, in a manner that is different from existing approaches. Numerical experiments on real datasets demonstrate the improvements due to proper utilization of approximate second-order information.

1 Introduction

We consider the problem of minimizing a strongly-convex objective function that is the sum of n smooth convex functions,

$$\underset{w\in\mathbb{R}^p}{\text{minimize}} \left\{ \phi(w) \triangleq \frac{1}{n}\sum_{i=1}^{n} \phi_i(w) \right\}, \tag{1}$$

a common example of this situation being regularized empirical risk minimization (e.g. $\phi_i(w) = \ell(a_i^T w) + \frac{\lambda}{2}\|w\|^2$ for some loss function $\ell(\cdot)$). We mainly focus on the regime $n \gg p \gg 1$. In this regime, even first-order batch methods like gradient descent that have per-iteration complexity $O(np)$ become computationally burdensome. As a result, incremental/stochastic methods with per-iteration complexity $O(p)$ such as stochastic gradient (SGD) are of interest. The problem of slow convergence of SGD due to high variance of the gradient estimate has been addressed successfully by the recent development of various variance-reducing stochastic gradient methods such as SVRG [8], SAG [13], SDCA [15] and Nesterov-accelerated counterparts [14, 17, 9], resulting in linear convergence[1]. Some of these methods require the functions $\phi_i(\cdot)$ to have a specific structure, but we do not need to make these assumptions.

Naturally, there have also been quite a few recent works that aim to accelerate convergence of stochastic methods by incorporating curvature information [3, 4, 16, 10, 11], while maintaining a per-iteration complexity of $O(p)$. Note that [16] suffers from a high memory requirement and expensive per-iteration complexity. The recent works [11, 10] are closely related to ours. These works

[1]Note that the convergence here is with respect to the point achieving the minimum of (1), i.e. we focus on minimizing the training error.

propose to combine SVRG with frequent limited-memory BFGS (LBFGS) updates of the Hessian estimate, either by making use of differences in stochastic gradients or subsampled-Hessian-vector products. However, the noise injected while estimating the differences between gradients results in inaccurate curvature estimates that can compromise the performance of these algorithms, especially when improvement in objective can be attained only by moving in low curvature directions. This is because the resulting curvature estimates in the low curvature directions end up being dominated by the noise. We instead propose to perform an LBFGS update only at the beginning of the SVRG-epoch, using the full gradients. Although this leads to less frequent updates of the approximate Hessian, we observe that it results in more stable and much faster convergence.

Another approach that we experiment with is to incorporate curvature information via periodic computations of a subsampled Hessian (only once per epoch). The use of a subsampled Hessian has indeed been proposed previously [3], where some iterations of the CG method are used to approximately compute the product of inverse of the subsampled Hessian and the stochastic gradient. However, as noted in the recent work [6], a subsampled Hessian is able to capture accurate information only about the high curvature directions. Due to inaccuracies in estimating the curvature in the low curvature directions, the authors in [6] proposed thresholding the low singular values to a higher value in order to stabilize the estimate, resulting in conservative moves in the estimated low-curvature directions. Specifically, if $\sigma_1, \sigma_2, \ldots, \sigma_p$ denote the singular values of the subsampled Hessian in descending order, the singular values $\sigma_{r+1}, \sigma_{r+2}, \ldots, \sigma_p$ are all set to σ_{r+1}, where r is chosen to be some small number. Other forms of shrinkage/thresholding can also be employed for the purpose of stabilizing the estimate, possibly using fewer computations, however we observe in this paper that this particular form has the desirable benefits that the inverse of the thresholded matrix can be obtained immediately and furthermore can be stored in only $O(rp)$ memory, by representing it in a compact manner (outer-product plus diagonal). This also means that the search direction can be obtained in $O(rp)$ time, thus the memory and computational requirements are comparable with stochastic first-order and quasi-Newton methods.

A sketch of the proof of convergence of the two approaches is presented in Section 3. Experimental results on real datasets are presented in Section 4.

2 Formal Setup and Algorithm

We assume that each function $\phi_i(\cdot)$ is L-smooth, while the overall objective function $\phi(\cdot)$ is γ-strongly convex, both with respect to the L_2 norm. The pseudo-code of the algorithm is provided in the box Algorithm 1. Note that the matrix $\hat{H}_z$ appearing in the zth epoch is not specified in the pseudo-code. We provide the following two options for $\hat{H}_z$:

- SVRG+I: Subsampled Hessian followed by singular value thresholding
  Sample a set $\mathbb{S}$ of S indices, and choose

$$\hat{H}_z^{-1} = Q\left(\Sigma_r^{-1} - \frac{1}{\sigma_{r+1}} I_r\right) Q' + \frac{1}{\sigma_{r+1}} I_p,$$

  where Σ_r is a diagonal matrix containing the top r singular values of the subsampled Hessian $\frac{1}{S}\sum_{i\in\mathbb{S}} \nabla^2 \phi_i(\widetilde{w}_s)$, the matrix $Q \in \mathbb{R}^{p\times r}$ contains its top r singular vectors.[2] This operation sets $\hat{H}_z$ as the subsampled Hessian whose singular values from σ_{r+2} to σ_p are set to be equal to σ_{r+1}. Due to the structure of $\hat{H}_z^{-1}$ (outer-product plus diagonal), the update step that involves computing $\hat{H}_z^{-1} v_t$ can be performed in $O(rp)$ time. The computations required for the subsampling and thresholding operation, performed once in each epoch, are $O(rp^2)$ using e.g. [7] (the randomized methods mentioned therein are parallelizable). Thus, thresholding has the double benefits of stabilizing the Hessian estimate and reducing storage, allowing the computation of the search direction in $O(rp)$ time, same as that of stochastic first-order and quasi-Newton methods.

- SVRG+II: LBFGS
  For this option, we define $\hat{H}_z^{-1}$ implicitly via the popular LBFGS method, i.e. at the beginning

[2] In this case, the left and right singular vectors are the same, since the subsampled Hessian is a real symmetric matrix.

of each epoch when the full gradient $\widetilde{\mu}_z$ is computed, we define

$$s_{z-1} \triangleq \widetilde{w}_z - \widetilde{w}_{z-1}, \qquad y_{z-1} \triangleq \widetilde{\mu}_z - \widetilde{\mu}_{z-1},$$

following which each update step in the epoch is computed using the two-loop procedure provided in [12, Algorithm 7.4].

We point out that the matrix $\hat{H}_z$ is fixed throughout an epoch.

Algorithm 1: SVRG + second-order information

Data: $\{\phi_i(\cdot)\}_{i=1}^n$, initial w_0, inner iterations m
Result: w
$w \leftarrow w_0$, $\widetilde{w}_1 \leftarrow w_0$
for $z = 1, 2, \ldots$ **do**
  $\widetilde{\mu}_z \leftarrow \frac{1}{n}\sum_{i=1}^n \nabla\phi_i(\widetilde{w}_z)$
  Choose $\hat{H}_z$
  for $t = 1$ **to** m **do**
    Sample i_t from $[1:n]$ uniformly at random
    $v_t \leftarrow \nabla\phi_{i_t}(w) - \nabla\phi_{i_t}(\widetilde{w}_z) + \widetilde{\mu}_z$
    $w \leftarrow w - \eta_z \hat{H}_z^{-1} v_t$
  end
  $\widetilde{w}_{z+1} \leftarrow w$
end

3 Convergence Results

Let $\hat{H}_z$ denote the scaling matrix used in epoch z and $\overline{w} = \hat{H}_z^{1/2} w$ denote the transformed parameter under the operator $\hat{H}_z$. Let ψ_i denote the function under this transformation, i.e., $\psi_i(\overline{w}) = \phi_i\big(\hat{H}_z^{-1/2}\overline{w}\big)$. Epoch z of Algorithm 1 effectively applies the vanilla SVRG to optimize the function $\frac{1}{n}\sum_{i=1}^n \psi_i(\overline{w})$. The reader is referred to [2, Section 9.4.1] for a discussion on this in the context of gradient descent. Combining the analysis in [8] with the coordinate transformation provides us with the following per-epoch bound

$$\mathbb{E}[\phi(\widetilde{w}_{z+1}) - \phi(w^*)] \leq \left(\frac{1}{\gamma_\psi \eta_z (1 - 2\eta_z L_\psi) m} + \frac{2\eta_z L_\psi}{1 - 2\eta_z L_\psi}\right) \mathbb{E}[\phi(\widetilde{w}_z) - \phi(w^*)],$$

where γ_ψ is the strong-convexity parameter of the function ψ and L_ψ is the largest of the smoothness parameters of the new functions ψ_i. So linear convergence can be established if one can prove an upper bound on L_ψ and a lower bound on γ_ψ, followed by choosing a step size of $\eta_z = O(1/L_\psi)$ and a sufficiently large $m = \Theta(L_\psi/\gamma_\psi)$, although the convergence rate obtained from such a bound can possibly be a very conservative estimate of the rate achievable in practice. For SVRG+II, the proof presented in [11] applies with little change. For SVRG+I, we have the following lemma, the proof of which employs the matrix Hoeffding inequality, and is skipped due to space constraints.

Lemma 1. We have $L_\psi \leq \frac{L}{\gamma - O\left(\sqrt{\frac{\log p}{S}}\right)}$ and $\gamma_\psi \geq \frac{\gamma}{L + O\left(\sqrt{\frac{\log p}{S}}\right)}$.

4 Experimental Results

The initial point is chosen to be all-zeros. We compare the performance of algorithms based on the log of suboptimality in function value vs number of passes through the data. For SVRG+I, the initial computations in each epoch required for the subsampling and rank-thresholding operations, normalized by the computations required for one pass through the data, are $O(rp^2/(np)) = O(rp/n)$, which is negligible in the $n \gg p$ regime. We also remark here that these initial computations in each epoch (computing full gradient, subsampled and thresholded Hessian) are heavily parallelizable. We aim to perform L_2-regularized logistic regression, with regularization $\lambda = 10^{-4}$:

$$\underset{w \in \mathbb{R}^p}{\text{minimize}} \left\{ \frac{1}{n}\sum_{i=1}^n \log(1 + e^{-b_i a_i^T w}) + \frac{\lambda}{2}\|w\|^2 \right\}.$$

We compare the two approaches described in the previous section (referred to as SVRG+I and SVRG+II respectively) with plain SVRG, stochastic-LBFGS method from [11] (denoted as SLBFGS) and batch-LBFGS. The first epoch of SVRG+II is run as an epoch of SVRG. The threshold for SVRG+I and the LBFGS memory parameter for SVRG+II, SLBFGS, batch-LBFGS are all set to be 20. The step-size for each stochastic method was chosen to be the one from the set $\{10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}, 10^{-6}\}$ that resulted in fastest and relatively stable convergence. Each epoch for SVRG, SVRG+I, SVRG+II and batch-LBFGS is counted as 2 passes over the data, while each epoch in SLBFGS (using parameter choices recommended in [11]) is counted as 3 passes.

We find in our experiments that the convergence rate of SVRG+I is not much higher than plain SVRG or SLBFGS. However, for all datasets, SVRG+II is by far the fastest in achieving a high accuracy solution. Though SVRG+II updates the Hessian approximation much less frequently than SLBFGS, the disadvantage of using an outdated approximation seems to be overcome significantly due to the use of exact gradient differences.

Dataset	n	p	Dataset	n	p	Dataset	n	p
mnist	60000	717	covtype	581012	54	Protein	145751	74

Table 1: Datasets used in experiments, obtained from [5] and the KDD cup 2004 website [1]

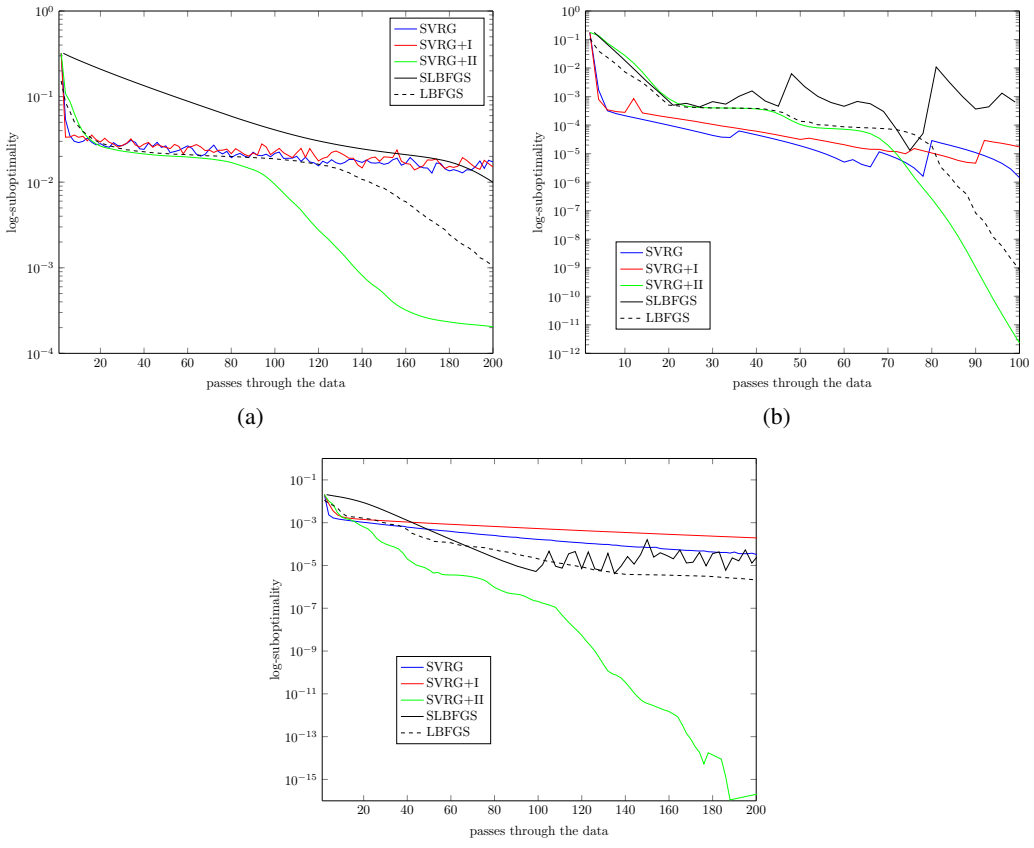

Figure 1: Log-suboptimality vs number of passes through the data for (a) mnist dataset, (b) covtype dataset, (c) Protein dataset

5 Extensions

Weighted sampling and optimization of minibatch size can be incorporated for further performance improvements. Extending the proposed approach to handle the presence of a non-smooth function via a proximal step should also be of interest. Finally, effects on test error remain to be investigated.

References

[1] Kdd cup, url = http://osmot.cs.cornell.edu/kddcup/. 2004.

[2] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, New York, NY, USA, 2004.

[3] 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*, 21(3):977–995, 2011.

[4] Richard H Byrd, SL Hansen, Jorge Nocedal, and Yoram Singer. A stochastic quasi-newton method for large-scale optimization. *arXiv preprint arXiv:1401.7020*, 2014.

[5] Chih-Chung Chang and Chih-Jen Lin. Libsvm: A library for support vector machines. *ACM Trans. Intell. Syst. Technol.*, 2(3):27:1–27:27, May 2011.

[6] M. A. Erdogdu and A. Montanari. Convergence rates of sub-sampled newton methods. *arXiv:1508.02810*, 2015.

[7] N. Halko, P. G. Martinsson, and J. A. Tropp. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM Rev.*, 53(2):217–288, May 2011.

[8] Rie Johnson and Tong Zhang. Accelerating stochastic gradient descent using predictive variance reduction. In *Advances in Neural Information Processing Systems 26*, pages 315–323. 2013.

[9] Qihang Lin, Zhaosong Lu, and Lin Xiao. An accelerated proximal coordinate gradient method. In *Advances in Neural Information Processing Systems 27*, pages 3059–3067. 2014.

[10] Aurélien Lucchi, Brian McWilliams, and Thomas Hofmann. A variance reduced stochastic newton method. *CoRR*, abs/1503.08316, 2015.

[11] P. Moritz, R. Nishihara, and M. I. Jordan. A linearly-convergent stochastic l-bfgs algorithm. *arXiv:1508.02087*, 2015.

[12] Jorge Nocedal and Stephen J. Wright. *Numerical Optimization, second edition*. World Scientific, 2006.

[13] Nicolas L. Roux, Mark Schmidt, and Francis R. Bach. A stochastic gradient method with an exponential convergence _rate for finite training sets. In *Advances in Neural Information Processing Systems 25*, pages 2663–2671. 2012.

[14] Shai Shalev-Shwartz and Tong Zhang. Accelerated proximal stochastic dual coordinate ascent for regularized loss minimization. *arXiv:1309.2375*, 2013.

[15] Shai Shalev-Shwartz and Tong Zhang. Stochastic dual coordinate ascent methods for regularized loss. *Journal of Machine Learning Research*, 14(1):567–599, February 2013.

[16] Jascha Sohl-Dickstein, Ben Poole, and Surya Ganguli. Fast large-scale optimization by unifying stochastic gradient and quasi-newton methods. *arXiv preprint arXiv:1311.2115*, 2013.

[17] Yuchen Zhang and Lin Xiao. Stochastic primal-dual coordinate method for regularized empirical risk minimization. *arXiv:1409.3257*, 2014.