Stochastic Recursive Variance-Reduced Cubic Regularization Methods

Dongruo Zhou
Department of Computer Science
University of California, Los Angeles

Quanquan Gu
Department of Computer Science
University of California, Los Angeles

Abstract

Stochastic Variance-Reduced Cubic regularization (SVRC) algorithms have received increasing attention due to its improved gradient/Hessian complexities (i.e., number of queries to stochastic gradient/Hessian oracles) to find local minima for nonconvex finite-sum optimization. However, it is unclear whether existing SVRC algorithms can be further improved. Moreover, the semi-stochastic Hessian estimator adopted in existing SVRC algorithms prevents the use of Hessian-vector product-based fast cubic subproblem solvers, which makes SVRC algorithms computationally intractable for high-dimensional problems. In this paper, we first present a Stochastic Recursive Variance-Reduced Cubic regularization method (SRVRC) using a recursively updated semi-stochastic gradient and Hessian estimators. It enjoys improved gradient and Hessian complexities to find an $(\epsilon, \sqrt{\epsilon})$-approximate local minimum, and outperforms the state-of-the-art SVRC algorithms. Built upon SRVRC, we further propose a Hessian-free SRVRC algorithm, namely SRVRC$_{\text{free}}$, which only needs $\widetilde{O}(n\epsilon^{-2} \wedge \epsilon^{-3})$ stochastic gradient and Hessian-vector product computations, where n is the number of component functions in the finite-sum objective and ϵ is the optimization precision. This outperforms the best-known result $\widetilde{O}(\epsilon^{-3.5})$ achieved by stochastic cubic regularization algorithm proposed in Tripuraneni et al. (2018).

Proceedings of the 23[rd]International Conference on Artificial Intelligence and Statistics (AISTATS) 2020, Palermo, Italy. PMLR: Volume 108.

1 Introduction

Many machine learning problems can be formulated as empirical risk minimization, which is in the form of finite-sum optimization as follows:

$$\min_{\mathbf{x}\in\mathbb{R}^d} F(\mathbf{x}) := n^{-1}\sum_{i=1}^n f_i(\mathbf{x}), \tag{1.1}$$

where each $f_i : \mathbb{R}^d \to \mathbb{R}$ can be a convex or nonconvex function. In this paper, we are particularly interested in nonconvex finite-sum optimization, where each f_i is nonconvex. This is often the case for deep learning (LeCun et al., 2015). In principle, it is hard to find the global minimum of (1.1) because of the NP-hardness of the problem (Hillar and Lim, 2013), thus it is reasonable to resort to finding local minima (a.k.a., second-order stationary points). It has been shown that local minima can be the global minima in certain machine learning problems, such as low-rank matrix factorization (Ge et al., 2016; Bhojanapalli et al., 2016; Zhang et al., 2018b) and training deep linear neural networks (Kawaguchi, 2016; Hardt and Ma, 2016). Therefore, developing algorithms to find local minima is important both in theory and in practice. More specifically, we define an (ϵ_g, ϵ_H)-approximate local minimum $\mathbf{x}$ of $F(\mathbf{x})$ as follows

$$\|\nabla F(\mathbf{x})\|_2 \le \epsilon_g, \quad \lambda_{\min}(\nabla^2 F(\mathbf{x})) \ge -\epsilon_H, \tag{1.2}$$

where $\epsilon_g, \epsilon_H > 0$ are predefined precision parameters. The most classic algorithm to find the approximate local minimum is cubic-regularized (CR) Newton method, which was originally proposed in the seminal paper by Nesterov and Polyak (2006). Generally speaking, in the k-th iteration, cubic regularization method solves a subproblem, which minimizes a cubic-regularized second-order Taylor expansion at the current iterate $\mathbf{x}_k$. The update rule can be written as follows:

$$\mathbf{h}_k = \operatorname*{argmin}_{\mathbf{h}\in\mathbb{R}^d} \langle \nabla F(\mathbf{x}_k), \mathbf{h}\rangle + 1/2\langle \nabla^2 F(\mathbf{x}_k)\mathbf{h}, \mathbf{h}\rangle + M/6\|\mathbf{h}\|_2^3, \tag{1.3}$$

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{h}_k, \tag{1.4}$$

where $M > 0$ is a penalty parameter. Nesterov and Polyak (2006) proved that to find an $(\epsilon, \sqrt{\epsilon})$-approximate local minimum of a nonconvex function F, cubic regularization requires at most $O(\epsilon^{-3/2})$ iterations. However, when applying cubic regularization to nonconvex finite-sum optimization in (1.1), a major bottleneck of cubic regularization is that it needs to compute n individual gradients $\nabla f_i(\mathbf{x}_k)$ and Hessian matrices $\nabla^2 f_i(\mathbf{x}_k)$ at each iteration, which leads to a total $O(n\epsilon^{-3/2})$ gradient complexity (i.e., number of queries to the stochastic gradient oracle $\nabla f_i(\mathbf{x})$ for some i and $\mathbf{x}$) and $O(n\epsilon^{-3/2})$ Hessian complexity (i.e., number of queries to the stochastic Hessian oracle $\nabla^2 f_i(\mathbf{x})$ for some i and $\mathbf{x}$). Such computational overhead will be extremely expensive when n is large as is in many large-scale machine learning applications.

To overcome the aforementioned computational burden of cubic regularization, Kohler and Lucchi (2017); Xu et al. (2017) used subsampled gradient and subsampled Hessian, which achieve $\widetilde{O}(n\epsilon^{-3/2} \wedge \epsilon^{-7/2})$ gradient complexity and $\widetilde{O}(n\epsilon^{-3/2} \wedge \epsilon^{-5/2})$ Hessian complexity. Zhou et al. (2018d) proposed a stochastic variance reduced cubic regularization method (SVRC), which uses novel semi-stochastic gradient and semi-stochastic Hessian estimators inspired by variance reduction for first-order finite-sum optimization (Johnson and Zhang, 2013; Reddi et al., 2016; Allen-Zhu and Hazan, 2016), which attains $O(n^{4/5}\epsilon^{-3/2})$ Second-order Oracle (SO) complexity[1]. Zhou et al. (2018b); Wang et al. (2018b); Zhang et al. (2018a) used a simpler semi-stochastic gradient compared with (Zhou et al., 2018d), and semi-stochastic Hessian, which achieves a better Hessian complexity, i.e., $O(n^{2/3}\epsilon^{-3/2})$. However, it is unclear whether the gradient and Hessian complexities of the aforementioned SVRC algorithms can be further improved. Furthermore, all these algorithms need to use the semi-stochastic Hessian estimator, which is not compatible with Hessian-vector product-based cubic subproblem solvers (Agarwal et al., 2017; Carmon and Duchi, 2016, 2018). Therefore, the cubic subproblem (1.4) in each iteration of existing SVRC algorithms has to be solved by computing the inverse of the Hessian matrix, whose computational complexity is at least $O(d^w)$[2]. This makes existing SVRC algorithms not very practical for high-dimensional problems.

In this paper, we first show that the gradient and Hessian complexities of SVRC-type algorithms can be further improved. The core idea is to use novel recursively updated semi-stochastic gradient and Hessian estimators, which are inspired by the stochastic path-integrated differential estimator (SPIDER) (Fang et al., 2018) and the StochAstic Recursive grAdient algoritHm (SARAH) (Nguyen et al., 2017) for first-order optimization. We show that such kind of estimators can be extended to second-order optimization to reduce the Hessian complexity. Nevertheless, our analysis is very different from that in Fang et al. (2018); Nguyen et al. (2017), because we study a fundamentally different optimization problem (i.e., finding local minima against finding first-order stationary points) and a completely different optimization algorithm (i.e., cubic regularization versus gradient method). In addition, in order to reduce the runtime complexity of existing SVRC algorithms, we further propose a *Hessian-free* SVRC method that can not only use the novel semi-stochastic gradient estimator, but also leverage the Hessian-vector product-based fast cubic subproblem solvers. Experiments on benchmark nonconvex finite-sum optimization problems illustrate the superiority of our newly proposed SVRC algorithms over the state-of-the-art (Due to space limit, we include the experiments in Appendix A).

In detail, our contributions are summarized as follows:

1. We propose a new SVRC algorithm, namely SRVRC, which can find an $(\epsilon, \sqrt{\epsilon})$-approximate local minimum with $\widetilde{O}(n\epsilon^{-3/2} \wedge \epsilon^{-3})$ gradient complexity and $\widetilde{O}(n \wedge \epsilon^{-1} + n^{1/2}\epsilon^{-3/2} \wedge \epsilon^{-2})$ Hessian complexity. Compared with previous work in cubic regularization, the gradient and Hessian complexity of SRVRC is strictly better than the algorithms in Zhou et al. (2018b); Wang et al. (2018b); Zhang et al. (2018a), and better than that in Zhou et al. (2018d); Shen et al. (2019) in a wide regime.

2. We further propose a new algorithm $\text{SRVRC}_{\text{free}}$, which requires $\widetilde{O}(\epsilon^{-3} \wedge n\epsilon^{-2})$ stochastic gradient and Hessian-vector product computations to find an $(\epsilon, \sqrt{\epsilon})$-approximate local minimum. $\text{SRVRC}_{\text{free}}$ is strictly better than the algorithms in (Agarwal et al., 2017; Carmon and Duchi, 2016; Tripuraneni et al., 2018) when $n \gg 1$. The runtime complexity of $\text{SRVRC}_{\text{free}}$ is also better than that of SRVRC when the problem dimension d is large.

In an independent and concurrent work (Shen et al., 2019), two stochastic trust region methods namely STR1 and STR2 were proposed, which are based on the same idea of variance reduction using SPIDER, and are related to our first algorithm SRVRC. Our SRVRC is better than STR1 because it enjoys the same Hessian complexity but a better gradient complexity than STR1. Compared with STR2, our SRVRC has a con-

[1] Second-order Oracle (SO) returns triple $[f_i(\mathbf{x}), \nabla f_i(\mathbf{x}), \nabla^2 f_i(\mathbf{x})]$ for some i and $\mathbf{x}$, hence the SO complexity can be seen as the maximum of gradient and Hessian complexities.

[2] w is the matrix multiplication constant, where $w = 2.37...$ (Golub and Van Loan, 1996).

sistently lower Hessian complexity and lower gradient complexity in a wide regime (i.e., $\epsilon \gg n^{-1/2}$). Since Hessian complexity is the dominating term in cubic regularization method (Zhou et al., 2018b; Wang et al., 2018b), our SRVRC is arguably better than STR2, as verified by our experiments.

For the ease of comparison, we summarize the comparison of methods which need to compute the Hessian explicitly in Table 1, the Hessian-free or Hessian-vector product based methods in Table 2.

2 Additional Related Work

In this section, we review additional related work that is not discussed in the introduction section.

Cubic Regularization and Trust-Region Methods Since cubic regularization was first proposed by Nesterov and Polyak (2006), there has been a line of followup research. It was extended to adaptive regularized cubic methods (ARC) by Cartis et al. (2011a,b), which enjoy the same iteration complexity as standard cubic regularization while having better empirical performance. The first attempt to make cubic regularization a Hessian-free method was done by Carmon and Duchi (2016), which solves the cubic sub-problem by gradient descent, requiring in total $\widetilde{O}(n\epsilon^{-2})$ stochastic gradient and Hessian-vector product computations. Agarwal et al. (2017) solved cubic sub-problem by fast matrix inversion based on accelerated gradient descent, which requires $\widetilde{O}(n\epsilon^{-3/2} + n^{3/4}\epsilon^{-7/4})$ stochastic gradient and Hessian-vector product computations. In the pure stochastic optimization setting, Tripuraneni et al. (2018) proposed stochastic cubic regularization method, which uses subsampled gradient and Hessian-vector product-based cubic subproblem solver, and requires $\widetilde{O}(\epsilon^{-3.5})$ stochastic gradient and Hessian-vector product computations. A closely related second-order method to cubic regularization methods are trust-region methods (Conn et al., 2000; Cartis et al., 2009, 2012, 2013). Recent studies (Blanchet et al., 2016; Curtis et al., 2017; Martínez and Raydan, 2017) proved that the trust-region method can achieve the same iteration complexity as the cubic regularization method. Xu et al. (2017) also extended trust-region method to subsampled trust-region method for nonconvex finite-sum optimization.

Local Minima Finding Besides cubic regularization and trust-region type methods, there is another line of research for finding approximate local minima, which is based on first-order optimization. Ge et al. (2015); Jin et al. (2017a) proved that (stochastic) gradient methods with additive noise are able to escape from nondegenerate saddle points and find approximate local minima. Carmon et al. (2018); Royer and Wright (2017); Allen-Zhu (2017); Xu et al. (2018); Allen-Zhu and Li (2018); Jin et al. (2017b); Yu et al. (2017, 2018); Zhou et al. (2018a); Fang et al. (2018) showed that by alternating first-order optimization and Hessian-vector product based negative curvature descent, one can find approximate local minima even more efficiently. Very recently, Fang et al. (2019); Jin et al. (2019) showed that stochastic gradient descent itself can escape from saddle points.

Variance Reduction Variance reduction techniques play an important role in our proposed algorithms. Variance reduction techniques were first proposed for convex finite-sum optimization, which use semi-stochastic gradient to reduce the variance of the stochastic gradient and improve the gradient complexity. Representative algorithms include Stochastic Average Gradient (SAG) (Roux et al., 2012), Stochastic Variance Reduced Gradient (SVRG) (Johnson and Zhang, 2013; Xiao and Zhang, 2014), SAGA (Defazio et al., 2014) and SARAH (Nguyen et al., 2017), to mention a few. For nonconvex finite-sum optimization problems, Garber and Hazan (2015); Shalev-Shwartz (2016) studied the case where each individual function is nonconvex, but their sum is still (strongly) convex. Reddi et al. (2016); Allen-Zhu and Hazan (2016) extended SVRG to noncovnex finite-sum optimization, which is able to converge to first-order stationary point with better gradient complexity than vanilla gradient descent. Fang et al. (2018); Zhou et al. (2018c); Wang et al. (2018a); Nguyen et al. (2019) further improved the gradient complexity for nonconvex finite-sum optimization to be (near) optimal.

3 Notation and Preliminaries

In this work, all index subsets are multiset. We use $\nabla f_{\mathcal{I}}(\mathbf{x})$ to represent $1/|\mathcal{I}| \cdot \sum_{i\in\mathcal{I}} \nabla f_i(\mathbf{x})$ if $|\mathcal{I}| < n$ and $\nabla F(\mathbf{x})$ otherwise. We use $\nabla^2 f_{\mathcal{I}}(\mathbf{x})$ to represent $1/|\mathcal{I}| \cdot \sum_{i\in\mathcal{I}} \nabla^2 f_i(\mathbf{x})$ if $|\mathcal{I}| < n$ and $\nabla^2 F(\mathbf{x})$ otherwise. For a vector $\mathbf{v}$, we denote its i-th coordinate by v_i. We denote vector Euclidean norm by $\|\mathbf{v}\|_2$. For any matrix $\mathbf{A}$, we denote its (i,j) entry by $A_{i,j}$, its Frobenius norm by $\|\mathbf{A}\|_F$, and its spectral norm by $\|\mathbf{H}\|_2$. For a symmetric matrix $\mathbf{H} \in \mathbb{R}^{d\times d}$, we denote its minimum eigenvalue by $\lambda_{\min}(\mathbf{H})$. For symmetric matrices $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{d\times d}$, we say $\mathbf{A} \succeq \mathbf{B}$ if $\lambda_{\min}(\mathbf{A} - \mathbf{B}) \geq 0$. We use $f_n = O(g_n)$ to denote that $f_n \leq Cg_n$ for some constant $C > 0$ and use $f_n = \widetilde{O}(g_n)$ to hide the logarithmic factors of g_n. We use $a \wedge b = \min\{a, b\}$.

We begin with a few assumptions that are needed for later theoretical analyses of our algorithms.

[3]The complexity for Natasha2 to find an $(\epsilon, \epsilon^{1/4})$-local minimum only requires $\widetilde{O}(\epsilon^{-3.25})$. Here we adapt the complexity result for finding an $(\epsilon, \epsilon^{1/2})$-approximate local minimum.

Table 1: Comparisons of different methods to find an $(\epsilon, \sqrt{\rho\epsilon})$-local minimum on gradient and Hessian complexity.

Algorithm	Gradient	Hessian
CR (Nesterov and Polyak, 2006)	$O\big(\frac{n}{\epsilon^{3/2}}\big)$	$O\big(\frac{n}{\epsilon^{3/2}}\big)$
SCR (Kohler and Lucchi, 2017; Xu et al., 2017)	$\widetilde{O}\big(\frac{n}{\epsilon^{3/2}} \wedge \frac{1}{\epsilon^{7/2}}\big)$	$\widetilde{O}\big(\frac{n}{\epsilon^{3/2}} \wedge \frac{1}{\epsilon^{5/2}}\big)$
SVRC (Zhou et al., 2018d)	$\widetilde{O}\big(\frac{n^{4/5}}{\epsilon^{3/2}}\big)$	$\widetilde{O}\big(\frac{n^{4/5}}{\epsilon^{3/2}}\big)$
(Lite-)SVRC (Zhou et al., 2018b; Wang et al., 2018b; Zhou et al., 2019)	$\widetilde{O}\big(\frac{n}{\epsilon^{3/2}}\big)$	$\widetilde{O}\big(\frac{n^{2/3}}{\epsilon^{3/2}}\big)$
SVRC (Zhang et al., 2018a)	$O\big(\frac{n}{\epsilon^{3/2}} \wedge \frac{n^{2/3}}{\epsilon^{5/2}}\big)$	$O\big(\frac{n^{2/3}}{\epsilon^{3/2}}\big)$
STR1 (Shen et al., 2019)	$\widetilde{O}\big(\frac{n}{\epsilon^{3/2}} \wedge \frac{n^{1/2}}{\epsilon^{2}}\big)$	$\widetilde{O}\big(\frac{n^{1/2}}{\epsilon^{3/2}} \wedge \frac{1}{\epsilon^{2}}\big)$
STR2 (Shen et al., 2019)	$\widetilde{O}\big(\frac{n^{3/4}}{\epsilon^{3/2}}\big)$	$\widetilde{O}\big(\frac{n^{3/4}}{\epsilon^{3/2}}\big)$
SRVRC (This work)	$\widetilde{O}\big(\frac{n}{\epsilon^{3/2}} \wedge \frac{n^{1/2}}{\epsilon^{2}} \wedge \frac{1}{\epsilon^{3}}\big)$	$\widetilde{O}\big(\frac{n^{1/2}}{\epsilon^{3/2}} \wedge \frac{1}{\epsilon^{2}}\big)$

Table 2: Comparisons of different methods to find an $(\epsilon, \sqrt{\rho\epsilon})$-local minimum both on stochastic gradient and Hessian-vector product computations.

Algorithm	Gradient & Hessian-vector product
SGD (Fang et al., 2019)	$\widetilde{O}\big(\frac{1}{\epsilon^{7/2}}\big)$
SGD (Jin et al., 2019)	$\widetilde{O}\big(\frac{1}{\epsilon^{4}}\big)$
Fast-Cubic (Agarwal et al., 2017)	$\widetilde{O}\big(\frac{n}{\epsilon^{3/2}} + \frac{n^{3/4}}{\epsilon^{7/4}}\big)$
GradientCubic (Carmon and Duchi, 2016)	$\widetilde{O}\big(\frac{n}{\epsilon^{2}}\big)$
STC (Tripuraneni et al., 2018)	$\widetilde{O}\big(\frac{1}{\epsilon^{7/2}}\big)$
SPIDER (Fang et al., 2018)	$\widetilde{O}\big((\frac{\sqrt{n}}{\epsilon^{2}} + \frac{1}{\epsilon^{2.5}}) \wedge \frac{1}{\epsilon^{3}}\big)$
$\text{SRVRC}_{\text{free}}$ (This work)	$\widetilde{O}\big(\frac{n}{\epsilon^{2}} \wedge \frac{1}{\epsilon^{3}}\big)$

The following assumption says that the gap between the function value at the initial point $\mathbf{x}_0$ and the minimal function value is bounded.

Assumption 3.1. For any function $F(\mathbf{x})$ and an initial point $\mathbf{x}_0$, there exists a constant $0 < \Delta_F < \infty$ such that $F(\mathbf{x}_0) - \inf_{\mathbf{x}\in\mathbb{R}^d} F(\mathbf{x}) \leq \Delta_F$.

We also need the following L-gradient Lipschitz and ρ-Hessian Lipschitz assumption.

Assumption 3.2. For each i, we assume that f_i is L-gradient Lipschitz continuous and ρ-Hessian Lipschitz continuous, where we have $\|\nabla f_i(\mathbf{x}) - \nabla f_i(\mathbf{y})\|_2 \leq L\|\mathbf{x}-\mathbf{y}\|_2$ and $\|\nabla^2 f_i(\mathbf{x}) - \nabla^2 f_i(\mathbf{y})\|_2 \leq \rho\|\mathbf{x}-\mathbf{y}\|_2$ for all $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$.

Note that L-gradient Lipschitz is not required in the original cubic regularization algorithm (Nesterov and Polyak, 2006) and the SVRC algorithm (Zhou et al., 2018d). However, for most other SVRC algorithms (Zhou et al., 2018b; Wang et al., 2018b; Zhang et al., 2018a), they need the L-gradient Lipschitz assumption.

In addition, we need the difference between the stochastic gradient and the full gradient to be bounded.

Assumption 3.3. We assume that F has M-bounded stochastic gradient, where we have $\|\nabla f_i(\mathbf{x}) - \nabla F(\mathbf{x})\|_2 \leq M, \forall \mathbf{x} \in \mathbb{R}^d, \forall i \in [n]$.

It is worth noting that Assumption 3.3 is weaker than the assumption that each f_i is Lipschitz continuous, which has been made in Kohler and Lucchi (2017); Zhou et al. (2018b); Wang et al. (2018b); Zhang et al. (2018a). We would also like to point out that we can make additional assumptions on the variances of the stochastic gradient and Hessian, such as the ones made in Tripuraneni et al. (2018). Nevertheless, making these additional assumptions does not improve the dependency of the gradient and Hessian complexities or the stochastic gradient and Hessian-vector product computations on ϵ and n. Therefore we chose not making these additional assumptions on the variances.

4 The Proposed SRVRC Algorithm

In this section, we present SRVRC, a novel algorithm which utilizes new semi-stochastic gradient and Hessian estimators compared with previous SVRC algorithms. We also provide a convergence analysis of the proposed

Algorithm 1 Stochastic Recursive Variance-Reduced Cubic Regularization (SRVRC)

Input: Total iterations T, batch sizes $\{B_t^{(g)}\}_{t=1}^T, \{B_t^{(h)}\}_{t=1}^T$, cubic penalty parameter $\{M_t\}_{t=1}^T$, inner gradient length $S^{(g)}$, inner Hessian length $S^{(h)}$, initial point $\mathbf{x}_0$, accuracy ϵ and Hessian Lipschitz constant ρ.
for $t = 0, \ldots, T-1$ **do**
  Sample index set $\mathcal{J}_t$ with $|\mathcal{J}_t| = B_t^{(g)}$; $\mathcal{I}_t$ with $|\mathcal{I}_t| = B_t^{(h)}$;

$$\mathbf{v}_t \leftarrow \begin{cases} \nabla f_{\mathcal{J}_t}(\mathbf{x}_t), & \mathrm{mod}(t, S^{(g)}) = 0 \\ \nabla f_{\mathcal{J}_t}(\mathbf{x}_t) - \nabla f_{\mathcal{J}_t}(\mathbf{x}_{t-1}) + \mathbf{v}_{t-1}, & \text{else} \end{cases} \tag{3.1}$$

$$\mathbf{U}_t \leftarrow \begin{cases} \nabla^2 f_{\mathcal{I}_t}(\mathbf{x}_t), & \mathrm{mod}(t, S^{(h)}) = 0 \\ \nabla^2 f_{\mathcal{I}_t}(\mathbf{x}_t) - \nabla^2 f_{\mathcal{I}_t}(\mathbf{x}_{t-1}) + \mathbf{U}_{t-1}, & \text{else} \end{cases} \tag{3.2}$$

$$\mathbf{h}_t \leftarrow \operatorname*{argmin}_{\mathbf{h}\in\mathbb{R}^d} m_t(\mathbf{h}) := \langle \mathbf{v}_t, \mathbf{h}\rangle + \frac{1}{2}\langle \mathbf{U}_t\mathbf{h}, \mathbf{h}\rangle + \frac{M_t}{6}\|\mathbf{h}\|_2^3 \tag{3.3}$$

  $\mathbf{x}_{t+1} \leftarrow \mathbf{x}_t + \mathbf{h}_t$
  if $\|\mathbf{h}_t\|_2 \le \sqrt{\epsilon/\rho}$ **then**
    return $\mathbf{x}_{t+1}$
  end if
end for

algorithm.

4.1 Algorithm Description

In order to reduce the computational complexity for calculating full gradient and full Hessian in (1.3), several ideas such as subsampled/stochastic gradient and Hessian (Kohler and Lucchi, 2017; Xu et al., 2017; Tripuraneni et al., 2018) and variance-reduced semi-stochastic gradient and Hessian (Zhou et al., 2018d; Wang et al., 2018b; Zhang et al., 2018a) have been used in previous work. SRVRC follows this line of work. The key idea is to use a new construction of semi-stochastic gradient and Hessian estimators, which are recursively updated in each iteration, and reset periodically after certain number of iterations (i.e., an epoch). This is inspired by the first-order variance reduction algorithms SPIDER (Fang et al., 2018) and SARAH (Nguyen et al., 2017). SRVRC constructs semi-stochastic gradient and Hessian as in (3.1) and (3.2) respectively. To be more specific, in the t-th iteration when $\mathrm{mod}(t, S^{(g)}) = 0$ or $\mathrm{mod}(t, S^{(h)}) = 0$, where $S^{(g)}, S^{(h)}$ are the epoch lengths of gradient and Hessian, SRVRC will set the semi-stochastic gradient $\mathbf{v}_t$ and Hessian $\mathbf{U}_t$ to be a subsampled gradient $\nabla f_{\mathcal{J}_t}(\mathbf{x}_t)$ and Hessian $\nabla^2 f_{\mathcal{J}_t}(\mathbf{x}_t)$ at point $\mathbf{x}_t$, respectively. In the t-th iteration when $\mathrm{mod}(t, S) \neq 0$ or $\mathrm{mod}(t, S^{(h)}) \neq 0$, SRVRC constructs semi-stochastic gradient and Hessian $\mathbf{v}_t$ and $\mathbf{U}_t$ based on previous estimators $\mathbf{v}_{t-1}$ and $\mathbf{U}_{t-1}$ recursively. With semi-stochastic gradient $\mathbf{v}_t$, semi-stochastic Hessian $\mathbf{U}_t$ and t-th Cubic penalty parameter M_t, SRVRC constructs the t-th Cubic subproblem m_t and solves for the solution to m_t as t-th update direction as (3.3). If $\|\mathbf{h}_t\|_2$ is less than a given threshold which we set it as $\sqrt{\epsilon/\rho}$, SRVRC returns $\mathbf{x}_{t+1} = \mathbf{x}_t + \mathbf{h}_t$ as its output. Otherwise, SRVRC updates $\mathbf{x}_{t+1} = \mathbf{x}_t + \mathbf{h}_t$ and continues the loop.

The main difference between SRVRC and previous stochastic cubic regularization algorithms (Kohler and Lucchi, 2017; Xu et al., 2017; Zhou et al., 2018d,b; Wang et al., 2018b; Zhang et al., 2018a) is that SRVRC adapts new semi-stochastic gradient and semi-stochastic Hessian estimators, which are defined recursively and have smaller asymptotic variance. The use of such semi-stochastic gradient has been proved to help reduce the gradient complexity in first-order nonconvex finite-sum optimization for finding stationary points (Fang et al., 2018; Wang et al., 2018a; Nguyen et al., 2019). Our work takes one step further to apply it to Hessian, and we will later show that it helps reduce the gradient and Hessian complexities in second-order nonconvex finite-sum optimization for finding local minima.

4.2 Convergence Analysis

In this subsection, we present our theoretical results about SRVRC. While the idea of using variance reduction technique for cubic regularization is hardly new, the new semi-stochastic gradient and Hessian estimators in (3.1) and (3.2) bring new technical challenges in the convergence analysis.

To describe whether a point $\mathbf{x}$ is a local minimum, we follow the original cubic regularization work (Nesterov and Polyak, 2006) to use the following criterion $\mu(\mathbf{x})$:

Definition 4.1. For any $\mathbf{x}$, define $\mu(\mathbf{x})$ as $\mu(\mathbf{x}) = \max\{\|\nabla F(\mathbf{x})\|_2^{3/2}, -\lambda_{\min}^3\big(\nabla^2 F(\mathbf{x})\big)/\rho^{3/2}\}$.

It is easy to note that $\mu(\mathbf{x}) \leq \epsilon^{3/2}$ if and only if $\mathbf{x}$ is an $(\epsilon, \sqrt{\rho\epsilon})$-approximate local minimum. Thus, in order to find an $(\epsilon, \sqrt{\rho\epsilon})$-approximate local minimum, it suffices to find a point $\mathbf{x}$ which satisfies $\mu(\mathbf{x}) \leq \epsilon^{3/2}$.

The following theorem provides the convergence guarantee of SRVRC for finding an $(\epsilon, \sqrt{\rho\epsilon})$-approximate local minimum.

Theorem 4.2. Under Assumptions 3.1, 3.2 and 3.3, set the cubic penalty parameter $M_t = 4\rho$ for any t and the total iteration number $T \geq 40\Delta_F \rho^{1/2} \epsilon^{-3/2}$. For t such that $\mathrm{mod}(t, S^{(g)}) \neq 0$ or $\mathrm{mod}(t, S^{(h)}) \neq 0$, set the gradient sample size $B_t^{(g)}$ and Hessian sample size $B_t^{(h)}$ as

$$B_t^{(g)} \geq n \wedge \frac{1440 L^2 S^{(g)} \|\mathbf{h}_{t-1}\|_2^2 \log^2(2T/\xi)}{\epsilon^2}, \tag{4.1}$$

$$B_t^{(h)} \geq n \wedge \frac{800 \rho S^{(h)} \|\mathbf{h}_{t-1}\|_2^2 \log^2(2Td/\xi)}{\epsilon}. \tag{4.2}$$

For t such that $\mathrm{mod}(t, S^{(g)}) = 0$ or $\mathrm{mod}(t, S^{(h)}) = 0$, set the gradient sample size $B_t^{(g)}$ and Hessian sample size $B_t^{(h)}$ as

$$B_t^{(g)} \geq n \wedge \frac{1440 M^2 \log^2(2T/\xi)}{\epsilon^2}, \tag{4.3}$$

$$B_t^{(h)} \geq n \wedge \frac{800 L^2 \log^2(2Td/\xi)}{\rho\epsilon}. \tag{4.4}$$

Then with probability at least $1 - \xi$, SRVRC outputs $\mathbf{x}_{\text{out}}$ satisfying $\mu(\mathbf{x}_{\text{out}}) \leq 600\epsilon^{3/2}$, i.e., an $(\epsilon, \sqrt{\rho\epsilon})$-approximate local minimum.

Next corollary spells out the exact gradient complexity and Hessian complexity of SRVRC to find an $(\epsilon, \sqrt{\rho\epsilon})$-approximate local minimum.

Corollary 4.3. Under the same conditions as Theorem 4.2, if we set $S^{(g)}, S^{(h)}$ as $S^{(g)} = \sqrt{\rho\epsilon}/L \cdot \sqrt{n \wedge M^2/\epsilon^2}$ and $S^{(h)} = \sqrt{n \wedge L/(\rho\epsilon)}$, and set $T, \{B_t^{(g)}\}, \{B_t^{(h)}\}$ as their lower bounds in (4.1)- (4.4), then with probability at least $1-\xi$, SRVRC will output an $(\epsilon, \sqrt{\rho\epsilon})$-approximate local minimum within

$$\widetilde{O}\bigg(n \wedge \frac{L^2}{\rho\epsilon} + \frac{\sqrt{\rho}\Delta_F}{\epsilon^{3/2}} \sqrt{n \wedge \frac{L^2}{\rho\epsilon}}\bigg)$$

stochastic Hessian evaluations and

$$\widetilde{O}\bigg(n \wedge \frac{M^2}{\epsilon^2} + \frac{\Delta_F}{\epsilon^{3/2}} \bigg[\sqrt{\rho} n \wedge \frac{L\sqrt{n}}{\sqrt{\epsilon}} \wedge \frac{LM}{\epsilon^{3/2}}\bigg]\bigg)$$

stochastic gradient evaluations.

Remark 4.4. For SRVRC, if we assume M, L, ρ, Δ_F to be constants, then its gradient complexity is $\widetilde{O}(n/\epsilon^{3/2} \wedge \sqrt{n}/\epsilon^2 \wedge \epsilon^{-3})$, and its Hessian complexity is $\widetilde{O}(n \wedge \epsilon^{-1} + n^{1/2}\epsilon^{-3/2} \wedge \epsilon^{-2})$. Regarding Hessian complexity, suppose that $\epsilon \ll 1$, then the Hessian complexity of SRVRC can be simplified as $\widetilde{O}(n^{1/2}\epsilon^{-3/2} \wedge \epsilon^{-2})$. Compared with existing SVRC algorithms (Zhou et al., 2018b; Zhang et al., 2018a; Wang et al., 2018b), SRVRC outperforms the best-known Hessian sample complexity by a factor of $n^{1/6} \wedge n^{2/3}\epsilon^{1/2}$. In terms of gradient complexity, SRVRC outperforms STR2 (Shen et al., 2019) by a factor of $n^{3/4}\epsilon^{3/2}$ when $\epsilon \gg n^{-1/2}$.

Remark 4.5. Note that both Theorem 4.2 and Corollary 4.3 still hold when Assumption 3.3 does not hold. In that case, $M = \infty$ and SRVRC's Hessian complexity remains the same, while its gradient complexity can be potentially worse, i.e., $\widetilde{O}(n/\epsilon^{3/2} \wedge \sqrt{n}/\epsilon^2)$, which degenerates to that of STR1 (Shen et al., 2019).

5 Hessian-Free SRVRC

While SRVRC adapts novel semi-stochastic gradient and Hessian estimators to reduce both the gradient and Hessian complexities, it has three limitations for high-dimensional problems with $d \gg 1$: (1) it needs to compute and store the Hessian matrix, which needs $O(d^2)$ computational time and storage space; (2) it needs to solve cubic subproblem m_t exactly, which requires $O(d^w)$ computational time because it needs to compute the inverse of a Hessian matrix (Nesterov and Polyak, 2006); and (3) it cannot leverage the Hessian-vector product-based cubic subproblem solvers (Agarwal et al., 2017; Carmon and Duchi, 2016, 2018) because of the use of the semi-stochastic Hessian estimator. It is interesting to ask whether we can modify SRVRC to overcome these shortcomings.

5.1 Algorithm Description

We present a Hessian-free algorithm SRVRC$_{\text{free}}$ to address above limitations of SRVRC for high-dimensional problems, which only requires stochastic gradient and Hessian-vector product computations. SRVRC$_{\text{free}}$ uses the same semi-stochastic gradient $\mathbf{v}_t$ as SRVRC. As opposed to SRVRC which has to construct semi-stochastic Hessian explicitly, SRVRC$_{\text{free}}$ only accesses to stochastic Hessian-vector product. In detail, at each iteration t, SRVRC$_{\text{free}}$ subsamples an index set $\mathcal{I}_t$ and defines a stochastic Hessian-vector product function $\mathbf{U}_t[\cdot] : \mathbb{R}^d \to \mathbb{R}^d$ as follows:

$$\mathbf{U}_t[\mathbf{v}] = \nabla^2 f_{\mathcal{I}_t}(\mathbf{x}_t)[\mathbf{v}], \qquad \forall \mathbf{v} \in \mathbb{R}^d.$$

Note that although the subproblem depends on $\mathbf{U}_t$, SRVRC$_{\text{free}}$ never explicitly computes this matrix. Instead, it only provides the subproblem solver access to $\mathbf{U}_t$ through stochastic Hessian-vector product function $\mathbf{U}_t[\cdot]$. The subproblem solver performs gradient-based optimization to solve the subproblem $m_t(\mathbf{h})$ as $\nabla m_t(\mathbf{h})$ depends on $\mathbf{U}_t$ only via $\mathbf{U}_t[\mathbf{h}]$. In detail,

Algorithm 2 Hessian Free Stochastic Recursive Variance-Reduced Cubic Regularization ($\text{SRVRC}_{\text{free}}$)

Input: Total iterations T, batch sizes $\{B_t^{(g)}\}_{t=1}^T, \{B_t^{(h)}\}_{t=1}^T$, cubic penalty parameter $\{M_t\}_{t=1}^T$, inner gradient length $S^{(g)}$, initial point $\mathbf{x}_0$, accuracy ϵ, Hessian Lipschitz constant ρ, gradient Lipschitz constant L and failure probability ξ.
for $t = 0, \ldots, T-1$ **do**
  Sample index set $\mathcal{J}_t, |\mathcal{J}_t| = B_t^{(g)}$; $\mathcal{I}_t, |\mathcal{I}_t| = B_t^{(h)}$;

$$\mathbf{v}_t \leftarrow \begin{cases} \nabla f_{\mathcal{J}_t}(\mathbf{x}_t), & \text{mod}(t, S^{(g)}) = 0 \\ \nabla f_{\mathcal{J}_t}(\mathbf{x}_t) - \nabla f_{\mathcal{J}_t}(\mathbf{x}_{t-1}) + \mathbf{v}_{t-1}, & \text{else} \end{cases}, \mathbf{U}_t[\cdot] \leftarrow \nabla^2 f_{\mathcal{I}_t}(\mathbf{x}_t)[\cdot]$$

  $\mathbf{h}_t \leftarrow$ Cubic-Subsolver$(\mathbf{U}_t[\cdot], \mathbf{v}_t, M_t, 1/(16L), \sqrt{\epsilon/\rho}, 0.5, \xi/(3T))$ {See Algorithm 3 in Appendix H}
  if $m_t(\mathbf{h}_t) < -4\rho^{-1/2}\epsilon^{3/2}$ **then**
    $\mathbf{x}_{t+1} \leftarrow \mathbf{x}_t + \mathbf{h}_t$
  else
    $\mathbf{h}_t \leftarrow$ Cubic-Finalsolver$(\mathbf{U}_t[\cdot], \mathbf{v}_t, M_t, 1/(16L), \epsilon)$ {See Algorithm 4 in Appendix H}
    return $\mathbf{x}_{t+1} \leftarrow \mathbf{x}_t + \mathbf{h}_t$
  end if
end for

following Tripuraneni et al. (2018), $\text{SRVRC}_{\text{free}}$ uses Cubic-Subsolver (See Algorithms 3 and 4 in Appendix H) and Cubic-Finalsolver from (Carmon and Duchi, 2016), to find an approximate solution $\mathbf{h}_t$ to the cubic subproblem in (3.3). Both Cubic-Subsolver and Cubic-Finalsolver only need to access gradient $\mathbf{v}_t$ and Hessian-vector product function $\mathbf{U}_t[\cdot]$ along with other problem-dependent parameters. With the output $\mathbf{h}_t$ from Cubic-Subsolver, $\text{SRVRC}_{\text{free}}$ decides either to update $\mathbf{x}_t$ as $\mathbf{x}_{t+1} \leftarrow \mathbf{x}_t + \mathbf{h}_t$ or to exit the loop. For the later case, $\text{SRVRC}_{\text{free}}$ will call Cubic-Finalsolver to output $\mathbf{h}_t$, and takes $\mathbf{x}_{t+1} = \mathbf{x}_t + \mathbf{h}_t$ as its final output.

The main differences between SRVRC and $\text{SRVRC}_{\text{free}}$ are two-fold. First, $\text{SRVRC}_{\text{free}}$ only needs to compute stochastic gradient and Hessian-vector product. Since both of these two computations only take $O(d)$ time in many applications in machine learning, $\text{SRVRC}_{\text{free}}$ is suitable for high-dimensional problems. In the sequel, following Agarwal et al. (2017); Carmon et al. (2018); Tripuraneni et al. (2018), we do not distinguish stochastic gradient and Hessian-vector product computations and consider them to have the same runtime complexity. Second, instead of solving cubic subproblem m_t exactly, $\text{SRVRC}_{\text{free}}$ adopts approximate subproblem solver Cubic-Subsolver and Cubic-Finalsolver, both of which only need to access gradient and Hessian-vector product function, and again only take $O(d)$ time. Thus, $\text{SRVRC}_{\text{free}}$ is computational more efficient than SRVRC when $d \gg 1$.

5.2 Convergence Analysis

We now provide the convergence guarantee of $\text{SRVRC}_{\text{free}}$, which ensures that $\text{SRVRC}_{\text{free}}$ will output an $(\epsilon, \sqrt{\rho\epsilon})$-approximate local minimum.

Theorem 5.1. Under Assumptions 3.1, 3.2, 3.3, suppose $\epsilon < L/(4\rho)$. Set the cubic penalty parameter $M_t = 4\rho$ for any t and the total iteration number $T \geq 25\Delta_F \rho^{1/2}\epsilon^{-3/2}$. Set the Hessian-vector product sample size $B_t^{(h)}$ as

$$B_t^{(h)} \geq n \wedge \frac{1200L^2 \log^2(3Td/\xi)}{\rho\epsilon}. \tag{5.1}$$

For t such that $\text{mod}(t, S^{(g)}) \neq 0$, set the gradient sample size $B_t^{(g)}$ as

$$B_t^{(g)} \geq n \wedge \frac{2640L^2 S^{(g)} \|\mathbf{h}_{t-1}\|_2^2 \log^2(3T/\xi)}{\epsilon^2}. \tag{5.2}$$

For t such that $\text{mod}(t, S^{(g)}) = 0$, set the gradient sample size $B_t^{(g)}$ as

$$B_t^{(g)} \geq n \wedge \frac{2640M^2 \log^2(3T/\xi)}{\epsilon^2}. \tag{5.3}$$

Then with probability at least $1-\xi$, $\text{SRVRC}_{\text{free}}$ outputs $\mathbf{x}_{\text{out}}$ satisfying $\mu(\mathbf{x}_{\text{out}}) \leq 1300\epsilon^{3/2}$, i.e., an $(\epsilon, \sqrt{\rho\epsilon})$-approximate local minimum.

The following corollary calculates the total amount of stochastic gradient and Hessian-vector product computations of $\text{SRVRC}_{\text{free}}$ to find an $(\epsilon, \sqrt{\rho\epsilon})$-approximate local minimum.

Corollary 5.2. Under the same conditions as Theorem 5.1, if set $S^{(g)} = \sqrt{\rho\epsilon}/L \cdot \sqrt{n \wedge M^2/\epsilon^2}$ and set $T, \{B_t^{(g)}\}, \{B_t^{(h)}\}$ as their lower bounds in (5.1)-(5.3), then with probability at least $1-\xi$, $\text{SRVRC}_{\text{free}}$ will

output an $(\epsilon, \sqrt{\rho\epsilon})$-approximate local minimum within

$$\widetilde{O}\bigg[\bigg(n \wedge \frac{M^2}{\epsilon^2}\bigg) + \frac{\Delta_F}{\epsilon^{3/2}}\bigg(\sqrt{\rho} n \wedge \frac{L\sqrt{n}}{\sqrt{\epsilon}} \wedge \frac{LM}{\epsilon^{3/2}}\bigg) + \bigg(\frac{L\Delta_F}{\epsilon^2} + \frac{L}{\sqrt{\rho\epsilon}}\bigg) \cdot \bigg(n \wedge \frac{L^2}{\rho\epsilon}\bigg)\bigg] \tag{5.4}$$

stochastic gradient and Hessian-vector product computations.

Remark 5.3. For SRVRC$_{\text{free}}$, if we assume ρ, L, M, Δ_F are constants, then (5.4) is $\widetilde{O}(n\epsilon^{-2} \wedge \epsilon^{-3})$. For stochastic algorithms, the regime $n \to \infty$ is of most interest. In this regime, (5.4) becomes $\widetilde{O}(\epsilon^{-3})$. Compared with other local minimum finding algorithms based on stochastic gradient and Hessian-vector product, SRVRC$_{\text{free}}$ outperforms the results achieved by Tripuraneni et al. (2018) and Allen-Zhu (2018) by a factor of $\epsilon^{-1/2}$. SRVRC$_{\text{free}}$ also matches the best-known result achieved by a recent first-order algorithm proposed in (Fang et al., 2018). Note that the algorithm proposed by Fang et al. (2018) needs to alternate the first-order finite-sum optimization algorithm SPIDER and negative curvature descent. In sharp contrast, SRVRC$_{\text{free}}$ is a pure cubic regularization type algorithm and does not need to calculate the negative curvature direction.

Remark 5.4. It is worth noting that both Theorem 5.1 and Corollary 5.2 still hold when Assumption 3.3 does not hold, and SRVRC$_{\text{free}}$'s runtime complexity remains the same. The only difference is: without Assumption 3.3, we need to use full gradient (i.e., $B_t^{(g)} = n$) instead of subsampled gradient at each iteration t.

5.3 Discussions on runtime complexity

We would like to further compare the runtime complexity between SRVRC and SRVRC$_{\text{free}}$. In specific, SRVRC needs $O(d)$ time to construct semi-stochastic gradient and $O(d^2)$ time to construct semi-stochastic Hessian. SRVRC also needs $O(d^w)$ time to solve cubic subproblem m_t for each iteration. Thus, with the fact that the total number of iterations is $T = O(\epsilon^{-3/2})$ by Corollary 4.3, SRVRC needs

$$\widetilde{O}\bigg(d\Big[\frac{n}{\epsilon^{3/2}} \wedge \frac{1}{\epsilon^3}\Big] + d^2\Big[n \wedge \frac{1}{\epsilon} + \frac{\sqrt{n}}{\epsilon^{3/2}} \wedge \frac{1}{\epsilon^2}\Big] + \frac{d^w}{\epsilon^{3/2}}\bigg)$$

runtime to find an $(\epsilon, \sqrt{\epsilon})$-approximate local minimum if we regard M, L, ρ, Δ_F as constants. As we mentioned before, for many machine learning problems, both stochastic gradient and Hessian-vector product computations only need $O(d)$ time, therefore the runtime of SRVRC$_{\text{free}}$ is $\widetilde{O}(dn\epsilon^{-2} \wedge d\epsilon^{-3})$. We conclude that SRVRC$_{\text{free}}$ outperforms SRVRC when d is large, which is in accordance with the fact that Hessian-free methods are superior for high dimension machine learning tasks. On the other hand, a careful calculation can show that the runtime of SRVRC can be less than that of SRVRC$_{\text{free}}$ when d is moderately small. This is also reflected in our experiments in Section A.

6 Conclusions and Future Work

In this work we present two faster SVRC algorithms namely SRVRC and SRVRC$_{\text{free}}$ to find approximate local minima for nonconvex finite-sum optimization problems. SRVRC outperforms existing SVRC algorithms in terms of gradient and Hessian complexities, while SRVRC$_{\text{free}}$ further outperforms the best-known runtime complexity for existing CR based algorithms. Whether our algorithms have achieved the optimal complexity under the current assumptions is still an open problem, and we leave it as a future work.

Acknowledgement

We would like to thank the anonymous reviewers for their helpful comments. This research was sponsored in part by the National Science Foundation IIS-1906169. The views and conclusions contained in this paper are those of the authors and should not be interpreted as representing any funding agencies.

References

Agarwal, N., Allen-Zhu, Z., Bullins, B., Hazan, E., and Ma, T. (2017). Finding approximate local minima faster than gradient descent. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, pages 1195–1199.

Allen-Zhu, Z. (2017). Natasha 2: Faster non-convex optimization than sgd. *arXiv preprint arXiv:1708.08694*.

Allen-Zhu, Z. (2018). Natasha 2: Faster non-convex optimization than sgd. In *Advances in Neural Information Processing Systems*, pages 2680–2691.

Allen-Zhu, Z. and Hazan, E. (2016). Variance reduction for faster non-convex optimization. In *International Conference on Machine Learning*, pages 699–707.

Allen-Zhu, Z. and Li, Y. (2018). Neon2: Finding local minima via first-order oracles. In *Advances in Neural Information Processing Systems*, pages 3720–3730.

Bhojanapalli, S., Neyshabur, B., and Srebro, N. (2016). Global optimality of local search for low rank matrix recovery. In *Advances in Neural Information Processing Systems*, pages 3873–3881.

Blanchet, J., Cartis, C., Menickelly, M., and Scheinberg, K. (2016). Convergence rate analysis of a stochastic trust region method for nonconvex optimization. *arXiv preprint arXiv:1609.07428*.

Carmon, Y. and Duchi, J. C. (2016). Gradient descent efficiently finds the cubic-regularized non-convex newton step. *arXiv preprint arXiv:1612.00547*.

Carmon, Y. and Duchi, J. C. (2018). Analysis of krylov subspace solutions of regularized non-convex quadratic problems. In *Advances in Neural Information Processing Systems*, pages 2680–2691.

Carmon, Y., Duchi, J. C., Hinder, O., and Sidford, A. (2018). Accelerated methods for nonconvex optimization. *SIAM Journal on Optimization*, 28(2):1751–1772.

Cartis, C., Gould, N. I., and Toint, P. L. (2009). Trust-region and other regularisations of linear least-squares problems. *BIT Numerical Mathematics*, 49(1):21–53.

Cartis, C., Gould, N. I., and Toint, P. L. (2011a). Adaptive cubic regularisation methods for unconstrained optimization. part i: motivation, convergence and numerical results. *Mathematical Programming*, 127(2):245–295.

Cartis, C., Gould, N. I., and Toint, P. L. (2012). Complexity bounds for second-order optimality in unconstrained optimization. *Journal of Complexity*, 28(1):93–108.

Cartis, C., Gould, N. I., and Toint, P. L. (2013). On the evaluation complexity of cubic regularization methods for potentially rank-deficient nonlinear least-squares problems and its relevance to constrained nonlinear optimization. *SIAM Journal on Optimization*, 23(3):1553–1574.

Cartis, C., Gould, N. I. M., and Toint, P. L. (2011b). *Adaptive cubic regularisation methods for unconstrained optimization. Part II: worst-case function- and derivative-evaluation complexity*. Springer-Verlag New York, Inc.

Chang, C.-C. and Lin, C.-J. (2011). Libsvm: a library for support vector machines. *ACM transactions on intelligent systems and technology (TIST)*, 2(3):27.

Conn, A. R., Gould, N. I., and Toint, P. L. (2000). *Trust region methods*. SIAM.

Curtis, F. E., Robinson, D. P., and Samadi, M. (2017). A trust region algorithm with a worst-case iteration complexity of $o(\epsilon^{-3/2})$ for nonconvex optimization. *Mathematical Programming*, 162(1-2):1–32.

Defazio, A., Bach, F., and Lacoste-Julien, S. (2014). Saga: A fast incremental gradient method with support for non-strongly convex composite objectives. In *Advances in Neural Information Processing Systems*, pages 1646–1654.

Fang, C., Li, C. J., Lin, Z., and Zhang, T. (2018). Spider: Near-optimal non-convex optimization via stochastic path integrated differential estimator. *arXiv preprint arXiv:1807.01695*.

Fang, C., Lin, Z., and Zhang, T. (2019). Sharp analysis for nonconvex sgd escaping from saddle points. *arXiv preprint arXiv:1902.00247*.

Garber, D. and Hazan, E. (2015). Fast and simple pca via convex optimization. *arXiv preprint arXiv:1509.05647*.

Ge, R., Huang, F., Jin, C., and Yuan, Y. (2015). Escaping from saddle pointsonline stochastic gradient for tensor decomposition. In *Conference on Learning Theory*, pages 797–842.

Ge, R., Lee, J. D., and Ma, T. (2016). Matrix completion has no spurious local minimum. In *Advances in Neural Information Processing Systems*, pages 2973–2981.

Golub, G. H. and Van Loan, C. F. (1996). *Matrix Computations (3rd Ed.)*. Johns Hopkins University Press, Baltimore, MD, USA.

Hardt, M. and Ma, T. (2016). Identity matters in deep learning. *arXiv preprint arXiv:1611.04231*.

Hillar, C. J. and Lim, L.-H. (2013). Most tensor problems are np-hard. *Journal of the ACM (JACM)*, 60(6):45.

Jin, C., Ge, R., Netrapalli, P., Kakade, S. M., and Jordan, M. I. (2017a). How to escape saddle points efficiently. *arXiv preprint arXiv:1703.00887*.

Jin, C., Netrapalli, P., Ge, R., Kakade, S. M., and Jordan, M. I. (2019). Stochastic gradient descent escapes saddle points efficiently. *arXiv preprint arXiv:1902.04811*.

Jin, C., Netrapalli, P., and Jordan, M. I. (2017b). Accelerated gradient descent escapes saddle points faster than gradient descent. *arXiv preprint arXiv:1711.10456*.

Johnson, R. and Zhang, T. (2013). Accelerating stochastic gradient descent using predictive variance reduction. In *Advances in neural information processing systems*, pages 315–323.

Kawaguchi, K. (2016). Deep learning without poor local minima. In *Advances in Neural Information Processing Systems*, pages 586–594.

Kohler, J. M. and Lucchi, A. (2017). Sub-sampled cubic regularization for non-convex optimization. *arXiv preprint arXiv:1705.05933*.

LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *Nature*, 521(7553):436–444.

Martínez, J. M. and Raydan, M. (2017). Cubic-regularization counterpart of a variable-norm trust-region method for unconstrained minimization. *Journal of Global Optimization*, 68(2):367–385.

Nesterov, Y. and Polyak, B. T. (2006). Cubic regularization of newton method and its global performance. *Mathematical Programming*, 108(1):177–205.

Nguyen, L. M., Liu, J., Scheinberg, K., and Takáč, M. (2017). Sarah: A novel method for machine learning problems using stochastic recursive gradient. In *International Conference on Machine Learning*, pages 2613–2621.

Nguyen, L. M., van Dijk, M., Phan, D. T., Nguyen, P. H., Weng, T.-W., and Kalagnanam, J. R. (2019). Optimal finite-sum smooth non-convex optimization with sarah. *arXiv preprint arXiv:1901.07648*.

Pinelis, I. (1994). Optimum bounds for the distributions of martingales in banach spaces. *The Annals of Probability*, pages 1679–1706.

Reddi, S. J., Hefny, A., Sra, S., Poczos, B., and Smola, A. (2016). Stochastic variance reduction for nonconvex optimization. pages 314–323.

Roux, N. L., Schmidt, M., and Bach, F. R. (2012). A stochastic gradient method with an exponential convergence _rate for finite training sets. In *Advances in Neural Information Processing Systems*, pages 2663–2671.

Royer, C. W. and Wright, S. J. (2017). Complexity analysis of second-order line-search algorithms for smooth nonconvex optimization. *arXiv preprint arXiv:1706.03131*.

Shalev-Shwartz, S. (2016). Sdca without duality, regularization, and individual convexity. In *International Conference on Machine Learning*, pages 747–754.

Shen, Z., Zhou, P., Fang, C., and Ribeiro, A. (2019). A stochastic trust region method for non-convex minimization. *arXiv preprint arXiv:1903.01540*.

Tripuraneni, N., Stern, M., Jin, C., Regier, J., and Jordan, M. I. (2018). Stochastic cubic regularization for fast nonconvex optimization. In *Advances in Neural Information Processing Systems*, pages 2904–2913.

Tropp, J. A. (2012). User-friendly tail bounds for sums of random matrices. *Foundations of computational mathematics*, 12(4):389–434.

Wang, Z., Ji, K., Zhou, Y., Liang, Y., and Tarokh, V. (2018a). Spiderboost: A class of faster variance-reduced algorithms for nonconvex optimization. *arXiv preprint arXiv:1810.10690*.

Wang, Z., Zhou, Y., Liang, Y., and Lan, G. (2018b). Sample complexity of stochastic variance-reduced cubic regularization for nonconvex optimization. *arXiv preprint arXiv:1802.07372*.

Xiao, L. and Zhang, T. (2014). A proximal stochastic gradient method with progressive variance reduction. *SIAM Journal on Optimization*, 24(4):2057–2075.

Xu, P., Roosta-Khorasani, F., and Mahoney, M. W. (2017). Newton-type methods for non-convex optimization under inexact hessian information. *arXiv preprint arXiv:1708.07164*.

Xu, Y., Rong, J., and Yang, T. (2018). First-order stochastic algorithms for escaping from saddle points in almost linear time. In *Advances in Neural Information Processing Systems*, pages 5531–5541.

Yu, Y., Xu, P., and Gu, Q. (2018). Third-order smoothness helps: faster stochastic optimization algorithms for finding local minima. In *Advances in Neural Information Processing Systems*, pages 4525–4535.

Yu, Y., Zou, D., and Gu, Q. (2017). Saving gradient and negative curvature computations: Finding local minima more efficiently. *arXiv preprint arXiv:1712.03950*.

Zhang, J., Xiao, L., and Zhang, S. (2018a). Adaptive stochastic variance reduction for subsampled newton method with cubic regularization. *arXiv preprint arXiv:1811.11637*.

Zhang, X., Wang, L., Yu, Y., and Gu, Q. (2018b). A primal-dual analysis of global optimality in nonconvex low-rank matrix recovery. In *International conference on machine learning*, pages 5857–5866.

Zhou, D., Xu, P., and Gu, Q. (2018a). Finding local minima via stochastic nested variance reduction. *arXiv preprint arXiv:1806.08782*.

Zhou, D., Xu, P., and Gu, Q. (2018b). Sample efficient stochastic variance-reduced cubic regularization method. *arXiv preprint arXiv:1811.11989*.

Zhou, D., Xu, P., and Gu, Q. (2018c). Stochastic nested variance reduced gradient descent for nonconvex optimization. In *Advances in Neural Information Processing Systems*, pages 3922–3933.

Zhou, D., Xu, P., and Gu, Q. (2018d). Stochastic variance-reduced cubic regularized Newton methods. In *Proceedings of the 35th International Conference on Machine Learning*, pages 5990–5999. PMLR.

Zhou, D., Xu, P., and Gu, Q. (2019). Stochastic variance-reduced cubic regularization methods. *Journal of Machine Learning Research*, 20(134):1–47.