Journal of Mathematical Extension
Vol. 14, No. 3, (2020), 65-83
ISSN: 1735-8299
URL: http://www.ijmex.com
Original Research Paper

A New Stochastic Limited Memory BFGS Algorithm

M. Momeni
K. N. Toosi University

M. R. Peyghami*
K. N. Toosi University

D. Ataee Tarzanagh
University of Florida

Abstract. In this paper, a new limited memory BFGS is proposed for solving stochastic optimization problems. Since the cost of storing and manipulating H_k is prohibitive in the large scale setting, the L-BFGS algorithms use the strategy of keeping the most recent correction pairs. Besides, in the stochastic regime, due to some noisy information in both gradient vector and Hessian approximation, the second-order model is not an accurate estimation of the function. To overcome this problem, our L-BFGS employs memory in an optimal manner by storing the correction pairs that have the least violation in the secant equation. Under some standard assumptions, the convergence property of the new algorithm is established for strongly convex functions. Numerical results on the problems arising in machine learning show that the new method is competitive and effective in practice.

AMS Subject Classification: 90C53; 90C06
Keywords and Phrases: Limited memory BFGS (L-BFGS), stochastic optimization, secant equation

Received: May 2019; Accepted: July 2019
*Corresponding author

1. Introduction

Using many more parameters to model large-scale data sets is a tendency in machine learning. The larger the scale of the dataset and hence the more the parameters used for modelling the dataset, the larger the scale of the optimization problem. Accordingly, designing efficient algorithms for these large-scale optimization problems is crucial in data science.

A special case that arises frequently in machine learning is the empirical risk minimization problem:

$$\min_{x\in\mathbb{R}^n} F(x) = \frac{1}{T}\sum_{i=1}^{T} f_i(x), \tag{1}$$

where $f_i : \mathbb{R}^n \to \mathbb{R}$ is the loss function corresponding to the i-th sample data, and T denotes the number of observations and is assumed to be extremely large. The large value of T necessitates the use of stochastic approximation algorithms (SA) which are based on small subsets of data. One of the most pioneering research in this field is attributed to Robbins and Monro [28]. This classical SA method mimics the steepest descent gradient method, by using a mini-batch stochastic gradient based on $b \ll T$ instances, i.e., the iteration $x^{k+1} = x^k - \alpha^k \hat{\nabla} F(x^k)$ scheme is employed, where the stochastic gradient is estimated by

$$\hat{\nabla} F(x) := \hat{\nabla}_{\mathcal{S}_g} F(x) = \frac{1}{b}\sum_{i\in\mathcal{S}_g} \nabla f_i(x), \tag{2}$$

where $\mathcal{S}_g \subset \{1, 2, ...T\}$ is a randomly chosen subset of training examples, and b is the cardinality of $\mathcal{S}_g$.

The SA method has been studied extensively in [7, 11, 12, 26, 27, 30, 31], in which the main focus has been the convergence of SA in different settings. The methods for solving convex stochastic optimization problems are of great interest in the literature [1, 2, 8, 11, 12, 13, 14, 15, 18, 19]. Unlike gradient-based methods, which fundamentally employ first-order models, quasi-Newton methods employ second-order models. The use of second-order derivative information makes these methods more ro-

bust. The quasi-Newton update is as follows:

$$x^{k+1} = x^k - \alpha^k H_k \nabla F(x^k), \tag{3}$$

where H_k is an estimation of the exact inverse Hessian $[\nabla^2 F(x^k)]^{-1}$.

There are a number of works in the literature attempting to design stochastic quasi-Newton algorithms for solving large scale setting as (1). These algorithms update the iterates via (3) using the stochastic gradient. Byrd et al. in [5] employed Hessian vector products to incorporate the second-order information by using sample average approximation (SAA) approach. Bordes et al. [3] used a diagonal re-scaling matrix and updated the Hessian at fixed intervals in order to reduce the computational costs.

Schraudolph et al. in [32] developed a BFGS framework for solving (1). The gradient difference in their BFGS update is obtained from two sampled gradients which are extracted from the same sample set.

Mokhtari and Ribeiro [23] makes use of regularized BFGS matrix to solve strongly convex problems. Moreover, they studied an online quasi-Newton method in [24].

In all the above-mentioned studies, the gradient noise is of special importance. To alleviate the gradient noise, Moritz et al. [25] integrated the L-BFGS method of [21] with the variance reduction technique (SVRG) proposed by Johnson and Zhang in [16]. Also, Lucchi et al. in [22] employed SVRG in the L-BFGS method. Byrd et al. [6] incorporated the stochastic regime in the L-BFGS method. They calculate H_k using subsampled Hessian matrices $\hat{\nabla}^2 F(x^k)$, based on the sample $\mathcal{S}_{\mathcal{H}}$, where $\mathcal{S}_{\mathcal{H}}$ is sampled uniformly at random and independently of $\mathcal{S}$. Their strategy for reducing computational costs is similar to that mentioned by Bordes [3]. In fact, they update the Hessian vector products at fixed intervals.

In this paper, we propose a new stochastic quasi-Newton method in the L-BFGS framework. Traditionally, the L-BFGS algorithms keep the most *recent* correction pairs $\{s_i, y_i\}$ [21], and drop the oldest vector pair in the current set of correction pairs and replace it by the new one once the new iterate is performed. However, in the stochastic structure, there

are some noisy information in both gradient vector and Hessian approximation matrix. This makes the secant equation to be violated by the correction pairs. Therefore, the second-order model is not an accurate estimation of the function. To overcome this problem, instead of the recent pairs, we keep the correction pairs in the memory that have the least violation of the secant equation. This helps us to better approximate the Hessian vector products by pairs and estimate the model. The convergence property of the new proposed approach for strongly convex functions is investigated under some standard assumptions. To see the practical performance of the new L-BFGS, numerical results of applying the new technique on some large-scale problems arising in machine learning are reported and compared with some other algorithms in this context. The results show the efficiency and effectiveness of our approach in practice.

The rest of the paper is organized as follows: In Section 2, we derive a new strategy for updating memory in L-BFGS method, and present the new limited memory quasi-Newton algorithm. Section 3 is devoted to establishing the convergence property of the new proposed algorithm for strongly convex functions under some assumptions. In Section 4, numerical experiments are provided that illustrate the practical performance of the new L-BFGS on some machine learning problems. Finally, some concluding remarks are given in Section 5.

2. A Stochastic L-BFGS Method

The BFGS method is one of the most popular quasi-Newton algorithm for minimizing a deterministic function $F(x)$. It is derived by forming the following quadratic model of the objective function at the current iterate x_k:

$$m_k(p) = F_k + \nabla {F_k}^T p + \frac{1}{2} p^T B_k p,$$

where $F_k = F(x_k)$, $\nabla F_k = \nabla F(x_k)$ and B_k is an $n \times n$ symmetric positive definite matrix that approximates the Hessian and is updated

at every iteration. The BFGS algorithm offers to update H_{k+1}, the approximation of the inverse Hessian, uniquely by H_k:

$$(BFGS) \qquad H_{k+1} = (I - \rho_k s_k {y_k}^T) H_k (I - \rho_k y_k {s_k}^T) + \rho_k s_k {s_k}^T, \quad (4)$$

where

$$s_k = x^{k+1} - x^k = \alpha^k p_k, \qquad y_k = \nabla F_{k+1} - \nabla F_k, \quad (5)$$

and

$$\rho_k = \frac{1}{{y_k}^T s_k}.$$

The pair $\{s_k, y_k\}$ is called a correction pair. In the quasi-Newton methods, a reasonable requirement on the approximation of the Hessian matrix is that the gradient of the model should match the gradient of the objective function in the latest two iterations, which is referred to as *secant equation*, i.e.,

$$B_{k+1} s_k = y_k, \quad (6)$$

For large-scale settings, it is necessary to employ a limited memory variant of the BFGS method. In the so-called L-BFGS method, only a certain number of latest correction pairs $\{s_i, y_i\}$ is stored in the memory in order to update H_k by (4).

Our algorithm is in the stochastic structure and uses the L-BFGS framework. Following Byrd et al. [6], we use regular intervals to update Hessian vector products instead of updating them in each iteration. Indeed, the displacement vector s is calculated by the following equation:

$$s_t = \bar{x}_t - \bar{x}_{t-1}, \qquad \bar{x}_t = \sum_{k-L}^{k} x^i, \quad (7)$$

where L is the fixed interval length, x^i is the i-th iteration and $\bar{x}_t$ is the average of the recent L iterations. We also follow the work suggested by Byrd et al. [6] in defining the displacement vector y, to prevent noisy

gradients. Hence, instead of using (5), the vector y is defined by the secant equation (6):

$$y_t = \hat{\nabla}^2_{\mathcal{S}_\mathcal{H}} F(\bar{x}_t) s_t, \tag{8}$$

where $\mathcal{S}_\mathcal{H} \subset \{1, ..., T\}$ is a randomly chosen subset of training examples, and $\hat{\nabla}^2_{\mathcal{S}_\mathcal{H}} F$ is a sub-sampled Hessian matrix defined by:

$$\hat{\nabla}^2_{\mathcal{S}_\mathcal{H}} F(x) = \frac{1}{b_H} \sum_{i \in \mathcal{S}_\mathcal{H}} \nabla^2 f(x), \tag{9}$$

where b_H is the cardinality of $\mathcal{S}_\mathcal{H}$.

The main contribution of our algorithm appears in the technique by which we use the memory in the L-BFGS method. As it was mentioned, since the cost of storing and manipulating H_k is prohibitive when the number of variables is large, the L-BFGS algorithm usually keeps the most *recent* correction pairs $\{s_i, y_i\}$ [21], and replaces the oldest one by the new pair $\{s_k, y_k\}$ right after computing the new iterate. However, in the stochastic regime, both the gradient and Hessian approximations are noisy. Therefore, the secant equation (6) might be violated by the pairs. In this case, keeping just M most recent correction pairs may cause some incompatibility between the model and the objective function. In order to prevent this phenomena, we propose to keep those pairs in the memory that have the least violation in the secant equation.

Let us define the *sec* vector as follows:

$$sec_k = H_{k+1} y_k - s_k = H_{k+1} y_k + \alpha^k p_k.$$

Once the memory for the pairs becomes full, the stored correction pair $\{s, y\}$, whose corresponding *sec* vector norm is large, is replaced by the new correction pair.

Algorithm 3 provides the main framework of our new proposed stochastic limited memory quasi-Newton method. Moreover, Algorithms 1 and 2 are the procedures which calculate vector y via (8) and the matrix-vector products, respectively.

Algorithm 1 Calculation of $y = getSH(x_0, x_1)$

Require: x_1 and x_0
Ensure: y
1: Choose a sample $S_H \in \{1, \ldots, T\}$,
2: Set $y = \hat{\nabla}^2_{\mathcal{S}_\mathcal{H}} F(x_1)(x_1 - x_0)$.

Algorithm 2 Calculation of matrix-vector product; $p = getHg(g, S, Y)$

Require: The vector g, and two matrices S and Y.
Ensure: The matrix-vector product p.
1: Let k be the number of non-zero columns in S, and $q = g$.
2: **for** $i = k$ **to** 1 **do**
3: Set $\rho_i = \frac{1}{y_i^T s_i}$, $\alpha_i = \rho_i s_i^T q$ and $q = q - \alpha_i y_i$.
4: **end for**
5: Set $r = \frac{s_k^T y_k}{y_k^T y_k} q$.
6: **for** $i = 1$ **to** k **do**
7: $\beta_i = \rho_i y_i^T r$, $r = r + s_i(\alpha_i - \beta_i)$.
8: **end for**
9: Set $p = r$.

Since in the first $2L$ iterations, no correction pair $\{s, y\}$ is calculated, the search direction is that of the gradient direction. Whenever the memory reaches its maximum size M, a correction pair $\{s, y\}$ with maximum corresponding $\|sec\|$ is found and is replaced by the newest one. The results show that even with extra computations for finding the correction pair whose corresponding *sec* vector norm is larger, the proposed algorithm performs well and is competitive with some other well-known algorithms both in computing time and optimality gap.

2.1 Convergence analysis

In this section, we investigate the convergence property of Algorithm 3 for strongly convex and twice continuously differentiable functions. The following assumptions are made in our analysis:

Assumption 1 The function $f_i : \mathbb{R}^n \to \mathbb{R}$ is convex and twice continuously differentiable, for all $1 \leqslant i \leqslant T$.

Assumption 2 There exist positive constants λ and Λ, such that

$$\lambda I \preceq \hat{\nabla}^2_{\mathcal{S}_\mathcal{H}} F(x) \preceq \Lambda I, \tag{10}$$

for all $x \in \mathbb{R}^n$ and all nonempty subsets $\mathcal{S}_\mathcal{H} \subset \{1, \ldots, T\}$. This implies that the true objective F satisfies

$$\lambda I \preceq \nabla^2 F(x) \preceq \Lambda I, \qquad \forall x \in \mathbb{R}^n. \tag{11}$$

Algorithm 3 Stochastic limited memory quasi-Newton

Choose a starting point x^0, positive integers M (memory parameter) and L, and step-length sequence $\alpha^k > 0$.
Let S and Y be two $n \times M$ zero matrices.
Set $t = -1$ and $\bar{x}_t = 0$.
Choose a sample $\mathcal{S}_g \subset \{1, \ldots, T\}$.
Set $p_0 = \hat{\nabla} F(x^0)$, $x^1 = x^0 - \alpha^0 p_0$, and $k = 1$.
repeat
  Set $\bar{x}_t = \bar{x}_t + x^k$, and choose a sample $\mathcal{S}_g \subset \{1, \ldots, T\}$,
  if $k \leq 2L$ **then**
    Set $p_k = \hat{\nabla} F(x^k)$.
  else
    Set $p_k = getHg(\hat{\nabla} F(x^k), S, Y)$.
  end if
  if $mod(k, L) = 0$ **then**
    Set $t = t + 1, \bar{x}_t = \frac{\bar{x}_{t-1}}{L}$.
    if $t > 0$ **then**
      Set $\bar{s} = \bar{x}_t - \bar{x}_{t-1}$ and $\bar{y} = getSH(\bar{x}_t, \bar{x}_{t-1})$.
      if $t \leq M$ **then**
        Set $s_t = \bar{s}$, $y_t = \bar{y}$, and $sec_t = getHg(y_t, S, Y) + \alpha^k p_k$.
      else
        Set $i = \texttt{argmax}\{\|sec_j\|, j = 1, \ldots, M\}$,
        for $l = i$ **to** $M - 1$ **do**
          $s_l = s_{l+1}, y_l = y_{l+1}, sec_l = sec_{l+1}$.
        end for
        $s_M = \bar{s}, y_M = \bar{y}, sec_M = getHg(y_M, S, Y) + \alpha^k p_k$.
      end if
    end if
    $\bar{x}_t = 0$.
  end if
  $k = k + 1$.
until convergence

Although the first assumption may appear to be unusual in some settings, such as logistic regression function, it is common in practice to either add an ℓ_2 regularization term or employ other mechanism to ensure that the iterates remain in a region where the function F is strongly convex. These assumptions imply that F has a unique minimizer. From now on, we denote the unique minimizer of F by x^*.

The following lemma provides an upper and a lower bound for the trace and determinant of the ${H_t}^{-1}$ matrices, respectively.

Lemma 2.1.1. *Suppose that Assumptions 1 and 2 hold and $B_t = {H_t}^{-1}$. Then,*

$$\begin{aligned} tr(B_t) &\leqslant (n+M)\Lambda, \\ det(B_t) &\geqslant \frac{\lambda^{n+M}}{(n+M)^M \Lambda^M}. \end{aligned}$$

Proof. Using (5), we have ${s_j}^T y_j = s_j \hat{\nabla}^2_{\mathcal{S}_\mathcal{H}} F(u_j) s_j$. Hence, from Assumption 2, it yields

$$\lambda \parallel s_j \parallel^2 \leqslant {s_j}^T y_j \leqslant \Lambda \parallel s_j \parallel^2.$$

We proceed by induction on t. First, we note that the limited memory quasi-Newton updating formula starts with:

$$\begin{aligned} {B_t}^0 &= \frac{{y_t}^T y_t}{{s_t}^T y_t} I \\ &= \frac{(\hat{\nabla}^2_{\mathcal{S}_\mathcal{H}} F(u_t) s_t)^T \hat{\nabla}^2_{\mathcal{S}_\mathcal{H}} F(u_t) s_t}{{s_t}^T \hat{\nabla}^2_{\mathcal{S}_\mathcal{H}} F(u_t) s_t} I \\ &= \frac{{s_t}^T \hat{\nabla}^2_{\mathcal{S}_\mathcal{H}} F(u_t)^{1/2} \hat{\nabla}^2_{\mathcal{S}_\mathcal{H}} F(u_t) \hat{\nabla}^2_{\mathcal{S}_\mathcal{H}} F(u_t)^{1/2} s_t}{{s_t}^T \hat{\nabla}^2_{\mathcal{S}_\mathcal{H}} F(u_t)^{1/2} \hat{\nabla}^2_{\mathcal{S}_\mathcal{H}} F(u_t)^{1/2} s_t} I, \end{aligned}$$

where the second equality is followed from $y_t = \hat{\nabla}^2_{\mathcal{S}_\mathcal{H}} F(u_t) s_t$, and the last equality is obtained from the fact that $\hat{\nabla}^2_{\mathcal{S}_\mathcal{H}} F(u_t)$ is symmetric and positive definite. Let $z_t = \hat{\nabla}^2_{\mathcal{S}_\mathcal{H}} F(u_t)^{1/2} s_t$. Then,

$${B_t}^0 = \frac{{z_t}^T \hat{\nabla}^2_{\mathcal{S}_\mathcal{H}} F(u_t) z_t}{{z_t}^T z_t} I.$$

From Assumption 2, we have

$$\lambda \leqslant \frac{{z_t}^T \hat{\nabla}^2_{\mathcal{S}_{\mathcal{H}}} F(u_t) z_t}{{z_t}^T z_t} \leqslant \Lambda,$$

which implies that

$$\begin{aligned} tr({B_t}^0) &= n\frac{{y_t}^T y_t}{{s_t}^T y_t} \leqslant n\Lambda, \\ det({B_t}^0) &= \left(\frac{{y_t}^T y_t}{{s_t}^T y_t}\right)^n \geqslant \lambda^n, \end{aligned}$$

for all t. Assume that the induction hypothesis holds. Using Sherman-Morrison-Woodbury formula, we can equivalently write (4) in terms of the Hessian approximation as follows:

$${B_t}^j = {B_t}^{j-1} - \frac{{B_t}^{j-1} s_t {s_t}^T {B_t}^{j-1}}{{s_t}^T {B_t}^{j-1} s_t} + \frac{y_t {y_t}^T}{{y_t}^T s_t}.$$

Now, from the linearity of the trace operator, we can write:

$$\begin{aligned} tr({B_t}^j) &= tr({B_t}^{j-1}) - \frac{\| {B_t}^{j-1} s_t\|^2}{{s_t}^T {B_t}^{j-1} s_t} + \frac{\| y_t\|^2}{{y_t}^T s_t} \\ &\leqslant tr({B_t}^{j-1}) + \frac{\| y_t\|^2}{{y_t}^T s_t} \\ &\leqslant tr({B_t}^{j-1}) + \Lambda \\ &\leqslant (n + M - 1)\Lambda + \Lambda \\ &= (n + M)\Lambda, \end{aligned}$$

where the last inequality follows from the fact that $j \leqslant M$.

In order to find a lower bound for the determinant, we use the property of the determinant, i.e., $det(AB) = det(A)det(B)$, we have:

$$det({B_t}^j) = det\left({B_t}^{j-1}\right) det\left(I - \frac{s_j {s_j}^T {B_t}^{j-1}}{{s_j}^T {B_t}^{j-1} s_j} + \frac{({B_t}^{j-1})^{-1} y_j {y_j}^T}{{y_j}^T s_j}\right).$$

In [9] it is shown that the following identity holds for all u_1, v_1, u_2 and v_2:

$$det(I + u_1 {v_1}^T + u_2 {v_2}^T) = (1 + {u_1}^T v_1)(1 + {u_2}^T v_2) - ({u_1}^T v_2)({v_1}^T u_2).$$

By setting $u_1 = -s_j$, $v_1 = \frac{{B_t}^{j-1}s_j}{{s_j}^T{B_t}^{j-1}s_j}$, $u_2 = \left({B_t}^{j-1}\right)^{-1} y_j$ and $v_2 = \frac{y_j}{{y_j}^T s_j}$, we have:

$$\begin{aligned} det\left({B_t}^j\right) &= det\left({B_t}^{j-1}\right)\frac{{y_j}^T s_j}{{s_j}^T{B_t}^{j-1}s_j} \\ &= det\left({B_t}^{j-1}\right)\frac{{y_j}^T s_j}{\parallel s_j\parallel^2}\frac{\parallel s_j\parallel^2}{{s_j}^T{B_t}^{j-1}s_j} \\ &\geqslant det\left({B_t}^{j-1}\right)\frac{\lambda}{\lambda_{\max}\left({B_t}^{j-1}\right)}, \end{aligned}$$

where the last inequality is followed from the fact that

$${s_j}^T{B_t}^{j-1}s_j \leqslant \lambda_{\max}\left({B_t}^{j-1}\right)\parallel s_j\parallel^2.$$

Now, using the fact that the largest eigenvalue of a positive definite matrix is bounded by its trace, we have:

$$\begin{aligned} det\left({B_t}^j\right) &\geqslant det\left({B_t}^{j-1}\right)\frac{\lambda}{tr\left({B_t}^{j-1}\right)} \\ &\geqslant det\left({B_t}^{j-1}\right)\frac{\lambda}{(n+M)\Lambda} \\ &= \frac{\lambda^{n+M}}{(n+M)^M\Lambda^M}. \end{aligned}$$

This completes the proof of the lemma. □

Lemma 2.1.2. *Suppose that Assumption 1 and 2 hold. Then, there exist constants* $0 < \gamma < \Gamma$ *so that:*

$$\gamma I \preceq H_t \preceq \Gamma I, \tag{12}$$

for all $t \geqslant 1$.

Proof. Using Lemma 2.1.1 and the fact that H_t is positive definite, we obtain:

$$\begin{aligned} \lambda_{\max}(B_t) &\leqslant tr(B_t) \leqslant (n+M)\Lambda, \\ \lambda_{\min}(B_t) &\geqslant \frac{det(B_t)}{\lambda_{\max}(B_t)^{n-1}} \geqslant \frac{\lambda^{n+M}}{(n+M)^{n+M-1}\Lambda^{n+M-1}}. \end{aligned}$$

Now, using $B_t = {H_t}^{-1}$, we have

$$\frac{1}{(n+M)\Lambda} I \preceq H_t \preceq \frac{((n+M)\Lambda)^{n+M-1}}{\lambda^{n+M}} I.$$

By setting $\gamma = \frac{1}{(n+M)\Lambda}$ and $\Gamma = \frac{((n+M)\Lambda)^{n+M-1}}{\lambda^{n+M}}$, the proof is completed. □

The following theorem establishes the global convergence property of Algorithm 3 for the strongly convex functions. The proof is similar to Theorem 2.1.3 in [6].

Theorem 2.1.3. *suppose that Assumptions 1 and 2 hold and there exists positive constant σ such that $E[\| \nabla F(x^k) \|]^2 \leqslant \sigma$. Let x^* be the unique minimizer of F, and suppose that*

$$\alpha^k = \frac{\beta}{k},$$

with $\beta > \frac{1}{2\gamma\lambda}$. Then, for all $k \geqslant 0$, we have

$$E[F(x^k) - F(x^*)] \leqslant \frac{Q}{k},$$

where

$$Q = \max\{F(x^1) - F(x^*), \sigma\}.$$

3. Experimental Results

To validate our approach, we compare the performance of our new proposed algorithm, denoted by "slbfgs", with the stochastic gradient descent (SGD) method [28], the stochastic quasi-Newton method (SQN) [6] and the SVRG-LBFGS [17]. We evaluate these algorithms on three popular machine learning models, including ℓ_2 and ℓ_1-norm regularized logistic regression (LR) problem and ℓ_2-norm regularized linear regression problem (ridge regression). Our experiments show the effectiveness of the algorithm on real-world and synthetic data. All the methods

were implemented in MATLAB using SGDLibrary [2]. The performance of stochastic algorithms is affected not only by the distribution of data but also by the step-size selection strategy [4]. Hence, we consider α as a constant. We set the memory to $M = 10$ for all the limited memory methods in all tests, which is a standard choice for the L-BFGS methods. For all the methods, we display the results for values of the batch size as $b = 10$, and for the quasi-Newton methods, we set the Hessian batch size as $b_H = 200$, and $L = 10$.

It is worth mentioning that in all figures of this section, the horizontal axis is the number of gradient evaluations unless otherwise stated. Besides, in the vertical axis, the "optimality gap" stands for the expected optimality gap which is $E\left[F(x^k) - F(x^*)\right]$), and "cost" denotes the function value.

3.1 Experiments with Synthetic Data

We first test our algorithm on a ℓ_2-norm logistic regression problem. The train/test data were generated randomly via a data generator function with $T = 100$ and $n = 10$. Figure 1 reports the performance of SGD, SQN, and the newly proposed algorithm on this problem. The figure represents the cost function and optimality gap values in terms of the number of gradient evaluations. It is noteworthy that each algorithm requires a different number of evaluations of samples in each epoch. Therefore, it is common to use the number of gradient evaluations to evaluate the algorithms instead of the number of iterations. It can be observed that the new method outperforms others both in computational costs and optimality error.

Figure 2 represents the performance of the considered algorithms on ℓ_1-norm regularized logistic regression problem with $\lambda = 10^{-1}$. As the figure illustrates, the new algorithm outperforms the other algorithms in finding the optimum of the function.

In Figure 3, we investigate the performance of ridge regression problem on the synthetic data. In this experiment, SGD and SVRG-LBFGS

[2] All the codes for the experiments can be downloaded from https://github.com/hiroyuki-kasai/SGDLibrary

make more progress initially but the new algorithm performs well at the end.

3.2 Experiments with RCV1 Data

Reuters Corpus Volume 1 (RCV1) [20] is a dataset which consists of over 800,000 Newswire Stories that have been manually categorized. Figure 4 displays test error and cost function of logistic regression problem on RCV1 data. The figure shows that the new proposed algorithm outperforms the considered algorithms both in optimality gap and costs.

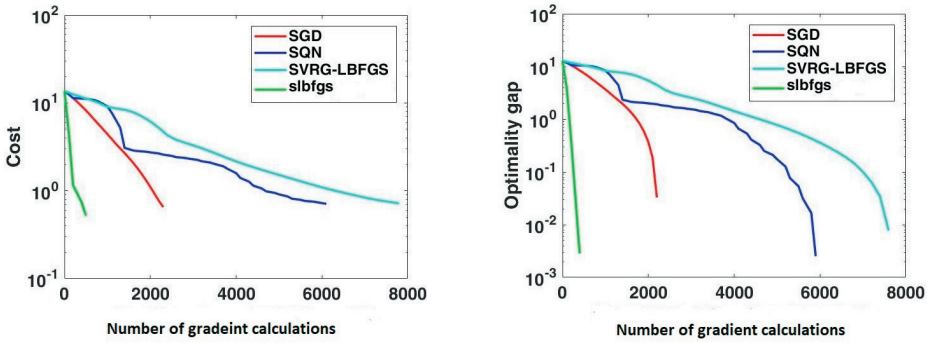

Figure 1. Performance of the considered algorithms on the synthetic data for ℓ_2-norm logistic regression problem

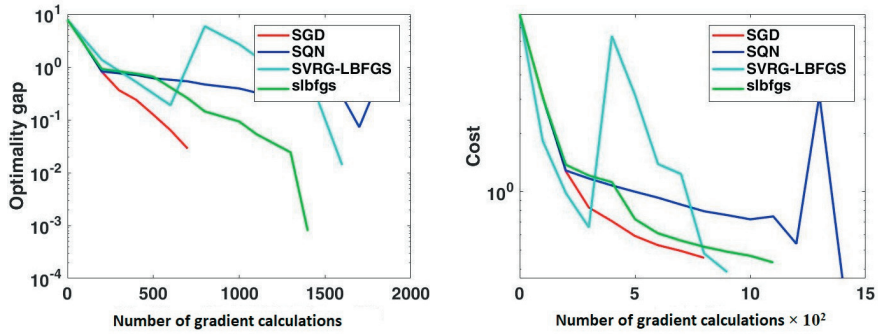

Figure 2. Performance of the considered algorithms on the synthetic data for ℓ_1-norm regularized logistic regression problem

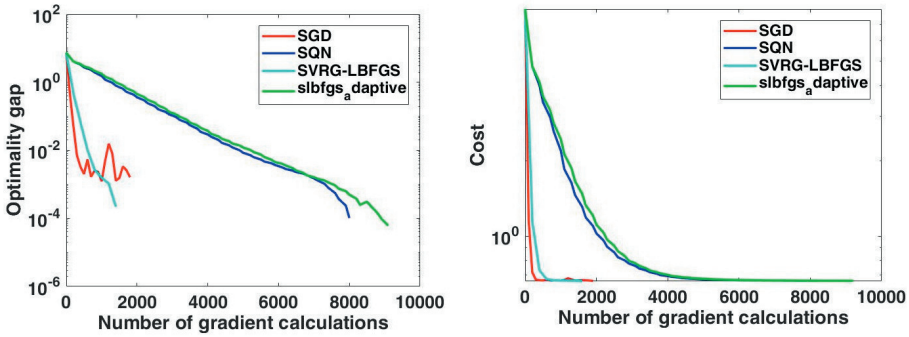

Figure 3. Performance of the considered algorithms on the synthetic data for ℓ_1-norm regularized logistic regression problem

4. Conclusions

In this paper a limited memory BFGS algorithm for solving stochastic optimization problems, mainly arising in the machine learning problems is proposed. In the new algorithm, instead of storing the last pairs of the correction vectors, the ones that provide more accurate curvature of the function are stored. The convergence property of the proposed scheme for the strongly convex functions under standard assumptions is investigated. In order to compare the new approach by some existing methods in the literature, the numerical results of applying the new scheme on some synthetic and real-world data sets are reported and compared.

Acknowledgements
The authors would like to thank the Research Council of K.N. Toosi University of Technology and University of Florida for supporting this research.

References

[1] F. Bach, Adaptivity of averaged stochastic gradient descent to local strong convexity for logistic regression, *J. Mach. Learn. Res.*, 15 (2014), 595–627.

[2] F. Bach and E. Moulines, Non-strongly-convex smooth stochastic approximation with convergence rate $O(1/n)$, *In: NIPS'13 Proceedings of the 26th International Conference on Neural Information Processing Systems*, 1 (2013), 773–781.

[3] A. Bordes, L. Bottou, and P. Gallinari, SGD-QN: Careful quasi-Newton stochastic gradient descent, *J. Mach. Learn. Res.*, 10 (2009), 1737–1754.

[4] L. Bottou, Online algorithm and stochastic approximations, *On-Line Learning in Neural Networks*, Cambridge University Press, 1998.

[5] R. H. Byrd, G. Chin, W. Neveitt, and J. Nocedal, On the use of stochastic hessian information in optimization methods for machine learning, *SIAM J. Optim.*, 21 (3) (2011), 977–995.

[6] 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), 1008–1031.

[7] K. L. Chung, On a stochastic approximation method, *Ann. Math. Stat.*, 25 (3) (1954), 463–483.

[8] A. Defazio, F. Bach, and S. Lacoste-Julien, SAGA: A fast incremental gradient method with support for non-strongly convex composite objectives, *Advances In Neural Information Processing Systems, Montreal, Canada*, (2014).

[9] J. E. Dennis and J. J. Moré, Quasi-Newton methods, motivation and theory, *SIAM Rev.*, 19 (1) (1977), 46–89.

[10] J. C. Duchi, E. Hazan, and Y. Singer, Adaptive subgradient methods for online learning and stochastic optimization, *J. Mach. Learn. Res.*, 12 (2011), 2121–2159.

[11] Y. Ermoliev, Stochastic quasigradient methods and their application to system optimization, *Stochastics*, 9 (1983), 1–36.

[12] A. A. Gaivoronski, Nonstationary stochastic programming problems, *Kibernetika*, 4 (1978), 8–92.

[13] S. Ghadimi and G. Lan, Optimal stochastic approximation algorithms for strongly convex stochastic composite optimization, I: a generic algorithmic framework, *SIAM J. Optim.*, 22 (2012), 1469–1492.

[14] A. Juditsky, A. Nazin, A. B. Tsybakov, and N. Vayatis, Recursive aggregation of estimators via the mirror descent algorithm with average, *Probl. Inform. Transm.*, 41 (4) (2005), 368–384.

[15] A. Juditsky, P. Rigollet, and A. B. Tsybakov, Learning by mirror averaging, *Ann. Stat.*, 36 (2008), 2183–2206.

[16] R. Johnson and T. Zhang, Accelerating stochastic gradient descent using predictive variance reduction, *NIPS'13 Proceedings of the 26th International Conference on Neural Information Processing Systems*, 1 (2013), 315-323.

[17] R. Kolte, M. Erdogdu, and A. Ozgur, Accelerating SVRG via second-order information, *OPT2015*, (2015).

[18] G. Lan, An optimal method for stochastic composite optimization,*Math. Program.*, 133 (1) (2012), 365–397.

[19] G. Lan, A. S. Nemirovski, and A. Shapiro, Validation analysis of mirror descent stochastic approximation method, *Math. Program.*, 134 (2012), 425–458.

[20] D. D. Lewis, Y. Yang, T. G. Rose, and F. Li. Rcv1: A new benchmark collection for text categorization research, *J. Mach. Learn. Res.*, 5 (2004), 361–397.

[21] D. C. Liu and J. Nocedal, On the limited memory BFGS method for large scale optimization, *Math. Program. Ser. B.*, 45 (3) (1989), 503–528.

[22] A. Lucchi, B. McWilliams, and T. Hofmann, A variance reduced stochastic Newton method, *CoRR*, abs/1503.08316 (2015).

[23] A. Mokhtari and A. Ribeiro. RES, Regularized stochastic BFGS algorithm. *IEEE Trans. Signal Process.*, 62 (23) (2014), 6089–6104.

[24] A. Mokhtari and A. Ribeiro, Global convergence of online limited memory BFGS, *J. Mach. Learn. Res.*, 16 (2015), 3151–3181.

[25] P. Moritz, R. Nishihara, and M. I. Jordan, A linearly-convergent stochasitic L-BFGS algorithm, *In Proceedings of the 19th International Conference on Artificial Intelligence and Statistics (AISTATS)*, (2016), 249–258.

[26] B. T. Polyak, New stochastic approximation type procedures, *Automat. Rem. Control*, 51 (1990), 937–946.

[27] B. T. Polyak and A. B. Juditsky, Acceleration of stochastic approximation by averaging, *SIAM J. Control Optim.*, 30 (1992), 838–855.

[28] H. Robbins and S. Monro, A stochastic approximatin method, *Ann. Math. Stat.*, 22 (1951), 400–407.

[29] N. Le Roux, M. Schmidt, and F. Bach, A stochastic gradient method with an exponential convergence rate for strongly convex optimization with finite training sets. *NIPS'12 Proceedings of the 25th International Conference on Neural Information Processing Systems*, 2 (2012), 2663–2671.

[30] A. Ruszczynski and W. Syski, A method of aggregate stochastic subgradients with on-line stepsize rules for convex stochastic programming problems, *Math. Program. Stud.*, 28 (1986), 113–131.

[31] J. Sacks, Asymptotic distribution of stochastic approximation, *Ann. Math. Stat.*, 29 (1958), 373–409.

[32] N. N. Schraudolph, J. Yu, and S. Gunter, A stochastic quasi-Newton method for online convex optimization, *Proceedings of the Eleventh International Conference on Artificial Intelligence and Statistics*, 2 (2007), 436-443.

Mojgan Momeni
Ph.D Candidate in Applied Mathematics
Department of Applied Mathematics
Faculty of Mathematics
Khajeh Nasir Toosi University of Technology
P.O. Box: 16765-3381
Tehran, Iran
E-mail: math.mojgan.momeni@gmail.com

Mohammad Reza Peyghami
Professor in Applied Mathematics (Optimization)
Department of Applied Mathematics
Faculty of Mathematics
Khajeh Nasir Toosi University of Technology
P.O. Box: 16765-3381
Tehran, Iran
E-mail: peyghami@kntu.ac.ir

Davoud Ataee Tarzanagh
Ph.D Candidate in Applied Mathematics
Department of Mathematics
University of Florida
Gainesville, USA
E-mail: tarzanagh@ufl.edu