Optimization Methods and Software

ISSN: 1055-6788 (Print) 1029-4937 (Online) Journal homepage: https://www.tandfonline.com/loi/goms20

Stochastic proximal quasi-Newton methods for non-convex composite optimization

Xiaoyu Wang, Xiao Wang & Ya-xiang Yuan

To cite this article: Xiaoyu Wang, Xiao Wang & Ya-xiang Yuan (2019) Stochastic proximal quasi-Newton methods for non-convex composite optimization, Optimization Methods and Software, 34:5, 922-948, DOI: 10.1080/10556788.2018.1471141

To link to this article: https://doi.org/10.1080/10556788.2018.1471141

Published online: 18 May 2018.

OPTIMIZATION METHODS & SOFTWARE
2019, VOL. 34, NO. 5, 922–948
https://doi.org/10.1080/10556788.2018.1471141

Taylor & Francis
Taylor & Francis Group

Stochastic proximal quasi-Newton methods for non-convex composite optimization

Xiaoyu Wang[a,b], Xiao Wang[b] and Ya-xiang Yuan[a]

[a]LSEC, Institute of Computational Mathematics and Scientific/Engineering Computing, AMSS, Chinese Academy of Sciences, Beijing, People's Republic of China; [b]School of Mathematical Sciences, University of Chinese Academy of Sciences, Beijing, People's Republic of China

ABSTRACT

In this paper, we propose a generic algorithmic framework for stochastic proximal quasi-Newton (SPQN) methods to solve non-convex composite optimization problems. Stochastic second-order information is explored to construct proximal subproblem. Under mild conditions we show the non-asympotic convergence of the proposed algorithm to stationary point of original problems and analyse its computational complexity. Besides, we extend the proximal form of Polyak-Łojasiewicz (PL) inequality to constrained settings and obtain the constrained proximal PL (CP-PL) inequality. Under CP-PL inequality linear convergence rate of the proposed algorithm is achieved. Moreover, we propose a modified self-scaling symmetric rank one incorporated in the framework for SPQN method, which is called stochastic symmetric rank one method. Finally, we report some numerical experiments to reveal the effectiveness of the proposed algorithm.

ARTICLE HISTORY
Received 28 January 2018
Accepted 17 April 2018

KEYWORDS
Non-convex composite optimization; Polyak-Łojasiewicz (PL) inequality; stochastic gradient; stochastic variance reduction gradient; symmetric rank one method; rank one proximity operator; complexity bound

MATHEMATICS SUBJECT CLASSIFICATION 2010
47N10; 65K10

1. Introduction

In this paper, we consider the following optimization problem:

$$\min_{x\in\mathcal{X}} P(x) = F(x) + h(x), \tag{1}$$

where $\mathcal{X}$ is a closed and convex set in $\mathbb{R}^d$ and $F(x)$ is an average of a number of component functions, i.e.

$$F(x) := \frac{1}{n}\sum_{i=1}^{n} f_i(x). \tag{2}$$

In particular, we are interested in the case when n is very large. Here, $f_i, i = 1, \ldots, n$, is smooth and possibly non-convex function. And h is a simple non-smooth convex function. Such kind of problems is fundamental and popular in machine learning and statistics,

CONTACT Xiao Wang wangxiao@ucas.ac.cn

known as the regularized empirical risk minimization . Many interesting problems, for example, the classification problem with sigmoid loss [4] and training neural networks [13,20] which are non-convex, have attracted more attention. If the regularization term is $h(x) = \lambda\|x\|_1$, it can promotes sparse solution, such as lasso [56], sparse logistic regression [53,60].

When h vanishes and $\mathcal{X} = \mathbb{R}^d$, the standard method to solve (1) and (2) is the gradient descent (GD) method [12] which can be described as follows:

$$x^{k+1} = x^k - \eta\nabla F(x^k) = x^k - \frac{\eta}{n}\sum_{i=1}^{n}\nabla f_i(x^k), \tag{3}$$

where η is the stepsize at kth iteration. However, when n is very large, computing the full gradient of F might be quite costly. To take advantage of the finite summation structure of F, the stochastic gradient-based methods update iterates as:

$$x^{k+1} = x^k - \eta B_k^{-1} g_k, \tag{4}$$

where g_k is an estimation to $\nabla F(x^k)$ and B_k is an approximation to the Hessian matrix $\nabla^2 F(x^k)$. If $B_k = \mathbb{I}_d$, (4) falls to the standard stochastic gradient (SG) method [44]. If B_k is some approximation generated based on stochastic gradients, (4) turns to stochastic quasi-Newton (SQN) methods. Although the computational cost per iteration in SG methods is lower, a lot of research has shown the superior overall performance of SQN methods, especially its advantages on highly nonlinear and ill-conditional problems [6,26,27]. Popular methods include oLBFGS [49], SGD-QN [7], RES [36], SQN [9], SdLBFGS [57], and SC-L-BFGS [15]. For details interested readers are referred to [57]. Although SG methods always make rapid progress in early iterations, it is slow and unstable near the optima due to the stochastic variance. To deal with this drawback, kinds of variance reduction techniques are proposed, such as SAG [48], SVRG [22,41], SDCA [51], SAGA [16,42], MISO [33] and so on. They can achieve the globally linear convergence rate when solving strongly convex problems. Recently, SQN methods based on variance reduction attract much interest, including SLBFGS [37], LiSSA [1], SdLBFGS-VR [57], and IQN [35]. It is shown that LiSSA [1] and IQN [35] can achieve globally linear convergence rate and locally superlinear convergence rate in strongly convex case. In addition, both SdLBFGS-VR [57] and SC-L-BFGS [15] have shown nice performances for solving general non-convex problems.

For general composite optimization (1) and (2), the well-known proximal gradient descent (Prox-GD) [34] method updates iterates as:

$$x^{k+1} = \mathrm{prox}_h(x^k - \eta\nabla F(x^k)), \tag{5}$$

where $\mathrm{prox}_h(y) = \arg\min_y\{h(y) + \frac{1}{2}\|x - y\|^2\}$. In order to utilize the second-order information of the objective function, the proximal Newton-type methods attract more attention. Those methods normally use the following piecewise quadratic model at kth iteration to approximate the objective function:

$$Q_k(y) = F(x^k) + \langle\nabla F(x^k), y - x^k\rangle + \tfrac{1}{2}(y - x^k)^T B_k(y - x^k) + h(y), \tag{6}$$

where B_k is (or an approximation to) the Hessian $\nabla^2 F(x^k)$. Such methods include proximal Newton or quasi-Newton methods [5,19,29,30], as well as projected Newton-type

methods [28,46,47] for constrained problems. It has been shown in [29] that the proximal Newton-type methods can achieve q-quadratical and q-superlinear convergence rates under standard assumptions if the approximate model (6) is minimized exactly. Since it is normally impractical to approximate model (6) exactly for general matrices B_k, Lee *et al.* [29] also consider inexact proximal Newton-type methods and show that q-linear and q-superlinear convergence rates can also be achieved under mild conditions. When $h = \lambda\|x\|_1$, there are a line of research on efficient proximal Newton-type methods for solving (1), including glmnet [17], QUIC [21], IMRO [24], BAS [8], and SQA [11]. These methods are designed to exploit the specific form of h or the structure of hessian of specific function to solve (6) more efficiently. Most of above mentioned works consider inexact minimizer of (6) except [5] and [24]. To obtain exact minimizer of (6), B_k [5,24] is set as a summation of a diagonal matrix and a rank one matrix. Note that all above mentioned methods require the full gradient $\nabla F(x)$, which in many cases is time-consuming. Proximal stochastic gradient methods, which use stochastic gradient g_k to approximate $\nabla F(x^k)$ in (6), attract much interest due to its advantages capturing the finite-sum structure of the original problem, such as RSPG [18], Prox-SVRG [43,58], SAGA [42], Prox-SDCA [50], and Natasha [2,3]. Motivated by the successful applications of stochastic quasi-Newton methods for smooth problems, we have reason to believe that exploring second-order information based on stochastic proximal gradient methods will help yield better performances. Some recent works [31,32,45,52] study stochastic proximal quasi-Newton (SPQN) methods for solving convex composite optimization. However, literatures focusing on non-convex problems are very limited. The key challenge to face is how to construct an effective Hessian approximation B_k such that the proximal quadratic model (6) is easy to be minimized and the global convergence of the proposed algorithm can be guaranteed. As is known, symmetric rank one (SR1) method is one of the most important quasi-Newton methods in nonlinear optimization. It shows superior performances when solving non-convex problems [14,25] . Inspired by this, we will in this paper explore the potential of an SPQN method integrated with symmetric rank one update for solving non-convex composite optimization (1) and (2).

To ensure the proximal model (6) uniquely minimized, we normally require B_k to be a symmetric positive definite matrix. However, the classic SR1 update cannot guarantee the positive definiteness of B_k, even though curvature condition $y_k^T s_k > 0$ holds where $s_k = x^{k+1} - x^k$ and $y_k = \nabla F(x^{k+1}) - \nabla F(x^k)$. In deterministic settings, we always exploit line search or trust region technique to guarantee the convergence of SR1 method. However, these techniques are impractical in stochastic settings, since the exact function and gradient values are not attainable. Osborne and Sun [39] have proposed a Davidon's optimal condition of self-scaling symmetric rank one (OCSSR1) method. They prove that the positive definiteness of B_k can be perserved provided that the curvature condition is satisfied. So now the question is how to satisfy the curvature condition. There are two widely used techniques in deterministic optimization: skipping [38] and damping [40]. However, the latter is more popular in stochastic optimization, such as SdLBFGS [57], RES [36], SC-BFGS [15]. In our work, we will apply the damping technique to propose a scheme to construct uniformly positive definite Hessian approximation B_k.

Contributions. Our contributions in this paper are in several folds.

(i) We present a generic algorithm framework SPQN for stochastic proximal quasi-Newton methods for solving general non-convex composite optimization problem (1) and (2). We employ the operator $\mathcal{D}_h^{\mathcal{X}}$ (see (11)) to present a new convergence criterion to analyse theoretical properties of SPQN, which is different from generalized projection operator widely used in literatures[18,43]. Application of $\mathcal{D}_h^{\mathcal{X}}$ can yields lower complexity bound than previous works.

(ii) We extend the proximal form of Polyak–Łojasiewicz (PL) inequality [23] to constrained settings, named as constrained proximal Polyak–Łojasiewicz (CP-PL) inequality. For problems satisfying this inequality, SPQN can achieve globally linear convergence rate.

(iii) We propose a modified self-scaling SR1 (MSSR1) method that falls into the framework of SPQN, which we call stochastic symmetric rank one (StSR1) method. This method provides an update strategy to generate uniformly positive definite Hessian approximation B_k. Such B_k can make the subproblem solved more easily at each iteration. We also present explicit expressions for accurate solution of subproblem.

Organization. This paper is organized as follows. In Section 2 we present some preliminary definitions and properties of operator $\mathcal{D}_h^{\mathcal{X}}$; In Section 3 we present the SPQN algorithm and give its theoretical properties; In Section 4 we present an MSSR1 method to update the Hessian approximations; In Section 5 we report some numerical results to show the efficiency of StSR1; Finally, we draw some conclusions in Section 6.

Notation. Throughout this paper, we use $\langle\cdot,\cdot\rangle$ to denote the Euclidean inner product, $\|\cdot\|$ to denote the usual Euclidean norm, i.e. $\|\cdot\|_2$, unless otherwise specified. We denote $\mathbb{I}_d$ as the identity matrix on $\mathbb{R}^{d\times d}$. For any real value r, we use $\lceil r\rceil$ and $\lfloor r\rfloor$ to denote the nearest integer to r from above and below, respectively. For any $A\in\mathbb{R}^d$, the notation A^{-1} represents its inverse. The expectation with respect to a random variable ξ is denoted by $\mathbb{E}_\xi[\cdot]$. For any function $f,g:\mathbb{R}^d\longmapsto\mathbb{R}^d$, the compostion function $f\circ g$ is defined by $f\circ g(x)=f(g(x))$ for any $x\in\mathbb{R}^d$. For a given $A\in\mathbb{R}^{d\times d}$, $\mathrm{tr}(A)$ denotes the sum of all diagonal elements of A, $\lambda_{\min}(A)$ and $\lambda_{\max}(A)$ denote the smallest and largest eigenvalues, respectively.

2. Preliminaries

We first give the following assumption which is required throughout this paper.

AS1 (a) each f_i is twice continuously differentiable, bounded below and L-smooth, i.e. there is a constant $L>0$ such that $\|\nabla f_i(x)-\nabla f_i(y)\|\le L\|x-y\|$, $\forall\, x,y\in\mathcal{X}$.
(b) $h(x)$ is lower semi-continuous ,[1] convex but possibly non-smooth;
(c) $\mathcal{X}$ is a closed convex subset of $\mathbb{R}^d$.

From AS1(a), we can obtain an important property of F, that is

$$F(y)\le F(x)+\langle\nabla F(x),y-x\rangle+\frac{L}{2}\|y-x\|^2,\quad \forall x,y\in\mathcal{X}. \tag{7}$$

Hence, F is also L-smooth. Here we assume that L is independent of n. For twicely differentiable functions, the property of L-smoothness means that the Hessian of the function is uniformly bounded, which is quite general in the nonlinear optimization.

It is popular in literatures to use the generalized projected gradient to analyse the convergence of algorithms for non-convex constrained composite optimization [18,43]. The generalized projected gradient $P_{\mathcal{X}}(x, g, \alpha)$ [18] is normally defined as

$$P_{\mathcal{X}}(x, g, \alpha) = \alpha(x - x^+), \tag{8}$$

where

$$x^+ = \arg\min_{y \in \mathcal{X}} \left\{ \langle g, y - x \rangle + \frac{\alpha}{2}\|y - x\|^2 + h(y) - h(x) \right\} \tag{9}$$

with $\alpha > 0$. In this paper, however, we analyse the theoretical performances of the proposed algorithm from a different point of view. This is motivated by the proximal PL inequality [23]: there exists a constant $\mu > 0$ such that

$$\tfrac{1}{2}\mathcal{D}_h(x, L) \geq \mu(P(x) - P^*), \tag{10}$$

where

$$\mathcal{D}_h(x, \alpha) = -2\alpha \min_y \left\{ \langle \nabla F(x), y - x \rangle + \frac{\alpha}{2}\|y - x\|^2 + h(y) - h(x) \right\} \quad \forall \alpha > 0. \tag{11}$$

Here P^* is the optimal value of objective function without any constraint. Notice that different from $P_{\mathcal{X}}$, the operator $\mathcal{D}_h$ is defined on function value at minimizer rather than the optimal point x^+. We now extend $\mathcal{D}_h$ to the constrained case. Here, we define an operator $\mathcal{D}_h^{\mathcal{X}}$:

$$\mathcal{D}_h^{\mathcal{X}}(x, g, B, \alpha) = -2\alpha \min_{y \in \mathcal{X}} \left\{ \langle g, y - x \rangle + \frac{\alpha}{2}\|y - x\|_B^2 + h(y) - h(x) \right\}, \quad \forall\, \alpha > 0, x \in \mathcal{X}, \tag{12}$$

where $B \in \mathbb{R}^{d \times d}$ is symmetric positive definite. The definition $\mathcal{D}_h^{\mathcal{X}}(x, g, B, \alpha)$ is different from that of $\mathcal{D}_h(x, \alpha)$ in (11). The first difference is that we extend the l_2-norm to general B-norm as the proximal term in $\mathcal{D}_h^{\mathcal{X}}(x, g, B, \alpha)$. The other is that $\nabla F(x)$ is replaced by any vector $g \in \mathbb{R}^d$. As is known when h is convex and the matrix B is symmetric positive definite, the definition of $\mathcal{D}_h^{\mathcal{X}}$ is reasonable.

Next, we give some important properties of the operator $\mathcal{D}_h^{\mathcal{X}}$ and characterize its relationship with $P_{\mathcal{X}}$. All the proofs are given in the appendix.

Lemma 2.1: *If x^+ is given by (9), then for any $x \in \mathcal{X}$, we have*

$$\langle g, P_{\mathcal{X}}(x, g, \alpha) \rangle \geq \|P_{\mathcal{X}}(x, g, \alpha)\|_B^2 + \alpha(h(x^+) - h(x)). \tag{13}$$

Lemma 2.2: *For any fixed $B \succ 0$, we have*

$$\mathcal{D}_h^{\mathcal{X}}(x, g, B, \alpha) \geq \|P_{\mathcal{X}}(x, g, \alpha)\|_B^2, \quad \forall x \in \mathcal{X},\ \alpha > 0. \tag{14}$$

Lemma 2.3: *For differentiable function f and convex function h, for fixed x, g, B, we have*

$$\mathcal{D}_h^{\mathcal{X}}(x,g,B,\alpha_2) \geq \mathcal{D}_h^{\mathcal{X}}(x,g,B,\alpha_1), \quad \forall\, \alpha_2 \geq \alpha_1 > 0. \tag{15}$$

Lemma 2.3 shows the monotonicity of the operator $\mathcal{D}_h^{\mathcal{X}}(x,g,B,\alpha_2)$ for fixed x,g and B. The inequality (15) has been shown for unconstrained case (refer to the Lemma 1 of Appendix E in [23]). We extend the result to constrained case and give a more simple proof than the prior literatures.

Definition 2.1: A Constrained Proximal Polyak-Łojasiewicz (CP-PL) inequality holds if there exists a constant $\mu > 0$ such that

$$\tfrac{1}{2}\mathcal{D}_h^{\mathcal{X}}(x,\nabla F(x),\mathbb{I}_d,L) \geq \mu(P(x)-P^*), \quad \forall x \in \mathcal{X}, \tag{16}$$

where P^* is the optimal value of problem (1) and (2).

It follows from Lemma 2.2 that $\mathcal{D}_h^{\mathcal{X}}$ is non-negative, which implies (2.1) is reasonable. We note that such a function P that satisfies (2.1) can be nonconvex. There are various functions that satisfy (2.1). In particular, all γ-strongly convex functions on the feasible set $\mathcal{X}$ satisfy the CP-PL inequality with $\mu = \lambda$.

In the followings, we use $\mathcal{D}_h^{\mathcal{X}}$ to analyse the theoretical properties of the proposed algorithm. First, we give the definition of ϵ-approximate solution.

Definition 2.2: A point $\bar{x} \in \mathcal{X}$ is said to be an ϵ-approximate solution of (1) and (2), if

$$\mathbb{E}[\mathcal{D}_h^{\mathcal{X}}(\bar{x},\nabla F(\bar{x}),\mathbb{I}_d,\alpha)] \leq \epsilon.$$

In this paper, we assume that the stochastic gradient can be obtained by a stochastic first-order oracle $\mathcal{SFO}$ [18]. $\mathcal{SFO}$ takes an index $i \in [n]$ and a point $x \in \mathbb{R}^d$, and returns $\nabla f_i(x)$. For problem (1) and (2), we assume that a constrained proximal oracle($\mathcal{CPO}$) can be obtained by taking a point x then returns an output of $D_h^{\mathcal{X}}$.

3. A framework for SPQN methods for non-convex composite optimization

In this section, we propose a general framework for SPQN methods for non-convex composite optimization (1) and (2). We first present the pseudocode of SPQN for solving problem (1) and (2) as Algorithm 3.1.

Algorithm 3.1 SPQN(x^0, B_0, T, m, b, η)

Input: starting vector $\hat{x}^0 = x_m^0 = x^0 \in \mathcal{X}$, initial matrix $B_0^1 = B_0$, inner loop update frequency m, batch size b and learning rate η
for $t = 0, 1, \ldots, S-1$ **do**
  Set $x_0^{t+1} = x_m^t$
  Calculate $\hat{g} = \frac{1}{n}\sum_{i=1}^{n} \nabla f_i(\hat{x}^t)$
  for $j = 0$ to $m-1$ **do**
    Pick a minibatch set M_j^t uniformly at random from $\{1, 2, \ldots, n\}$(with replacement) such that $|M_j^t| = b$
    Calculate $g_j^{t+1} = \frac{1}{b}\sum_{i\in M_j}(\nabla f_i(x_j^{t+1}) - \nabla f_i(\hat{x}^t)) + \hat{g}$
    Obtain the exact solution x_{j+1}^{t+1} of the following subproblem

$$\min_{y\in\mathcal{X}} \quad \left\langle g_j^{t+1}, y - x_j^{t+1}\right\rangle + \frac{1}{2\eta}\left\|y - x_j^{t+1}\right\|_{B_j^{t+1}}^2 + h(y) - h(x_j^{t+1}) \tag{17}$$

    Generate a symmetric positive definite matrix B_{j+1}^{t+1}
  end for
  Set $\hat{x}^{t+1} = x_m^{t+1}$
end for
Output: x_a, uniformly chosen from $\left\{\left\{x_j^{t+1}\right\}_{j=0}^{m-1}\right\}_{t=0}^{S-1}$

Notice that SPQN algorithm is a two-loop procedure. At each outer iteration, we choose a candidate point $\hat{x}^t$ at which the full gradient $\nabla F(\hat{x}^t)$ is calculated. This full gradient is used in each inner iteration to construct a stochastic gradient g_j^{t+1}. It is easy to check that the conditional expectation satisfies $\mathbb{E}[g_j^{t+1}|x_j^{t+1}] = \nabla F(x_j^{t+1})$. The subproblem is built based on a Hessian approximation matrix B_j^{t+1}, about which we need the following assumption.

AS2 For any $t = 0, \ldots, S-1, j = 0, \ldots, m-1$, B_j^t is independent of M_j^t and there exist two positive constants $\underline{\lambda}, \overline{\lambda}$ such that

$$\underline{\lambda}\mathbb{I}_d \preccurlyeq B_j^t \preccurlyeq \overline{\lambda}\mathbb{I}_d.$$

The assumption AS2 is quite common for the quasi-Newton methods in stochastic optimization [15,57]. We will specify the way to construct B in Section 4.

3.1. Theoretical properties

In this part, we will analyse the theoretical convergence properties of SPQN.

Theorem 3.1: *Under assumptions* AS1–AS2, *assume* $c_m = 0$, $c_j = c_{j+1}(1 + (1/\beta)) + (2L^2/b\theta)$ $(\beta, \theta > 0)$, $\eta \le (\underline{\lambda}/(\theta + L + 2c_0(1+\beta)))$, *and* $T = Sm$. *Then for the output* x_a

of SPQN, we have

$$\mathbb{E}\left[\mathcal{D}_g^{\mathcal{X}}\left(x_a, \nabla F(x_a), \mathbb{I}_d, \frac{\overline{\lambda}}{\eta}\right)\right] \leq \frac{2\overline{\lambda}(P(x^0) - P^*)}{\eta T}, \tag{18}$$

where P^* is the optimal value of problem (1) and (2).

If we set η equal to its upper bound, that is $\eta = \underline{\lambda}/(\theta + L + 2c_0(1+\beta))$, η may depend on n, because both β and θ may depend on n. In order to explicitly express the dependence, we appropriately give some specific value of m, β and θ as follows.

Theorem 3.2: *Under the same conditions as Theorem* 3.1, *further assume that* $m = \lfloor n^r \rfloor (r > 0)$, $\beta = n^r$, $\theta = Ln^r/\sqrt{b}$, *then there exists a constant* $\nu > 0$ *such that* $\eta = (\nu\underline{\lambda}\sqrt{b}/Ln^r)$, *and*

$$\mathbb{E}\left[\mathcal{D}_h^{\mathcal{X}}\left(x_a, \nabla F(x_a), \mathbb{I}_d, \frac{\overline{\lambda}}{\eta}\right)\right] \leq \frac{2L\overline{\lambda}n^r}{\nu\underline{\lambda}\sqrt{b}T}(P(x^0) - P^*). \tag{19}$$

Corollary 3.1 (Complexity): *Under the same conditions as Theorem* 3.2, *assume that* $b = n^{2/3}$, $m = n^{1/3}$, *then the step size* $\eta \leq (\nu\underline{\lambda}/L)$. *Therefore, the SFO and CPO complexity of Algorithm* 3.1 *to achieve an* ϵ*-approximate solution of (1) and (2) are* $O(n + (\kappa_1 n^{2/3}/\epsilon))$ *and* $O(\kappa_1/\epsilon)$, *respectively, where* $\kappa_1 = \overline{\lambda}/\underline{\lambda}$.

Recall that the initial proximal SVRG method for non-convex composite optimization achieves the $\mathcal{SFO}$ complexity $O(n + (n^{2/3}/\epsilon))$ [43]. Compared with this, SPQN achieves a similar result which adds the coefficient κ_1 which to some extent reflects the condition number of Hessian approximations. If we choose all Hessian approximations equal to the identity matrix and $\mathcal{X} = \mathbb{R}^d$, SPQN turns to initial proximal SVRG and the complexity bound becomes the same as that of [43].

3.2. Linear convergence rate under CP-PL inequality

In this subsection, we analyse the convergence rate of SPQN method for solving problems satisfying the CP-PL inequality (16). We now present a variant algorithm GD-SPQN.

Algorithm 3.2 GD-SPQN($x^0, B_0, T, m, b\eta$)

1: **Input**: starting vector $\hat{x}^0 = x_m^0 = x^0 \in \mathcal{X}$, initial matrix B_0, innerloop update frequency m and learning rate η
2: **for** $k = 0, 1, \ldots, K-1$ **do**
3: $\quad x^{k+1} = \text{SPQN}(x^k, B_k, T, m, b, \eta)$
4: **end for**
5: **Output:** x^K

The following theorem shows the globally linear convergence rate of GD-SPQN.

Theorem 3.3: *Under the same conditions as Theorem* 3.2, *assume the CP-PL inequality (16) holds with the parameter* $\mu > 0$ *and set the parameter T as* $T = \lceil((L\bar{\lambda})/(2\mu\nu\underline{\lambda}))(n^r/\sqrt{b})\rceil$, *then we have*

$$\mathbb{E}[P(x^k) - P^*] \leq (2^{-k})[P(x^0) - P^*)]. \tag{20}$$

Corollary 3.2 (Complexity): *Under the same conditions as Theorem* 3.3, *the* $\mathcal{SFO}$ *and* $\mathcal{CPO}$ *complexity of GD-SPQN to achieve* ϵ*-approximate solution are* $O((n + \kappa_1\kappa_2((n/\sqrt{b}) + n^r\sqrt{b}))\log(1/\epsilon))$ *and* $O((\kappa_1\kappa_2 n^r/\sqrt{b})\log(1/\epsilon))$, *respectively*, *where* $\kappa_1 = (\bar{\lambda}/\underline{\lambda})$, $\kappa_2 = L/\mu$.

4. A modified self-scaling SR1 method

In this section, we propose a modified self-scaling symmetric rank one (MSSR1) method to generate the Hessian approximation satisfying the assumption AS2. In deterministic optimization, the classic SR1 method for minimizing $f(x)$ calculates iterates through (4) where the quasi-Newton matrix B_k is updated by

$$B_{k+1} = B_k + \frac{(y_k - B_k s_k)(y_k - B_k s_k)^T}{(y_k - B_k s_k)^T s_k},$$

where $s_k = x_{k+1} - x_k$ and $y_k = \nabla f(x_{k+1}) - \nabla f(x_k)$. By setting $H_k = B_k^{-1}$, the inverse update of SR1 is formulated by

$$H_{k+1} = H_k + \frac{(s_k - H_k y_k)(s_k - H_k y_k)^T}{(s_k - H_k y_k)^T y_k}.$$

For related works on the SR1 methods interested readers are referred to [10,14,25]. Compared with other quasi-Newton methods, SR1 method normally shows superior performance on solving non-convex problems [10,14]. However, classic SR1 method has a major drawback that it cannot guarantee the positive definiteness of quasi-Newton matrices even even though the curvature condition $s_k^T y_k > 0$ is satisfied. Many works are proposed to modify the SR1 update to preserve positive definiteness, see [54]. There are some successful attempts in a trust region or line search framework to avoid the loss of positive definiteness [10,25]. These techniques, however, are impractical to implement in stochastic settings, since exact function and gradient values are not accessible. Another important approach to preserve positive definiteness of the SR1 update is to scale the current update [39,55]. Osborne and Sun [39] propose a self-scaling SR1 method, OCSSR1, which updates

$$H_{k+1} = \tau H_k + \frac{(s_k - \tau H_k y_k)(s_k - \tau H_k y_k)^T}{(s_k - \tau H_k y_k)^T y_k}, \tag{21}$$

where the scalar scaling parameter τ can be chosen as $\tau = (a/b) - \sqrt{(a/b)^2 - (a/c)}$, with $a = s_k^T B_k s_k$, $b = y_k^T s_k$ and $c = y_k^T H_k y_k$. Sun [55] presents a limited memory variant of OCSSR1 method. Combining with the scaling strategy, it could not only adjust the eigenvalue distribution of updating matrices to improve the SR1 algorithm, but also maintain the positive definiteness of updating matrices. So the key issue lies on how to guarantee the curvature condition $s_k^T y_k > 0$. In our iterative framework, we refer a special damping

technique in SC-BFGS [15] that the vector v_k is defined to be the linear combination of s_k and ηy_k, that is

$$v_k = \beta s_k + (1-\beta)\eta y_k$$

for some $\beta \in [0,1]$ satisfying the condition that there exist constants $\theta_1 \in (0,1), \theta_2 \in (1,\infty)$ such that

$$\theta_1 \le \frac{v_j^T s_j}{s_j^T s_j}, \quad \frac{v_j^T v_j}{v_j^T s_j} \le \theta_2. \tag{22}$$

Then by replacing y_k with v_k in (21) we obtain a modified self-scaling SR1(MSSR1) method.

For the unconstrained case, i.e. $\mathcal{X} = \mathcal{R}^d$, the subproblem (17) is equivalent to solve the following scaled proximal operator per iteration:

$$\tilde{x} = \text{prox}_h^{H^{-1}}(x - \eta H g),$$

where H is the inverse of B. However, it is often difficult to solve this scaled proximal operator precisely for general symmetric positive definite matrix B. Theorem 7 in [5] shows that if H has special structure with a diagnosed matrix plus a rank one correction, i.e. $H = D + uu^T$, it is possible to implement the calculation of this proximity operator efficiently. The next theorem extends the result to the case $H = D - uu^T$.

Theorem 4.1: *Let h be a proper and lower semi-continuous convex function, and $H = D + \sigma uu^T$ (σ is $+1$ or -1), where D is a diagonal matrix with positive diagonal elements and $u \in \mathbb{R}^d$. Then we have*

$$prox_h^H(x) = D^{-1/2} \circ prox_{h\circ D^{-1/2}}(D^{1/2}x - \sigma v),$$

where $v = \alpha D^{-1/2}u$ and α is the unique root of the function

$$p(\alpha) = \langle u, x - D^{-1/2} \circ prox_{h\circ D^{-1/2}} \circ D^{1/2}(x - \sigma\alpha D^{-1}u)\rangle + \alpha,$$

which is a Lipschitz continuous and strictly increasing function on $\mathbb{R}$.

The proof refers to appendix. Becker and Fadili [5] prove the case that $\sigma = 1$. In our numerical experiments, we compute prox_h^H corresponding to the case $\sigma = -1$. It follows from Theorem 4.1 that for special h, such as L_1 norm, L_∞-ball, box constraint and positive constraint, the scaled operator prox_h^H will be easily handled and can be expected to be computed exactly at cost of $d\log(d)$ [5].

In order to obtain a minimizer of (6) efficiently, we now propose a modified self-scaling symmetric rank one (MSSR1) method with H_k in (21) set as the identity matrix $\mathbb{I}_d$, which yields the zero memory form of OCSSR1 method. We generate the auxiliary stochastic gradient at x_{j+1}^{t+1} with the sampling in previous jth iteration:

$$\bar{g}_{j+1}^{t+1} := \frac{1}{b}\sum_{i\in M_j} \nabla f_i(x_{j+1}^{t+1}).$$

The stochastic gradient difference y_j is defined as

$$y_j := \bar{g}_{j+1}^{t+1} - g_j^{t+1}. \tag{23}$$

Algorithm 4.1 Modified self-scaling symmetric rank one (MSSR1)

Input: Given $\epsilon > 0, \theta_1 \in (0,1)$ and $\theta_2 \in (1,\infty)$
Set $s_j = x_{j+1}^{t+1} - x_j^{t+1}$, $y_j = \bar{g}_{j+1}^{t+1} - g_j^{t+1}$
Compute $\beta_j = \arg\min \left\{ \beta \in [0,1] \mid v(\beta) = \beta s_j + (1-\beta)\eta y_j \text{ satisfies (22)} \right\}$
Set $v_j = v(\beta_j)$
Compute $\tau = \frac{s_j^T s_j}{v_j^T s_j} - \left(\frac{(s_j^T s_j)^2}{(v_j^T s_j)^2} - \frac{s_j^T s_j}{v_j^T v_j}\right)^{\frac{1}{2}}$ and $\rho = v_j^T s_j - \tau v_j^T v_j$
if $\rho \leq \epsilon \left\| s_j - \tau v_j \right\| \left\| v_j \right\|$ **then**
    set $u_j = 0$
else
    set $u_j = \frac{s_j - \tau v_j}{\sqrt{\rho}}$
end if
Set $H_{j+1}^{t+1} = \tau \mathbb{I}_d + u_j u_j^T$

The following theorem shows that H_{j+1}^{t+1} generated by MSSR1 is symmetric and uniformly positive definite.

Theorem 4.2: *Let H_{j+1}^{t+1} be updated by MSSR1. Then we have*

$$\underline{\lambda}\mathbb{I}_d \preccurlyeq H_{j+1}^{t+1} \preccurlyeq \overline{\lambda}\mathbb{I}_d,$$

where $\underline{\lambda} = 1/(2d\theta_2)$, $\overline{\lambda} = \tau d + 1/(\epsilon\theta_1)$.

Since y_j defined in (23) is independent of current random sampling set M_{j+1}, together with Theorem 4.2 it follows that H_{j+1}^{t+1} satisfies AS2. According to *Sherman-Morrison* formula, B_{j+1}^{t+1} is guaranteed to satisfy the assumption AS2. Therefore, the theoretical properties of StSR1 method, which applies MSSR1 method to update the Hessian inverse matrix in the framework of SPQN, can be guaranteed based on the analysis in Section 3.

5. Numerical experiments

In this section, we empirically test the StSR1 method which is the MSSR1 update incorporated in the framework of SPQN and compare its performance with some related algorithms.

We consider the following sparse non-convex support vector machine (SVM) problem with a sigmoid loss function considered in [59]:

$$P(x) = \frac{1}{n}\sum_{i=1}^{n}(1 - \tanh(b_i\langle a_i, x\rangle)) + \lambda\|x\|_1, \tag{24}$$

where $\{(a_i, b_i)\}_{i=1}^n$ is a training sample set with $a_i \in \mathbb{R}^d$ being the feature vector and $b_i \in \{-1, +1\}$ being the corresponding label. And $\lambda \geq 0$ is the regularization parameter. In our numerical experiments, we compare StSR1 to Prox-SVRG [43] and standard proximal gradient method, abbreviated as Prox-GD [34]. All the methods were implemented in

Table 1. Summary of datasets and regularization parameters used in numerical experiments.

Dataset	*n*/*N*	*d*	λ
rcv1.binary	13,495/20,242	47,236	10^{-5}
w6a	17,188/49,749	300	10^{-5}
real-sim	48,206/72,309	20,958	10^{-5}

Matlab 2014b under Windows 7 operating system on Dell desktop with Intel(R) Core(TM) i7-4790U CPU @3.6 GHz, 8 GB Memory.

For the problem (24), algorithms were tested with different input parameters. We report the squared norm of gradient (training data) and accuracy(percentage of correctly classified testing data) as criterions to measure the performance of tested algorithms. For all those algorithms, we compare these criterions against the number of effective pass through the data, that is $\mathcal{SFO}$ calls divide by n. The algorithms were terminated when the total number of component gradient evaluations, i.e. $\mathcal{SFO}$, is larger than the maximum value we set. We run n proximal SGD iterations and obtain an iterate as the starting point for all algorithms, suggested in [43]. For MSSR1, all the combination of $\theta_1 \in \{2^{-3}, 2^{-4}, 2^{-5}, 2^{-6}, 2^{-7}\}$ and $\theta_2 \in \{2^1, 2^2, 2^3\}$ were tested. For the minibatch size b, we test on $b \in \{1, n^{1/3}/2, n^{1/3}, 2n^{1/3}, 3n^{1/3}\}$. Moreover, we set $\epsilon = 10^{-12}$. In the experiments, we choose the same range of stepsize for Prox-SVRG and Prox-GD methods, that is $\eta \in \{10^{-2}, 10^{-1}, 1, 10^1, 10^2\}$, and choose $\eta \in \{2^{-2}, 2^{-1}, 1, 2^1, 2^2\}$ for StSR1.

In Table 1, we list three datasets from LIBSVM website[2] we tested in our numerical experiments. Here, n denotes the number of the training data and N denotes the number of the whole data including both training and testing data. And d is the dimension of the dataset. The regularization parameter $\lambda = 10^{-5}$ is set for every dataset. For the dataset w6a, we use the training data and testing data from the website provided. For the datasets real-sim and rcv1.binary, we use 2/3 of the data as the training data, and the remaining 1/3 as the testing data.

5.1. Experiment results on rcv1.binary

In this subsection, we compare StSR1, Prox-GD and Prox-SVRG on rcv1.binary dataset for solving (24). Firstly, we compare the impact of the input parameters $(b, \theta_1, \theta_2, \eta)$ on StSR1. In Figure 1, we compare StSR1 with different minibatch size b. The input parameters $(\theta_1, \theta_2, \eta) = (2^{-5}, 2^2, 1)$. The best batchsize $b = 0.5n^{1/3}$ is achieved according to the accuracy in the right figure. Moreover, Figure 1 shows that the minibatch technique usually provides better performance in practice and make the algorithm more stable.

The impact of parameter θ_1 for MSSR1 method on rcv1.binary is shown in Figure 2. The parameter θ_1 is chosen from the set $\{2^{-3}, 2^{-4}, 2^{-5}, 2^{-6}, 2^{-7}\}$. Other used parameters are set as $(b, \theta_2, \eta) = (0.5n^{1/3}, 2^2, 1)$. From Figure 2 we can see that StSR1 method is not sensitive to θ_1, although smaller θ_1 can achieve a better performance. However, the smaller θ_1 may hurt the accuracy according to Figure 2. Overall speaking, the best θ_1 is achieved at $\theta_1 = 2^{-6}$. In our numerical experiments, we find that for any $\theta_2 \in \{2, 2^2, 2^3\}$ the same performance can be achieved. Thus, we fix $\theta_2 = 2^2$ in all following experiments.

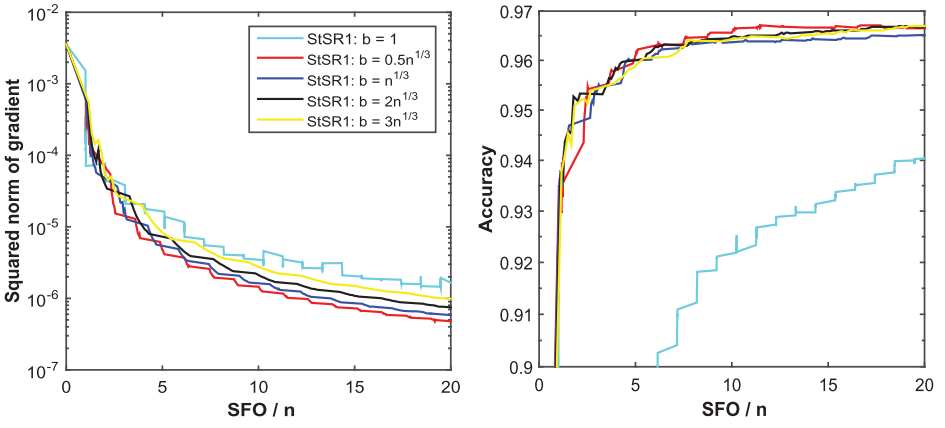

Figure 1. Comparison on rcv1.binary dataset with different minibatch size b for StSR1 method. The input parameters are set as $(\theta_1, \theta_2, \eta) = (2^{-5}, 2^2, 1)$.

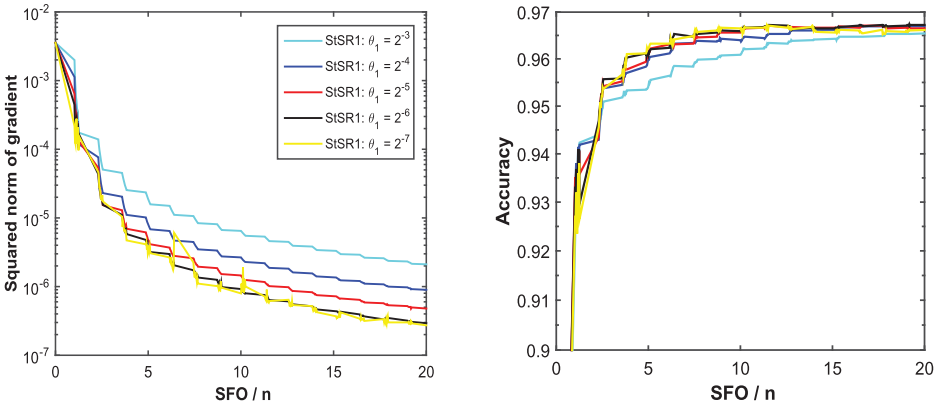

Figure 2. Comparison on rcv1.binary dataset with different θ_1 for StSR1 method. The input parameters are set as $(b, \theta_2, \eta) = (0.5n^{1/3}, 2^2, 1)$.

Figure 3 shows the influence of stepsize on rcv1.binary dataset. We can observe that StSR1 is not very sensitive to stepsize but prefer smaller stepsize. Thus, we set the exponential change of stepsize is based on 2, that is $\eta \in \{2^{-2}, 2^{-1}, 1, 2^1, 2^2\}$. The best chosen stepsize η is $\eta = 1$. The other input parameters are set as $(b, \theta_1, \theta_2) = (0.5n^{1/3}, 2^{-6}, 2^2)$.

In Figure 4, we compare the three methods StSR1, Prox-GD and Prox-SVRG on rcv1.binary dataset. To fairly compare all the three algorithms, we test them with their best parameter settings. For StSR1 the best performance is achieved with parameters set as $(b, \theta_1, \theta_2, \eta) = (0.5n^{1/3}, 2^{-6}, 2^2, 1)$, while for Prox-GD, the best performance is achieved when $\eta = 10^2$. And according to the accuracy Prox-SVRG performs best when $\eta = 10$. Figure 4 shows that StSR1 performs better compared to best-tuned Prox-GD and Prox-SVRG.

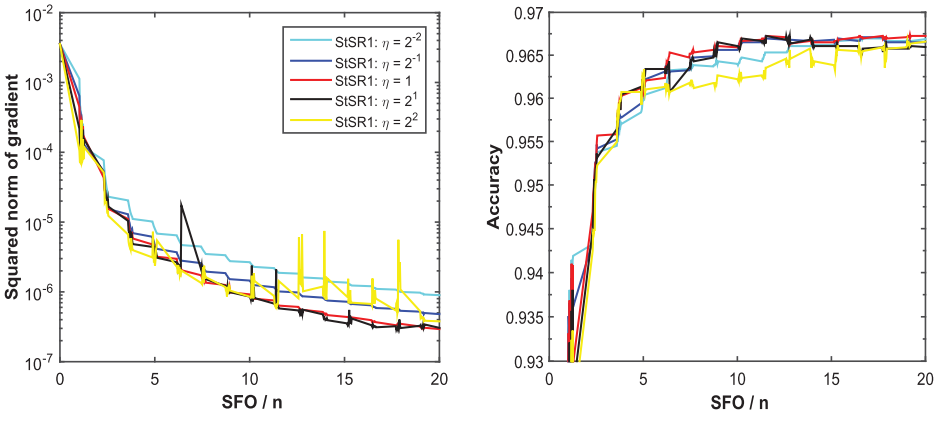

Figure 3. Comparison on rcv1.binary dataset with different stepsize η for StSR1 method. The input parameters are set as $(b, \theta_1, \theta_2) = (0.5n^{1/3}, 2^{-6}, 2^2)$.

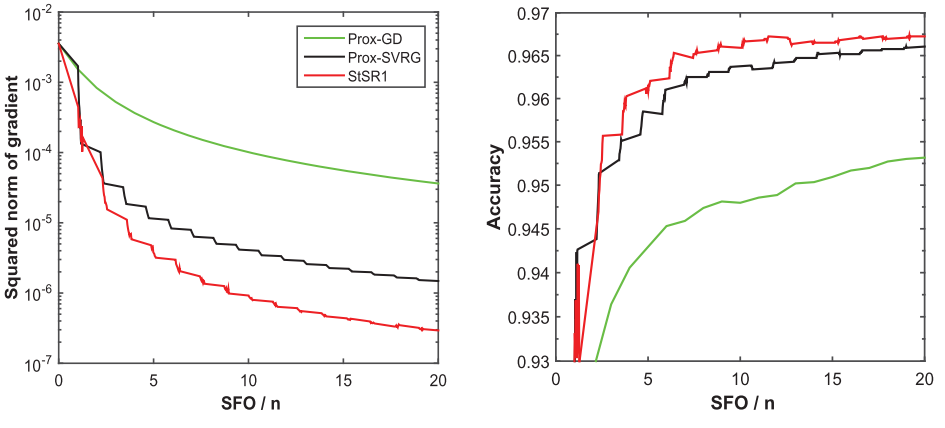

Figure 4. Comparison of StSR1, Prox-GD, and Prox-SVRG on the rcv1.binary dataset. For StSR1 input parameters are set as $(b, \theta_1, \theta_2, \eta) = (0.5n^{1/3}, 2^{-6}, 2^2, 1)$, while $\eta = 10^2$ for Prox-GD and $\eta = 10$ for Prox-SVRG.

5.2. Numerical experiments on w6a and real-sim datasets

In this subsection, we compare the performance of StSR1, Prox-GD and Prox-SVRG on w6a and real-sim dataset. Similar to subsection (5.1), we consider the input parameters $(b, \theta_1, \theta_2, \eta)$. The best performance is achieved with the input $(b, \theta_1, \theta_2, \eta) = (n^{1/3}, 2^{-5}, 2^2, 1)$ for StSR1 on the two datasets, and for Prox-SVRG and Prox-GD the best-tuned stepsizes are $\eta = 10$ and $\eta = 10^2$, respectively. Again, better performance of StSR1 is demonstrated in both Figures 5 and 6.

6. Conclusion

In this paper, we proposed a general framework, SPQN, for stochastic proximal quasi-Newton methods to solve non-convex composition optimization problems. In SPQN, iterates are updated through solving a scaled proximal operator, which is designed based

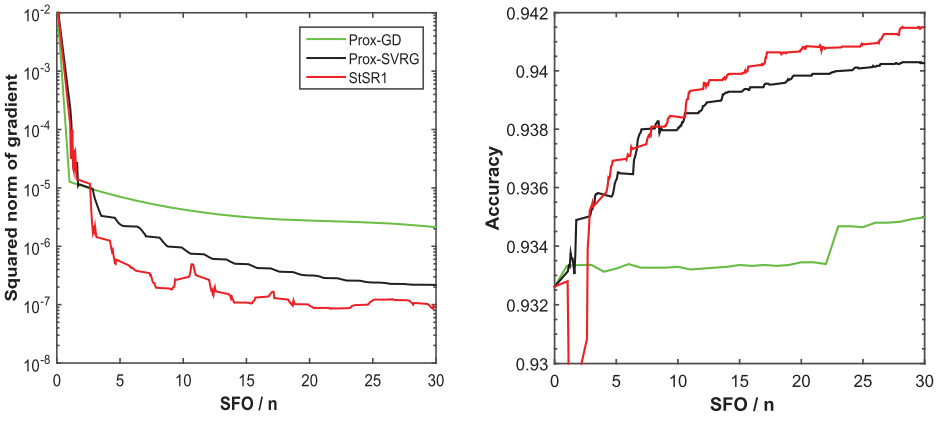

Figure 5. Comparison of StSR1, Prox-GD, and Prox-SVRG on the w6a dataset. For StSR1 input parameters are set as $(b, \theta_1, \theta_2, \eta) = (n^{1/3}, 2^{-5}, 2^2, 1)$, while $\eta = 10^2$ for Prox-GD and $\eta = 10$ for Prox-SVRG.

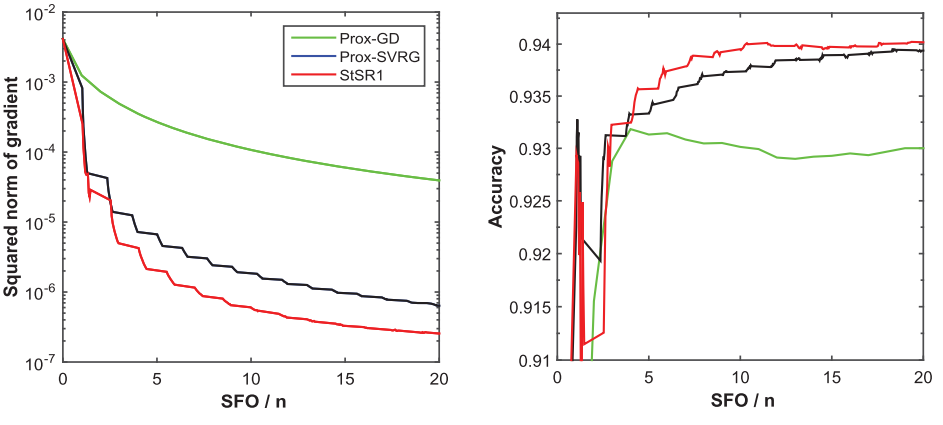

Figure 6. Comparison of StSR1, Prox-GD, and Prox-SVRG on the real-sim dataset. For StSR1 input parameters are set as $(b, \theta_1, \theta_2, \eta) = (n^{1/3}, 2^{-5}, 2^2, 1)$, while $\eta = 10^2$ for Prox-GD and $\eta = 10$ for Prox-SVRG.

on a symmetric positive definite quasi-Newton matrix. We analysed its theoretical properties and proved the global linear convergence rate under CP-PL inequality. Moreover, we proposed a modified self-scaling symmetric rank one (MSSR1) method to update the quasi-Newton matrix, which is incorporated in the framework of SPQN called StSR1 method. In this way, not only the quasi-Newton matrix could satisfy the assumption required to guarantee the convergence of SPQN, but also the proximal subproblem could be solved very efficiently. Finally we reported some numerical results which show the comparable performance of StSR1 method to proximal SVRG and proximal GD methods.

Notes

1. We say a real-valued function f is lower semi-continuous if $\liminf_{x \to x_0} f(x) \geq f(x_0)$.
2. https://www.csie.ntu.edu.tw/cjlin/libsvmtools/datasets/

Disclosure statement

No potential conflict of interest was reported by the authors.

Funding

This research is partially supported by the National Natural Science Foundation of China 11331012, 11301505, 11731013 and 11688101.

References

[1] N. Agarwal, B. Bullins, and E. Hazan., *Second order stochastic optimization for machine learning in linear time*, J. Mach. Learn. Res. 18(116) (2017), pp. 1–40.
[2] Z. Allen-Zhu, *Natasha 2: Faster non-convex optimization than SGD*, preprint (2017). Available at arXiv:1708.08694v2.
[3] Z. Allen-Zhu, *Natasha: Faster stochastic non-convex optimization via strongly non-convex parameter*, preprint (2017). Available at arXiv:1702.00763.
[4] Z. Allen-Zhu and E. Hazan, *Variance reduction for faster non-convex optimization*, International Conference on Machine Learning, New York, NY, 2016, pp. 699–707.
[5] S. Becker and J. Fadili, *A quasi-Newton proximal splitting method*, Advances in Neural Information Processing Systems, Lake Tahoe, 2012, pp. 2618–2626.
[6] A.S. Berahas, R. Bollapragada, and J. Nocedal, *An investigation of Newton-sketch and subsampled Newton methods*, preprint (2017). Available at arXiv:1705.06211.
[7] A. Bordes, L. Bottou, and P. Gallinari, *Sgd-qn careful quasi-Newton stochastic gradient descent*, J. Mach. Learn. Res. 10 (2009), pp. 1737–1754.
[8] R.H. Byrd, G.M. Chin, J. Nocedal, and F. Oztoprak, *A family of second-order methods for convex l_1-regularized optimization*, Math. Program. 159(1–2) (2016), pp. 435–467.
[9] R.H. Byrd, S.L. Hansen, J. Nocedal, and Y. Singer, *A stochastic quasi-Newton method for large-scale optimization*, SIAM J. Optim. 26(2) (2016), pp. 1008–1031.
[10] R.H. Byrd, H.F. Khalfan, and R.B. Schnabel, *Analysis of a symmetric rank-one trust region method*, SIAM J. Optim. 6(4) (1996), pp. 1025–1039.
[11] R.H. Byrd, J. Nocedal, and F. Oztoprak, *An inexact successive quadratic approximation method for convex l_1 regularized optimization*, preprint (2013). Available at arXiv:1309.3529.
[12] A.L. Cauchy, *Méthode générale pour la résolution des systèmes d'équations simultanées*, Comptes rendus des séances de l'Académie des sciences de Paris. 25 (1847), pp. 536–538.
[13] K. Cho, B. Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, *Learning phrase representations using rnn encoderdecoder for statistical machine translation*, preprint (2014). Available at arXiv:1406.1078.
[14] A.R. Conn, N.I.M. Gould, and Ph.L. Toint, *Convergence of quasi-Newton matrices generated by the symmetric rank one update*, Math. Program. 50(2) (1991), pp. 177–195.
[15] F.E. Curtis, *A self-correcting variable-metric algorithm for stochastic optimization*, International Conference on Machine Learning, New York, NY, 2016, pp. 632–641.
[16] A. Defazio, F. Bach, and S.L. Julien, *Saga: A fast incremental gradient method with support for non-strongly convex composite objectives*, Advances in Neural Information Processing Systems, Montreal, 2014, pp. 1646–1654.
[17] J. Friedman, T. Hastie, H. Höfling, and R. Tibshirani, *Pathwise coordinate optimization*, Ann. Appl. Stat. 1(2) (2007), pp. 302–332.
[18] S. Ghadimi, G. Lan, and H. Zhang, *Mini-batch stochastic approximation methods for nonconvex stochastic composite optimization*, Math. Program. 155(1–2) (2016), pp. 267–305.
[19] H. Ghanbari and K. Scheinberg, *Proximal quasi-Newton methods for regularized convex optimization with linear and accelerated sublinear convergence rates*, preprint (2016). Available at arXiv:1607.03081.
[20] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*, MIT Press, Cambridge, MA, 2016.

[21] C.J. Hsieh, M.A. Sustik, I.S. Dhillon, and P. Ravikumar, *Sparse inverse covariance matrix estimation using quadratic approximation*, Advances in Neural Information Processing Systems, Granada, 2011, pp. 2330–2338.
[22] R. Johnson and T. Zhang, *Accelerating stochastic gradient descent using predictive variance reduction*, Advances in Neural Information Processing Systems, Lake Tahoe, 2013, pp. 315–323.
[23] H. Karimi, J. Nutini, and M. Schmidt, *Linear convergence of gradient and proximal-gradient methods under the Polyak-łojasiewicz condition*, Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Springer, Riva del Garda, 2016, pp. 795–811.
[24] S. Karimi and S. Vavasis, *Imro: A proximal quasi-Newton method for solving l_1-regularized least squares problems*, SIAM J. Optim. 27(2) (2017), pp. 583–615.
[25] H.F. Khalfan, R.H. Byrd, and R.B. Schnabel, *A theoretical and experimental study of the symmetric rank one update*, SIAM J. Optim. 3(1), pp. 1–24.
[26] F.R. Khorasani and M.W. Mahoney, *Sub-sampled Newton methods I: Globally convergent algorithms*, preprint (2016). Available at arXiv:1601.04737.
[27] F.R. Khorasani and M.W. Mahoney, *Sub-sampled Newton methods II: Local convergence rates*, preprint (2016). Available at arXiv:1601.04738.
[28] D. Kim, S. Sra, and I.S. Dhillon, *Tackling box-constrained optimization via a new projected quasi-Newton approach*, SIAM J. Sci. Comput. 32(6) (2010), pp. 3548–3563.
[29] J. Lee, Y. Sun, and M. Saunders, *Proximal Newton-type methods for minimizing composite functions*, SIAM. J. Optim. 24(3) (2014), pp. 1420–1443.
[30] H. Lin, J. Mairal, and Z. Harchaoui, *A generic quasi-Newton algorithm for faster gradient-based optimization*, preprint (2016). Available at arXiv:1610.00960.
[31] X. Liu, C. Hsieh, J.D. Lee, and Y. Sun, *An inexact subsampled proximal Newton-type method for large-scale machine learning*, preprint (2017). Available at arXiv:1708.08552.
[32] L. Luo, Z. Chen, Z. Zhang, and W. Li, *A proximal stochastic quasi-Newton algorithm*, preprint (2016). Available at arXiv:1602.00223.
[33] J. Mairal, *Incremental majorization–minimization optimization with application to large-scale machine learning*, SIAM J. Optim. 25(2) (2015), pp. 829–855.
[34] H. Mine and M. Fukushima, *A minimization method for the sum of a convex function and a continuously differentiable function*, J. Optim. Theory Appl. 33(1) (1981), pp. 9–23.
[35] A. Mokhtari, M. Eisen, and A. Ribeiro, *IQN: An incremental quasi-Newton method with local superlinear convergence rate*, preprint (2017). Available at arXiv:1702.00709.
[36] A. Mokhtari and A. Ribeiro, *RES: Regularized stochastic BFGS algorithm*, IEEE Trans. Signal Process. 62(23) (2014), pp. 6089–6104.
[37] P. Moritz, R. Nishihara, and M.I. Jordan, *A linearly-convergent stochastic L-BFGS algorithm*, Artificial Intelligence and Statistics, 2016, pp. 249–258.
[38] J. Nocedal and S.J. Wright, *Numerical Optimization*, 2nd ed., Springer, New York, 2006.
[39] M.R. Osborne and L.P. Sun, *A new approach to symmetric rank one updating*, IMA J. Numer. Anal. 19(4) (1999), pp. 497–507.
[40] M.J.D. Powell, *Algorithms for nonlinear constraints that use lagrangian functions*, Math. Program. 14(1) (1978), pp. 224–248.
[41] S.J. Reddi, A. Hefny, S. Sra, B. Póczos, and A.J. Smola, *Stochastic variance reduction for nonconvex optimization*, International Conference on Machine Learning, New York, NY, 2016, pp. 314–323.
[42] S.J. Reddi, S. Sra, B. Póczos, and A. Smola, *Fast incremental method for nonconvex optimization*, preprint (2016). Available at arXiv:1603.06159.
[43] S.J. Reddi, S. Sra, B. Póczos, and A. Smola, *Fast stochastic methods for nonsmooth nonconvex optimization*, preprint (2016). Available at arXiv:1605.06900.
[44] H. Robbins and S. Monro, *A stochastic approximation method*, Ann. Math. Stat. 22(3) (1951), pp. 400–407.
[45] A. Rodomanov and D. Kropotov, *A superlinearly-convergent proximal Newton-type method for the optimization of finite sums*, International Conference on Machine Learning, New York, NY, 2016, pp. 2597–2605.

[46] M. Schmidt, E. Berg, M.P. Friedlander, and K. Murphy, *Optimizing costly functions with simple constraints: A limited-memory projected quasi-Newton algorithm*, J. Mach. Learn. Res. 5 (2009), pp. 456–463.
[47] M. Schmidt, D. Kim, and S. Sra, *Projected Newton-type methods in machine learning*, in *Optimization for Machine Learning*, S. Sra, S. Nowozin, and S.Wright, eds., MIT Press, Cambridge, MA, 2011.
[48] M. Schmidt, N.L. Roux, and F. Bach, *Minimizing finite sums with the stochastic average gradient*, Math. Program. 160(1–2) (2017), pp. 83–112.
[49] N.N. Schraudolph, J. Yu, and S. Günte, *A stochastic quasi-Newton method for online convex optimization*, J. Mach. Learn. Res. 2 (2007), pp. 436–443.
[50] S. Shalev-Shwartz and T. Zhang, *Proximal stochastic dual coordinate ascent*, preprint (2012). Available at arXiv:1211.2717.
[51] S. Shalev-Shwartz and T. Zhang, *Stochastic dual coordinate ascent methods for regularized loss*, J. Mach. Learn. Res. 14(1) (2017), pp. 567–599.
[52] Z. Shi and R. Liu, *Large scale optimization with proximal stochastic Newton-type gradient descent*, Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Springer, 2015, pp. 691–704.
[53] S. Shwartz and A. Tewari, *Stochastic methods for l_1-regularized loss minimization*, J. Mach. Learn. Res. 12 (2011), pp. 1865–1892.
[54] P. Spellucci, *A modified rank one update which converges q-superlinearly*, Comput. Optim. Appl. 19(4) (2001), pp. 273–296.
[55] L.P. Sun, *Updating the self-scaling symmetric rank one algorithm with limited memory for large-scale unconstrained optimization*, Comput. Optim. Appl. 27(1), pp. 23–29.
[56] R. Tibshirani, *Regression shrinkage and selection via the lasso*, J. R. Stat. Soc. Ser. B 58(1) (1996), pp. 267–288.
[57] X. Wang, S. Ma, D. Goldfarb, and W. Liu, *Stochastic quasi-Newton methods for nonconvex stochastic optimization*, SIAM J. Optim. 27(2) (2017), pp. 927–956.
[58] L. Xiao and T. Zhang, *A proximal stochastic gradient method with progressive variance reduction*, SIAM J. Optim. 24(4) (2014), pp. 2057–2075.
[59] X. Yu and D. Tao, *Variance-reduced proximal stochastic gradient descent for non-convex composite optimization*, preprint (2016). Available at arXiv:1606.00602.
[60] G.X. Yuan, K.W. Chang, C.J. Hsieh, and C.J. Lin, *A comparison of optimization methods and software for large-scale l_1-regularized linear classification*, J. Mach. Learn. Res. 11 (2010), pp. 3183–3234.

Appendices

Appendix 1. Proofs of lemmas in Section 2

Proof of Lemma 2.1.: Recall that $P_{\mathcal{X}}(x, g, \alpha)$ is defined in (8). According to optimality conditions for (9), there exists a subgradient $z \in \partial h(x^+)$ such that

$$\langle g + \alpha B(x^+ - y) + z, y - x^+ \rangle \geq 0, \quad \text{for any } y \in \mathcal{X}.$$

Let $y = x$ and apply the convexity of h at x^+. Then we have

$$\begin{aligned} \langle g, x - x^+ \rangle &\geq \alpha \langle B(x - x^+), x - x^+ \rangle + \langle z, x^+ - x \rangle \\ &\geq \frac{1}{\alpha} \|P_{\mathcal{X}}(x, g, \alpha)\|_B^2 + h(x^+) - h(x). \end{aligned}$$

Then the relationship between $P_{\mathcal{X}}$ and x^+ yields (13). ■

Proof of Lemma 2.2.: Recalling the definitions $\mathcal{D}_h^{\mathcal{X}}(x,g,B,\alpha)$ in (12) and $P_{\mathcal{X}}(x,g,\alpha)$ in (8), we have,

$$\begin{aligned}
\mathcal{D}_h^{\mathcal{X}}(x,g,B,\alpha) &= -2\alpha\left\{\langle g, x^+ - x\rangle + \frac{\alpha}{2}\|x^+ - x\|_B^2 + h(x^+) - h(x)\right\} \\
&= -2\alpha\left\{\left\langle g, -\frac{1}{\alpha}P_{\mathcal{X}}(x,g,\alpha)\right\rangle + \frac{1}{2\alpha}\|P_{\mathcal{X}}(x,g,\alpha)\|_B^2 + h(x^+) - h(x)\right\} \\
&= 2\langle g, P_{\mathcal{X}}(x,g,\alpha)\rangle - \|P_{\mathcal{X}}(x,g,\alpha)\|_B^2 - 2\alpha(h(x^+) - h(x)) \\
&\geq 2\|P_{\mathcal{X}}(x,g,\alpha)\|_B^2 + 2\alpha(h(x^+) - h(x)) - \|P_{\mathcal{X}}(x,g,\alpha)\|_B^2 - 2\alpha(h(x^+) - h(x)) \\
&= \|P_{\mathcal{X}}(x,g,\alpha)\|_B^2,
\end{aligned}$$

where the first equality follows from the fact that the minimizer of big brace is at x^+ and the inequality follows from Lemma 2.1. ■

Proof of Lemma 2.3.: For simplicity, we denote $\mathcal{G}(y) := \langle g, y - x\rangle + (\alpha/2)\|y - x\|_B^2 + h(y) - h(x)$. Since h is convex on $\mathcal{X}$ and B is a symmetric positive definite matrix, so $\mathcal{G}(y)$ is strongly convex on $\mathcal{X}$. Thus, for any closed and convex subset $\mathcal{X}$, the minimizer of $\mathcal{G}(y)$ exists.

For any $\alpha_1 > 0$, there exists $y_1 \in \mathcal{X}$ such that

$$\mathcal{D}_h^{\mathcal{X}}(x,g,B,\alpha_1) = -2\alpha_1\left\{\langle g, y_1 - x\rangle + \frac{\alpha_1}{2}\|y_1 - x\|_B^2 + h(y_1) - h(x)\right\}.$$

Let y_2 satisfies the following equality

$$\alpha_2(y_2 - x) = \alpha_1(y_1 - x),$$

or y_2 can be written as the linear combination of x and y, that is,

$$y_2 = x + \frac{\alpha_1}{\alpha_2}(y_1 - x) = \frac{\alpha_1}{\alpha_2}y_1 + \frac{\alpha_2 - \alpha_1}{\alpha_2}x.$$

For any $0 < \alpha_1 \leq \alpha_2$, y_1 and $x \in \mathcal{X}$, $\mathcal{X}$ is closed convex subset of $\mathbb{R}^d$, so we have $y_2 \in \mathcal{X}$. Applying the convexity of h on $\mathcal{X}$, we can obtain

$$h(y_2) \leq \frac{\alpha_1}{\alpha_2}h(y_1) + \frac{\alpha_2 - \alpha_1}{\alpha_2}h(x), \tag{A1}$$

which is equivalent to

$$\alpha_2[h(y_2) - h(x)] \leq \alpha_1[h(y_1) - h(x)]. \tag{A2}$$

Since $y_2 \in \mathcal{X}$, we have

$$\begin{aligned}
&-2\alpha_2\left\{\langle g, y_2 - x\rangle + \frac{\alpha_2}{2}\|y_2 - x\|_B^2 + h(y_2) - h(x)\right\} \\
&\quad\leq -2\alpha_2 \min_{y\in\mathcal{X}}\left\{\langle g, y - x\rangle + \frac{\alpha_2}{2}\|y - x\|_B^2 + h(y) - h(x)\right\} \\
&\quad= \mathcal{D}_h(x,g,B,\alpha_2).
\end{aligned} \tag{A3}$$

Using the definition of y_2, we obtain,

$$\begin{aligned}
&-2\alpha_2\left\{\langle g, y_2 - x\rangle + \frac{\alpha_2}{2}\|y_2 - x\|_B^2 + h(y_2) - h(x)\right\}\\
&= -2\left\{\langle g, \alpha_2(y_2 - x)\rangle + \frac{\alpha_2^2}{2}\|y_2 - x\|_B^2 + \alpha_2[h(y_2) - h(x)]\right\}\\
&= -2\left\{\langle g, \alpha_1(y_1 - x)\rangle + \frac{\alpha_1^2}{2}\|y_1 - x\|_B^2 + \alpha_2[h(y_2) - h(x)]\right\}\\
&\geq -2\alpha_1\left\{\langle g, y_1 - x\rangle + \frac{\alpha_1}{2}\|y_1 - x\|_B^2 + h(y_1) - h(x)\right\}\\
&= \mathcal{D}_h(x, g, B, \alpha_1),
\end{aligned} \tag{A4}$$

where the inequality follows from (A2). Then together with (A3) and (A4), it yields the result of Lemma 2.3. ■

Appendix 2. Proofs of theorems and corollary in Section 3

In order to analyse the convergence of SPQN, we first give two lemmas.

Lemma A.1: *Given ξ as a random variable, assume that $H(y, \xi)$ is uniformly bounded below and y is independent of ξ, then we have*

$$\mathbb{E}_\xi \min_{y\in\mathcal{X}}\{H(y, \xi)\} \leq \min_{y\in\mathcal{X}} \mathbb{E}_\xi H(y, \xi). \tag{A5}$$

Proof: Since the random variable ξ is independent of y, for any fixed $\hat{y} \in \mathcal{X}$, we have

$$\min_{y\in\mathcal{X}}\{H(y, \xi)\} \leq H(\hat{y}, \xi). \tag{A6}$$

Then taking expectation on both sides of (A6) with respect to ξ, we have

$$\mathbb{E}_\xi \min_{y\in\mathcal{X}}\{H(y, \xi)\} = \int_\xi \min_{y\in\mathcal{X}}\{H(y, \xi)\}\, \mathrm{d}P \leq \int_\xi H(\hat{y}, \xi)\, \mathrm{d}P = \mathbb{E}_\xi H(\hat{y}, \xi).$$

Hence,

$$\mathbb{E}_\xi \min_{y\in\mathcal{X}}\{H(y, \xi)\} \leq \min_{y\in\mathcal{X}} \mathbb{E}_\xi H(y, \xi).$$

■

Let

$$H(y, \xi) = \langle g_k, y - x_k\rangle + \frac{\alpha}{2}\|y - x_k\|_{B_k}^2 + h(y) - h(x_k).$$

Since the random variable ξ comes from the computation of stochastic gradient g_k, Lemma A.1 implies the following result.

Lemma A.2: *Assume that ξ is a random variable and is generated during the computation of g_k. Further assume that $\mathbb{E}[g_k|x_k] = \nabla F(x_k)$ and B_k is a symmetric positive definite matrix and independent of ξ. Then for any $\alpha > 0$, taking conditional expectation on $H(y, \xi)$ with respect to ξ yields*

$$\mathbb{E}\{\mathcal{D}_h^{\mathcal{X}}(x_j^{t+1}, g_j^{t+1}, B_j^{t+1}, \alpha)\} \geq \mathcal{D}_h^{\mathcal{X}}(x_j^{t+1}, \nabla F(x_j^{t+1}), B_j^{t+1}, \alpha). \tag{A7}$$

Proof: Due to the convexity of h on $\mathcal{X}$, $H(y,\xi)$ is uniformly bounded below with respect to y. Notice that y is independent of ξ. Then it follows from Lemma A.1 that

$$\begin{aligned}
&\mathbb{E}\min_{y\in\mathcal{X}}\left\{\langle g_k, y-x_k\rangle+\frac{\alpha}{2}\|y-x_k\|_{B_k}^2+h(y)-h(x_k)\right\}\\
&\quad\le \min_{y\in\mathcal{X}}\mathbb{E}\left\{\langle g_k, y-x_k\rangle+\frac{\alpha}{2}\|y-x_k\|_{B_k}^2+h(y)-h(x_k)\right\}\\
&\quad= \min_{y\in\mathcal{X}}\left\{\langle \nabla F(x_k), y-x_k\rangle+\frac{\alpha}{2}\|y-x_k\|_{B_k}^2+h(y)-h(x_k)\right\}\\
&\quad= -\frac{1}{2\alpha}\mathcal{D}_h^{\mathcal{X}}(x_k,\nabla F(x_k),B_k,\alpha),
\end{aligned}$$

which implies (A7). ■

We now give the proof of Theorem 3.1.

Proof of Theorem 3.1.: By using the L-smoothness of F, we have

$$\begin{aligned}
F(x_{j+1}^{t+1}) &\le F(x_j^{t+1})+\langle\nabla F(x_j^{t+1}), x_{j+1}^{t+1}-x_j^{t+1}\rangle+\frac{L}{2}\|x_{j+1}^{t+1}-x_j^{t+1}\|^2\\
&\le F(x_j^{t+1})+\langle\nabla F(x_j^{t+1})-g_j^{t+1}+g_j^{t+1}, x_{j+1}^{t+1}-x_j^{t+1}\rangle+\frac{L}{2}\|x_{j+1}^{t+1}-x_j^{t+1}\|^2\\
&\le F(x_j^{t+1})+\langle g_j^{t+1}, x_{j+1}^{t+1}-x_j^{t+1}\rangle+\frac{L}{2}\|x_{j+1}^{t+1}-x_j^{t+1}\|^2+\langle\nabla F(x_j^{t+1})-g_j^{t+1}, x_{j+1}^{t+1}-x_j^{t+1}\rangle\\
&\le F(x_j^{t+1})+\min_{y\in\mathcal{X}}\left\{\langle g_j^{t+1}, y-x_j^{t+1}\rangle+\frac{1}{2\eta}\|y-x_j^{t+1}\|_{B_j^{t+1}}^2+h(y)-h(x_j^{t+1})\right\}\\
&\quad+\left(\frac{L}{2}\|x_{j+1}^{t+1}-x_j^{t+1}\|^2-\frac{1}{2\eta}\|x_{j+1}^{t+1}-x_j^{t+1}\|_{B_j^{t+1}}^2\right)\\
&\quad+\langle\nabla F(x_j^{t+1})-g_j^{t+1}, x_{j+1}^{t+1}-x_j^{t+1}\rangle+h(x_j^{t+1})-h(x_{j+1}^{t+1})\\
&\le F(x_j^{t+1})+\min_{y\in\mathcal{X}}\left\{\langle g_j^{t+1}, y-x_j^{t+1}\rangle+\frac{1}{2\eta}\|y-x_j^{t+1}\|_{B_j^{t+1}}^2+h(y)-h(x_j^{t+1})\right\}\\
&\quad+\left(\frac{L}{2}-\frac{\lambda}{2\eta}\right)\|x_{j+1}^{t+1}-x_j^{t+1}\|^2+\langle\nabla F(x_j^{t+1})-g_j^{t+1}, x_{j+1}^{t+1}-x_j^{t+1}\rangle+h(x_j^{t+1})-h(x_{j+1}^{t+1}),
\end{aligned}\tag{A8}$$

where the fourth inequality follows from the definition of x_{j+1}^{t+1}, that is,

$$x_{j+1}^{t+1}=\arg\min_{y\in X}\left\{\langle g_j^{t+1}, y-x_j^{t+1}\rangle+\frac{1}{2\eta}\|y-x_j^{t+1}\|_{B_j^{t+1}}^2+h(y)-h(x_j^{t+1})\right\}.$$

Re-arranging (A8) and shifting the $h(x_{j+1}^{t+1})$ to the left side, we obtain

$$\begin{aligned}
P(x_{j+1}^{t+1}) &\le P(x_j^{t+1})+\min_{y\in\mathcal{X}}\left\{\langle g_j^{t+1}, y-x_j^{t+1}\rangle+\frac{1}{2\eta}\|y-x_j^{t+1}\|_{B_j^{t+1}}^2+h(y)-h(x_j^{t+1})\right\}\\
&\quad+\left(\frac{L}{2}-\frac{\lambda}{2\eta}\right)\|x_{j+1}^{t+1}-x_j^{t+1}\|^2+\langle\nabla F(x_j^{t+1})-g_j^{t+1}, x_{j+1}^{t+1}-x_j^{t+1}\rangle.
\end{aligned}$$

Taking expectation upon the both side of above inequality conditioned on x_j^{t+1} and applying the Lemma A.2 with $\alpha = 1/\eta$, we have

$$\begin{aligned}
\mathbb{E}[P(x_{j+1}^{t+1})] &\le \mathbb{E}[P(x_j^{t+1})] + \left(-\frac{\eta}{2}\right)\mathcal{D}_h^{\mathcal{X}}\left(x_j^{t+1}, \nabla f(x_j^{t+1}), B_j^{t+1}, \frac{1}{\eta}\right) \\
&\quad + \left(\frac{L}{2} - \frac{\lambda}{2\eta}\right)\mathbb{E}\|x_{j+1}^{t+1} - x_j^{t+1}\|^2 + \mathbb{E}\langle \nabla F(x_j^{t+1}) - g_j^{t+1}, x_{j+1}^{t+1} - x_j^{t+1}\rangle \\
&\le \mathbb{E}[P(x_j^{t+1})] - \frac{\eta}{2}\mathcal{D}_h^{\mathcal{X}}\left(x_j^{t+1}, \nabla f(x_j^{t+1}), B_j^{t+1}, \frac{1}{\eta}\right) + \left(\frac{L}{2} - \frac{\lambda}{2\eta}\right)\mathbb{E}\|x_{j+1}^{t+1} - x_j^{t+1}\|^2 \\
&\quad + \frac{1}{2\theta}\mathbb{E}\|\nabla F(x_j^{t+1}) - g_j^{t+1}\|^2 + \frac{\theta}{2}\mathbb{E}\|x_{j+1}^{t+1} - x_j^{t+1}\|^2 \\
&\le \mathbb{E}[P(x_j^{t+1})] - \frac{\eta}{2}\mathcal{D}_h^{\mathcal{X}}\left(x_j^{t+1}, \nabla F(x_j^{t+1}), B_j^{t+1}, \frac{1}{\eta}\right) \\
&\quad + \left(\frac{\theta}{2} + \frac{L}{2} - \frac{\lambda}{2\eta}\right)\mathbb{E}\|x_{j+1}^{t+1} - x_j^{t+1}\|^2 + \frac{1}{2\theta}\mathbb{E}\|\nabla F(x_j^{t+1}) - g_j^{t+1}\|^2, \qquad \text{(A9)}
\end{aligned}$$

where the second inequality uses Cauchy–Schwarz inequality $2\langle a, b\rangle \le (1/\theta)\|a\|^2 + \theta\|b\|^2$.

We now estimate the bound on $\mathbb{E}[\|\nabla F(x_j^{t+1}) - g_j^{t+1}\|^2]$:

$$\begin{aligned}
\mathbb{E}\|\nabla F(x_j^{t+1}) - g_j^{t+1}\|^2 &= \mathbb{E}\left\|\frac{1}{b}\sum_{i\in M_j}[\nabla f_i(x_j^{t+1}) - \nabla f_i(\hat{x}^t) + \nabla F(\hat{x}^t) - \nabla F(x_j^{k+1})]\right\|^2 \\
&\le \frac{2}{b^2}\sum_{i\in M_j}\left\{\mathbb{E}\|\nabla f_i(x_j^{t+1}) - \nabla f_i(\hat{x}^t)\|^2 + \mathbb{E}\|\nabla F(\hat{x}^t) - \nabla F(x_j^{t+1})\|^2\right\} \\
&\le \frac{4L^2}{b}\mathbb{E}\|x_j^{t+1} - \hat{x}^t\|^2. \qquad \text{(A10)}
\end{aligned}$$

Consider the Lyapunov function

$$R_j^{t+1} = \mathbb{E}[P(x_j^{t+1}) + c_j\|x_j^{t+1} - \hat{x}^t\|^2].$$

Note that

$$\begin{aligned}
\mathbb{E}[\|x_{j+1}^{t+1} - \hat{x}^t\|^2] &= \mathbb{E}[\|x_{j+1}^{t+1} - x_j^{t+1} + x_j^{t+1} - \hat{x}^t\|^2] \\
&\le (1+\beta)\mathbb{E}[\|x_{j+1}^{t+1} - x_j^{t+1}\|^2] + \left(1 + \frac{1}{\beta}\right)\mathbb{E}[\|x_j^{t+1} - \hat{x}^t\|^2], \qquad \text{(A11)}
\end{aligned}$$

where the inequality is due to the fact that $\|a+b\|^2 \le (1+\beta)\|a\|^2 + (1+(1/\beta))\|b\|^2$.

Combining the inequalities (A9), (A10), and (A11), we have

$$\begin{aligned}
&\mathbb{E}[P(x_{j+1}^{t+1})] + c_{j+1}\mathbb{E}[\|x_{j+1}^{t+1} - \hat{x}^t\|^2] \\
&\le \left\{\mathbb{E}[P(x_j^{t+1})] + \left(c_{j+1}\left(1 + \frac{1}{\beta}\right) + \frac{2L^2}{b\theta}\right)\mathbb{E}\|x_j^{t+1} - \hat{x}^t\|^2\right\} \\
&\quad - \frac{\eta}{2}\mathcal{D}_h\left(x_j^{t+1}, \nabla F(x_j^{t+1}), B_j^{t+1}, \frac{1}{\eta}\right) \\
&\quad + \left(\frac{\theta}{2} + \frac{L}{2} - \frac{\lambda}{2\eta} + c_{j+1}(1+\beta)\right)\mathbb{E}\|x_{j+1}^{t+1} - x_j^{t+1}\|^2 \\
&\le \{\mathbb{E}[P(x_j^{t+1})] + c_j\mathbb{E}\|x_j^{t+1} - \hat{x}^t\|^2\} - \frac{\eta}{2}\mathcal{D}_h\left(x_j^{t+1}, \nabla F(x_j^{t+1}), B_j^{t+1}, \frac{1}{\eta}\right). \qquad \text{(A12)}
\end{aligned}$$

The second inequality of (A12) follows from the fact that according to the definition of c_j, the sequence of $\{c_j\}_{j=0}^m$ is decreasing and $\eta \le (\underline{\lambda}/(\theta + L + 2c_0(1+\beta))) \le (\underline{\lambda}/(\theta + L + 2c_{j+1}(1+\beta)))$. Consequently, $\theta/2 + L/2 - \underline{\lambda}/2\eta + c_{j+1}(1+\beta) \le 0$ for all $j \in [m]$.

Re-arranging (A12) and use the definition of Lyapunov function R_j^{t+1}, we have

$$\mathcal{D}_h^{\mathcal{X}}\left(x_j^{t+1}, \nabla F(x_j^{t+1}), B_j^{t+1}, \frac{1}{\eta}\right) \le \frac{2}{\eta}\mathbb{E}[R_j^{t+1} - R_{j+1}^{t+1}].$$

Summing up the above inequality, for $j = 0, 1, \ldots, m-1$, we have,

$$\sum_{j=0}^{m-1} \mathcal{D}_h^{\mathcal{X}}\left(x_j^{t+1}, \nabla F(x_j^{t+1}), B_j^{t+1}, \frac{1}{\eta}\right) \le \frac{2}{\eta}\mathbb{E}[R_0^{t+1} - R_m^{t+1}]. \tag{A13}$$

Recalling the update rule that $\hat{x}^{t+1} = x_m^{t+1}$, if we choose $c_m = 0$, we have

$$\mathbb{E}[R_0^{t+1}] = \mathbb{E}[P(x_0^{t+1}) + c_0\|x_0^{t+1} - \hat{x}^t\|^2] = \mathbb{E}[P(\hat{x}^t)],$$

and

$$\mathbb{E}[R_m^{t+1}] = \mathbb{E}[P(x_m^{t+1}) + c_m\|x_m^{t+1} - \hat{x}^t\|^2] = \mathbb{E}[P(x_m^{t+1})] = \mathbb{E}[P(\hat{x}^{t+1})].$$

Thus, if follows from (A13) that

$$\sum_{j=0}^{m-1} \mathcal{D}_h\left(x_j^{t+1}, \nabla F(x_j^{t+1}), B_j^{t+1}, \frac{1}{\eta}\right) \le \frac{2}{\eta}\mathbb{E}[P(\hat{x}^t) - P(\hat{x}^{t+1})].$$

Summing up the above inequality for $t = 0, 1, \ldots, S-1$ and multiplying both sides with $\frac{1}{T}$, we obtain

$$\frac{1}{T}\sum_{t=0}^{S-1}\sum_{j=0}^{m-1} \mathcal{D}_h^{\mathcal{X}}\left(x_j^{t+1}, \nabla F(x_j^{t+1}), B_j^{t+1}, \frac{1}{\eta}\right) \le \frac{2\mathbb{E}[P(x^0) - P(\hat{x}^S)]}{\eta T}. \tag{A14}$$

Notice that $\underline{\lambda}\mathbb{I}_d \preccurlyeq B_j^{t+1} \preccurlyeq \overline{\lambda}\mathbb{I}_d$, then we have $\|y - x_j^{t+1}\|_{B_j^{t+1}}^2 \le \overline{\lambda}\|y - x_j^{t+1}\|^2$. Hence,

$$\begin{aligned}
&\langle \nabla F(x_j^{t+1}), y - x_j^{t+1}\rangle + \frac{1}{2\eta}\|y - x_j^{t+1}\|_{B_j^{t+1}}^2 + h(y) - h(x_j^{t+1}) \\
&\le \langle \nabla F(x_j^{t+1}), y - x_j^{t+1}\rangle + \frac{\overline{\lambda}}{2\eta}\|y - x_j^{t+1}\|^2 + h(y) - h(x_j^{t+1})
\end{aligned} \tag{A15}$$

for all $y \in \mathcal{X}$. Consequently,

$$\begin{aligned}
&\min_{y\in\mathcal{X}}\left\{\langle \nabla F(x_j^{t+1}), y - x_j^{t+1}\rangle + \frac{1}{2\eta}\|y - x_j^{t+1}\|_{B_j^{t+1}}^2 + h(y) - h(x_j^{t+1})\right\} \\
&\le \min_{y\in\mathcal{X}}\left\{\langle \nabla F(x_j^{t+1}), y - x_j^{t+1}\rangle + \frac{\overline{\lambda}}{2\eta}\|y - x_j^{t+1}\|^2 + h(y) - h(x_j^{t+1})\right\}.
\end{aligned}$$

Here, it follows from the simple truth that if $f_1(y) \le f_2(y)$ for all y, then $\min f_1(y) \le \min f_2(y)$. Recalling the definition $D_h^{\mathcal{X}}(x_j^{t+1}, \nabla F(x_j^{t+1}), B_j^{t+1}, \alpha)$ with $\alpha = 1/\eta$, we have

$$\begin{aligned} & D_h^{\mathcal{X}}\left(x_j^{t+1}, \nabla F(x_j^{t+1}), B_j^{t+1}, \frac{1}{\eta}\right) \\ & \quad \ge -\frac{2}{\eta} \min_{y \in \mathcal{X}} \left\{ \langle \nabla F(x_j^{t+1}), y - x_j^{t+1} \rangle + \frac{\overline{\lambda}}{2\eta} \|y - x_j^{t+1}\|^2 + h(y) - h(x_j^{t+1}) \right\} \\ & \quad = \frac{1}{\frac{=}{\lambda}} D_h^{\mathcal{X}}\left(x_j^{t+1}, \nabla F(x_j^{t+1}), \mathbb{I}_d, \frac{\overline{\lambda}}{\eta}\right). \end{aligned}$$

Thus, the above inequality and (A14) imply that

$$\begin{aligned} & \frac{1}{\frac{=}{\lambda} T} \sum_{t=0}^{S-1} \sum_{j=0}^{m-1} D_h^{\mathcal{X}}\left(x_j^{t+1}, \nabla F(x_j^{t+1}), I, \frac{\overline{\lambda}}{\eta}\right) \\ & \quad \le \frac{1}{T} \sum_{t=0}^{S-1} \sum_{j=0}^{m-1} \mathcal{D}_h^{\mathcal{X}}\left(x_j^{t+1}, \nabla F(x_j^{t+1}), B_j^{k+1}, \frac{1}{\eta}\right) \\ & \quad \le \frac{2\mathbb{E}[P(x^0) - P(\hat{x}^S)]}{\eta T}. \end{aligned}$$

Since the output x_a is uniformly chose from $\{\{x_j^{t+1}\}_{j=0}^{m-1}\}_{t=0}^{S-1}$, we obtain

$$\begin{aligned} \mathbb{E}\left[D_h^{\mathcal{X}}\left(x_a, \nabla F(x_a), I, \frac{\overline{\lambda}}{\eta}\right)\right] &= \frac{1}{T} \sum_{t=0}^{S-1} \sum_{j=0}^{m-1} D_h^{\mathcal{X}}\left(x_j^{t+1}, \nabla F(x_j^{t+1}), I, \frac{\overline{\lambda}}{\eta}\right) \\ &\le \frac{2\overline{\lambda}\mathbb{E}[P(x^0) - P(\hat{x}^S)]}{\eta T} \\ &\le \frac{2\overline{\lambda}\mathbb{E}[P(x^0) - P^*]}{\eta T}, \end{aligned}$$

which completes the proof. ■

Proof of Theorem 3.2.: Since $c_j = c_{j+1}(1 + (1/\beta)) + (2L^2/b\theta)$ and $c_m = 0$, we observe that

$$c_0 = \frac{2L^2}{b\theta} \frac{(1 - (1 + \frac{1}{\beta})^m)}{1 - (1 + \frac{1}{\beta})} = \frac{2L^2\beta((1 + \frac{1}{\beta})^m - 1)}{b\theta}.$$

Using $m = \lfloor n^r \rfloor (r > 0)$, and $\beta = n^r$, we have

$$c_0 = \frac{2L^2}{b\theta} n^r \left(\left(1 + \frac{1}{n^r}\right)^{\lfloor n^r \rfloor} - 1\right) \le \frac{2L^2 n^r}{b\theta}(e - 1), \tag{A16}$$

where the inequality uses the well-known fact that $(1 + (1/t))^t < e$ for t ¿ 0.

According to (A16), we have

$$2c_0(1 + \beta) \le 4c_0\beta \le \frac{8L^2 n^{2r}}{b\theta}(e - 1),$$

and then

$$\frac{\underline{\lambda}}{\theta + L + 2c_0(1 + \beta)} \ge \frac{\underline{\lambda}}{\theta + L + \frac{8L^2 n^{2r}}{b\theta}(e - 1)}.$$

We require that $\eta \le (\underline{\lambda}/(\theta + L + (8L^2n^{2r}/b\theta)(e-1))$, applying the specific value of $\theta = Ln^r/\sqrt{b}$, and there exists a constant $\nu > 0$ (independent of n), such that $\eta \le \nu\underline{\lambda}\sqrt{b}/Ln^r$. If we choose the upper bound of η, and applying to the result of Theorem 3.1, this yields the result we need. ■

Proof of Corollary 3.1.: For the algorithm SPQN, we can obtain that the number of $\mathcal{SFO}$ and $\mathcal{CPO}$ is $O(n + bT + n(T/m))$ and $O(T)$, respectively. If $m = n^{1/3}$ means that $r = \frac{1}{3}$ in Theorem 3.2. Because the batchsize is $b = n^{2/3}$, the stepsize can be a constant independent with n, that is $\eta \le \nu\underline{\lambda}/L$. In order to achieve ϵ-approximate solution, we know $(2L\overline{\lambda}/\nu\underline{\lambda})(n^r/\sqrt{b}T) \le \epsilon$, so $T \ge (\kappa_1 n^r/\sqrt{b}\epsilon)$ where $\kappa_1 = (\overline{\lambda}/\underline{\lambda})$. In Algorithm 3.1, we run n SGD iterations to obtain the initial point. Thus if we choose $b = n^{2/3}$ and $r = \frac{1}{3}$, the $\mathcal{SFO}$ and $\mathcal{CPO}$ complexity should be $O(n + \kappa_1 n^{2/3}/\epsilon)$ and $O(\kappa_1/\epsilon)$, respectively. ■

Proof of Theorem 3.3.: At each inner iteration, applying SPQN we obtain the convergence result

$$\mathbb{E}\left[\mathcal{D}_h^{\mathcal{X}}\left(x^{k+1}, \nabla F(x^{k+1}), \frac{1}{\eta}\right)\right] \le \frac{2L\overline{\lambda}}{\nu\underline{\lambda}}\left(\frac{n^r}{\sqrt{b}T}\right)\mathbb{E}[P(x^k) - P^*]. \tag{A17}$$

Since $\eta \le (\underline{\lambda}/(\theta + L + 2c_0(1+\beta)))$, $\eta \le (\underline{\lambda}/L)$, thus

$$\frac{\overline{\lambda}}{\eta} \ge \frac{\overline{\lambda}L}{\underline{\lambda}} \ge L,$$

then it follows from Lemma 2.3 that

$$D_h^{\mathcal{X}}(x_j^{t+1}, \nabla F(x_j^{t+1}), \mathbb{I}_d, L) \le D_h^{\mathcal{X}}\left(x_j^{t+1}, \nabla F(x_j^{t+1}), \mathbb{I}_d, \frac{\overline{\lambda}}{\eta}\right). \tag{A18}$$

Applying the CP-PL inequality at $x = x^{k+1}$, taking expectation on the both sides of (A18), we have

$$2\mu\mathbb{E}[P(x^{k+1} - P^*)] \le \mathbb{E}[\mathcal{D}_h^{\mathcal{X}}(x^{k+1}, \nabla F(x^{k+1}), \mathbb{I}_d, L)]. \tag{A19}$$

Then combining the above inequalities (A17)–(A19), and substituting the specific value of T , we obtain

$$\mathbb{E}[P(x^{k+1} - P^*)] \le \frac{2L\overline{\lambda}}{2\mu\nu\underline{\lambda}}\left(\frac{n^r}{\sqrt{b}T}\right)\mathbb{E}[P(x^k) - P^*] \le \frac{1}{2}\mathbb{E}[P(x^k) - P^*].$$

Thus applying the above inequality recursively yields the result. ■

Proof of Corollary 4.: For GD-SPQN, the $\mathcal{SFO}$ and $\mathcal{CPO}$ complexity are $O(n + K(bT + n(T/m)))$ and $O(KT)$, respectively. In order to achieve ϵ-approximate solution, $K = \log(1/\epsilon)$. So if $T = \lceil (L\overline{\lambda}/(2\mu\nu\underline{\lambda}))(n^r/\sqrt{b})\rceil$, $m = \lfloor n^r \rfloor$, the $\mathcal{SFO}$ complexity is $O((n + \kappa_1\kappa_2((n/\sqrt{b}) + n^r\sqrt{b}))\log(1/\epsilon))$, and $\mathcal{CPO}$ complexity is $O((\kappa_1\kappa_2 n^r/\sqrt{b})\log(1/\epsilon))$ where $\kappa_1 = \overline{\lambda}/\underline{\lambda}$, $\kappa_2 = L/\mu$. ■

Appendix 3. Proofs of theorems in Section 4

Proof of Theorem 4.1.: Let $p = \text{prox}_h^H(x)$, where $H = D + \sigma uu^T$ (σ is $+1$ or -1). Then we obtain

$$\begin{aligned} &\arg\min_y \frac{1}{2}\|y - x\|_H^2 + h(y) \\ \Longleftrightarrow\ &\arg\min_y \frac{1}{2}\|y - x\|_D^2 + \frac{\sigma}{2}\langle u, y - x\rangle^2 + h(y) \\ \xLeftrightarrow{\hat{y} = D^{1/2}y}\ &\arg\min_{\hat{y}} \frac{1}{2}\|\hat{y} - D^{1/2}x\|^2 + \frac{\sigma}{2}\langle D^{-1/2}u, \hat{y} - D^{1/2}x\rangle^2 + h(D^{-1/2}\hat{y}). \end{aligned} \tag{A20}$$

Define $\hat{y}^*$ as the optimal point of the second equation of (A20), which means

$$p = D^{-1/2}\hat{y}^*. \tag{A21}$$

By the first-order optimality conditions for (A22), we obtain

$$0 \in \hat{y}^* - D^{1/2}x + \sigma\langle D^{-1/2}u, \hat{y}^* - D^{1/2}x\rangle D^{-1/2}u + D^{-1/2}\partial h(D^{-1/2}\hat{y}^*). \tag{A22}$$

By defining $\alpha = \langle D^{-1/2}u, \hat{y}^* - D^{1/2}x\rangle$, we can rewrite (A22) as

$$D^{1/2}x - \sigma\alpha D^{-1/2}u - \hat{y}^* \in D^{-1/2}\partial h(D^{-1/2}\hat{y}^*). \tag{A23}$$

Here we use the following fact that

$$v - p \in \partial h(p) \Leftrightarrow p = \mathrm{prox}_h(v), \quad \text{for any } v. \tag{A24}$$

So (A23) is equivalent to

$$\hat{y}^* = \mathrm{prox}_{h\circ D^{-1/2}}(D^{1/2}x - \sigma\alpha D^{-1/2}u).$$

Thus, it follows from (A21) that

$$p = D^{-1/2} \circ \mathrm{prox}_{h\circ D^{-1/2}}(D^{1/2}x - \sigma\alpha D^{-1/2}u),$$

where $\alpha = \langle D^{-1/2}u, D^{1/2}p - D^{1/2}x\rangle$, or equivalently

$$\langle u, x - D^{-1/2} \circ \mathrm{prox}_{h\circ D^{-1/2}}(D^{1/2}x - \sigma\alpha D^{-1/2}u)\rangle + \alpha = 0.$$

■

In Algorithm 4.1, the self-scaling parameter τ plays an important role in preserving the positive definiteness of quasi-Newton matrices. We now give a lemma showing the bound of τ.

Lemma A.3: *Assume that τ is defined as in Algorithm* 4.1 *and $v_j^T s_j > 0$, then we have*

$$\frac{v_j^T s_j}{2v_j^T v_j} < \tau \le \frac{v_j^T s_j}{v_j^T v_j}. \tag{A25}$$

Proof: It is easy to obtain that $v_j^T v_j s_j^T s_j \ge (v_j^T s_j)^2$ by Cauchy–Schwarz inequality, so the definition of τ is reasonable. For simplicity, let $a = s_j^T s_j$, $b = s_j^T v_j$, and $c = v_j^T v_j$. Without causing any confusion, we can omit the subscripts and rewrite τ as

$$\tau = \frac{a}{b} - \sqrt{\left(\frac{a}{b}\right)^2 - \frac{b}{c}}. \tag{A26}$$

To prove that $\tau \le (v_j^T s_j / v_j^T v_j)$, it suffices to prove $\tau \le (b/c)$. Multiplying b/a on the both sides of (A26), we have

$$\tau\frac{b}{a} = 1 - \sqrt{1 - \frac{b^2}{ac}} \le \frac{b^2}{ac}.$$

It is easy to obtain that $\tau \le (b/c)$ and the equality is achieved when $b^2 = ac$. Consequently, we have

$$\tau = \frac{a}{b} - \sqrt{\left(\frac{a}{b}\right)^2 - \frac{b}{c}} = \frac{\frac{b}{c}}{\frac{a}{b} + \sqrt{\left(\frac{a}{b}\right)^2 - \frac{b}{c}}} > \frac{\frac{a}{c}}{2\frac{a}{b}} = \frac{b}{2c}.$$

■

Proof of Theorem 4.2.: From Lemma A.3, we have the results that $\tau \le (v_j^T s_j / v_j^T v_j)$ and $\tau > (v_j^T s_j / 2 v_j^T v_j)$, so

$$\rho = v_j^T (s_j - \tau v_j) \ge 0,$$

and $\rho = 0$ if and only if $v_j^T v_j s_j^T s_j = (v_j^T s_j)^2$. Moreover, we have $(1/2\theta_2) < \tau \le (1/\theta_1)$.

If $\rho \le \epsilon \|s_j - \tau v_j\| \|v_j\|$, then we have $H_{j+1}^{t+1} = \tau \mathbb{I}_d$. In this case, we have $\lambda_{\min}(H_{j+1}^{t+1}) = \lambda_{\max}(H_{j+1}^{t+1}) = \tau$. Otherwise,

$$\begin{aligned}
u_j^T u_j &= \frac{(s_j - \tau v_j)^T (s_j - \tau v_j)}{(s_j - \tau v_j)^T v_j} \\
&\le \frac{(s_j - \tau v_j)^T (s_j - \tau v_j)}{\epsilon \|s_j - \tau v_j\|_2 \|v_j\|_2} \\
&\le \frac{\|s_j - \tau v_j\|_2}{\epsilon \|v_j\|_2} \\
&\le \frac{1}{\epsilon} \sqrt{\frac{s_j^T s_j}{v_j^T v_j} - \frac{2\tau v_j^T s_j}{v_j^T v_j} + \tau^2} \\
&\le \frac{1}{\epsilon} \sqrt{\frac{s_j^T s_j}{v_j^T v_j} - \frac{3}{4} \left(\frac{v_j^T s_j}{v_j^T v_j} \right)^2} \\
&\le \frac{1}{\epsilon} \sqrt{\frac{s_j^T s_j}{v_j^T v_j}} \le \frac{1}{\epsilon} \sqrt{\frac{s_j^T s_j}{v_j^T s_j} \frac{v_j^T s_j}{v_j^T v_j}} \\
&\le \frac{1}{\epsilon} \frac{s_j^T s_j}{v_j^T s_j} \le \frac{1}{\epsilon \theta_1},
\end{aligned} \tag{A27}$$

where the third inequality follows from the bound of τ that $(v_j^T s_j / 2 v_j^T v_j) < \tau \le (v_j^T s_j / v_j^T v_j)$, and the last inequality uses the truth that $v_j^T v_j s_j^T s_j \ge (v_j^T s_j)^2$.

We now estimate the bound of $u_j^T u_j$. First, we have

$$\overline{\lambda} = \lambda_{\max}(H_{j+1}^{t+1}) \le \mathrm{tr}(H_{j+1}^{t+1}) \le \mathrm{tr}(\tau \mathbb{I}_d + u_j u_j^T) \le \tau d + \mathrm{tr}(u_j u_j^T) \le \tau d + u_j^T u_j \le \tau d + \frac{1}{\epsilon \theta_1}.$$

Let $B_{j+1}^{t+1} = (H_{j+1}^{t+1})^{-1}$. Then

$$B_{j+1}^{t+1} = \frac{1}{\tau} \mathbb{I}_d - \frac{u_j u_j^T}{\tau(\tau + u_j^T u_j)},$$

which yields that

$$\lambda_{\max}(B_{j+1}^{t+1}) \le \mathrm{tr}(B_{j+1}^{t+1}) = \mathrm{tr}\left(\frac{1}{\tau} \mathbb{I}_d \right) - \mathrm{tr}\left(\frac{u_j u_j^T}{\tau(\tau + u_j^T u_j)} \right) \le \frac{d}{\tau},$$

and

$$\underline{\lambda} = \lambda_{\min}(H_{j+1}^{t+1}) = \frac{1}{\lambda_{\max}(B_{j+1}^{t+1})} \ge \frac{\tau}{d} \ge \frac{1}{2 d \theta_2}.$$

Therefore, the proof is complete. ■