

# A Stochastic Quasi-Newton Method for Large-Scale Nonconvex Optimization with Applications

H. Chen, H. C. Wu, *Member, IEEE*, S. C. Chan, *Member, IEEE*, W. H. Lam, *Senior Member, IEEE*

**Abstract**—Ensuring the positive definiteness and avoiding ill-conditioning of the Hessian update in the stochastic Broyden–Fletcher–Goldfarb–Shanno (BFGS) method are significant in solving nonconvex problems. This paper proposes a novel stochastic version of damped and regularized BFGS method for addressing the above problems. While the proposed regularized strategy helps to prevent the BFGS matrix from being close to singularity, the new damped parameter further ensures positivity of the product of correction pairs. To alleviate the computational cost of the stochastic LBFGS updates, and to improve its robustness, the curvature information is updated using the averaged iterate at spaced intervals. The effectiveness of the proposed method is evaluated through the logistic regression and Bayesian logistic regression problems in machine learning. Numerical experiments are conducted by using both synthetic dataset and several real datasets. The results show that the proposed method generally outperforms the stochastic damped limited memory BFGS (SdLBFGS) method. In particular, for problems with small sample sizes, our method has shown superior performance and is capable of mitigating ill-conditioned problems. Furthermore, our method is more robust to the variations of the batch size and memory size than the SdLBFGS method.

**Index Terms**—nonconvex optimization, stochastic quasi-Newton method, LBFGS, damped parameter, nonconjugate exponential models, variational inference.

## I. INTRODUCTION

STOCHASTIC optimization algorithms have been extensively studied over decades and can be traced back to the epochal work [22], which have been widely employed in different areas, e.g., machine learning [23]–[25], [52], [53], power systems [51], wireless communication [5]–[7], and bioinformatics [50]. In particular, the classical stochastic approximation (SA) of the exact gradient, also known as stochastic gradient descent (SGD), has been widely applied to these stochastic optimization problems, where the gradient information is employed in finding the search direction. However, in many applications, the exact gradient depends on certain random variables with unknown distributions and thus is difficult to evaluate explicitly. Furthermore, in many applications with extremely massive data samples, the exact gradient of the objective function is rather expensive to compute. In SGD, an unbiased estimator of the gradient is derived using a mini-batch of data points randomly sampled from the full dataset. This substantially reduces the computational cost.

In the theoretical aspect, SGD algorithm has been widely used in the problems with the assumption that the objective function  $f(\cdot)$  is twice continuously differentiable and strongly

convex. In particular, [20] has proposed a robust mirror descent SA algorithm, which is also applicable to general convex objective functions. Recently, there has been an increasing interest in SA based algorithms for solving nonconvex stochastic optimization problems [8], [19], [21]. Specifically, [21] has investigated a stochastic block mirror descent method to solve large scale nonconvex optimization problems with high dimensional optimization variables. [19] has studied a framework of randomized stochastic gradient (RSG) methods by randomly selecting a solution from the previous iterates. The Monte Carlo integration has been adopted for the stochastic search direction [29], [30]. Moreover, the control variate technique [29] is proposed to reduce the variance of the SA.

In the deterministic optimization settings, quasi-Newton or Newton methods can achieve higher accuracy and faster convergence by utilizing the second-order information [8], [12]. For the stochastic regime, stochastic quasi-Newton’s methods (SQN) have been extensively studied in [1]–[3], [8]–[13], [16], [54]. In particular, [16] has developed a stochastic variable-metric method with subsampled gradients. In [2], a SGD-QN scheme has been proposed in which the diagonal elements of the Hessian matrix are approximated to rescale the SGD. Since it only involves scalar computation, the method is quite efficient. It should be noted that direct application of the deterministic quasi-Newton methods brings noisy curvature approximation and thus affects the robustness of the iteration [10]. In [9], the incremental quasi-Newton method (IQN) is proposed to minimize the objective function written in a sum of large amounts of strongly convex functions. It alleviates the high computational cost at each iteration. The main ingredients are as follows. In lieu of random selection of an individual function, incremental methods choose this individual function in a cyclic routine. Thus, it leads to efficient implementation of both the BFGS and iterate updates. The aggregated gradients of all functions are successful in reducing the noise of gradient approximation. Moreover, it satisfies the Dennis-Moré condition. This indicates that IQN method yields local superlinear convergence rate.

Furthermore, the quality of the curvature estimate may be difficult to control in stochastic regime. To alleviate it, [10] has investigated an efficient subsampled Hessian-vector product to estimate the curvature information based on the limited memory BFGS (LBFGS). This method is applied in strongly convex optimization and can avoid doubly evaluating gradients. In [11], the subsampled Hessian matrix scheme is adopted in matrix-vector product form, and the conjugate gradient method is further applied to obtain the search direction. Moreover, the subsampled Hessian matrix is also used as the initial Hessian approximation matrix in the LBFGS method. This is because

The authors are with the Department of Electrical and Electronic Engineering, The University of Hong Kong, Hong Kong, China (e-mail: hmchen@eee.hku.hk; andrewhcwu@eee.hku.hk; scchan@eee.hku.hk; whlam@eee.hku.hk).

the traditional choice contains little curvature information about the problem. In [1], the subsampled Hessian matrix has been adopted to formulate the stochastic block BFGS scheme. The main ingredient is left-multiplying the inverse equation by a randomly generated matrix with few columns. Hence, the computational cost is substantially reduced. In [31], the stochastic variance reduced gradient (SVRG) strategy has been employed to reduce the variance of the stochastic gradient.

It should be noted that the above discussed second-order methods have been proposed for solving convex problems. They cannot be directly applied to nonconvex problems. Moreover, tackling non-convexity and ill-conditioning are two major challenges in stochastic nonconvex optimization problems. To this end, damped BFGS [8] and regularized BFGS [3] have been proposed to deal with the non-convexity and ill-conditioning of the stochastic optimization problem, respectively. In stochastic BFGS methods, the Hessian approximation matrices are ensured to be positive definite in strongly convex optimization problems [14]. However, it is not the case for nonconvex objective functions. In [8], a stochastic damped BFGS based on [13] is proposed to address this issue. However, the BFGS update may still be ill-conditioned if there are insufficient samples. Moreover, the convergence may be significantly affected if the BFGS matrix is close to singularity or even singular. In [3], a regularized stochastic BFGS (RES) method is proposed to improve the numerical condition mentioned above. However, if the problem is nonconvex, the BFGS update may become non-positive definite and hence a descent step may not be guaranteed. Moreover, directly combining the damped scheme [3] and this regularized formulation may still not be able to guarantee positive definiteness of the BFGS update and a descent step. To this end, we propose in this paper a novel stochastic quasi-Newton method, called Sd-REG-LBFGS method, to address the above problems. Our main contributions are as follows:

- **New damped BFGS scheme:** We propose a new stochastic regularized damped BFGS method containing a novel damped parameter and a new gradient difference scheme. The proposed scheme guarantees positive definiteness of the BFGS update and improves the numerical condition of the optimization problem.
- **Choice of Regularization Parameters:** The choice of the regularization parameters for the new regularized gradient difference and damped parameter schemes is crucial to ensure positive definiteness of the BFGS update. We proved that if the chosen regularization parameters satisfy a certain condition (Lemma 1) we have derived, then positive definiteness is guaranteed for the proposed approach.
- **Convergence Analysis:** The convergence property of the proposed method is thoroughly analyzed. In particular, we show that the norm of the updated Hessian approximation matrix is uniformly bounded (see Lemmas 2 and 3), which is a necessary condition for convergence. Furthermore, we showed that with a specified step size, the iteration number  $N$  required to reach a norm of gradient of  $\frac{1}{N} \sum_{k=0}^{N-1} \mathbb{E}(\|\nabla f(x_k)\|^2) < \epsilon$  is at most  $O(\epsilon^{-\frac{1}{1-v}})$ , for  $0.5 < v < 1$ . All the above convergence results

are independent of the convexity assumption. Thus, our proposed method can be applied to nonconvex problems.

For numerical study, the proposed approach is evaluated using a logistic regression, a Bayesian logistic regression and a nonconvex relaxed soft margin support vector machine (SVM)<sup>1</sup>. Experimental results using a synthetic dataset and several real datasets [38], [41]–[45] show that the proposed regularized damped stochastic BFGS method performs better than the conventional damped stochastic BFGS and other algorithms in terms of classification accuracy (ACC) and norm of gradient (NOG), which suggest it converges closer to the stationary point. Moreover, the sensitivity of the proposed algorithm on various algorithmic parameters and the complexity of the proposed algorithm are also studied. Due to page limitation, it is omitted here and interested readers are referred to Sections III and IV of the supplementary material for details.

The rest of the paper is organized as follows: Section II reviews the general formulation of the SQN framework. In Section III, we provide the detail derivation of our proposed algorithm, including the uniform bound on the norm of LBFGS matrix and the convergence results. In Sections IV and V, the effectiveness of the proposed Sd-REG-LBFGS algorithm is demonstrated through solving several machine learning problems, and the numerical experiments are conducted to evaluate the performance of the proposed algorithm with a comparison with conventional algorithms. The conclusion is provided in Section VI.

*Mathematical Notation:* we use  $\|a\|$  to denote the Euclidean norm of vector  $a$  and  $\|A\|$  to denote the matrix norm of a matrix  $A$ . The trace operator of  $A$  is written as  $\text{Tr}(A)$  and the determinant as  $\det A$ . The operator  $\mathbb{E}_{\Xi}(\cdot)$  stands for the expectation taken with respect to random variable  $\Xi$ .  $A \succeq B$  indicates the matrix  $A - B$  is positive semidefinite. The identity matrix with appropriate dimension is signified as  $I$ .

## II. PROBLEM FORMULATION

Consider the following general optimization problem in expectation form:

$$\min_{x \in \mathbb{R}^n} f(x) := \mathbb{E}[F(x, \Xi)], \quad (1)$$

where  $\Xi \in \mathbb{R}^d$  denotes a random variable, and  $F : \mathbb{R}^n \times \mathbb{R}^d \rightarrow \mathbb{R}$  is possibly a nonconvex random function. In many applications, the expectation in (1) is intractable, or the value and gradient of  $f$  are not easily obtained. For example, in machine learning problems, the random variables may contain the input features  $Y$  and the class labels  $Z$ , i.e.  $\Xi = (Y, Z)$ , which may follow some unknown distribution  $P$ , in which inferences are to be made. The training set is assumed to be a collection of independent and identically distributed (i.i.d.) samples  $\xi_i = (y_i, z_i)$  with  $i = 1, \dots, N$ , distributed according to  $P$  via certain observations. The expectation of  $F(x, \Xi)$  in (1) can be approximated by the following empirical average  $\bar{f}(x) = 1/N \sum_{i=1}^N F(x, \xi_i)$ , where  $F(x, \xi_i)$  is the empirical

<sup>1</sup> Due to page limitation, the simulation results for the nonconvex relaxed soft margin SVM is omitted here and interested readers are referred to Section V of the supplementary material.

loss function corresponding to the same  $i$ -th sample  $\xi_i$ . For a large-scale problem where  $N$  is large, this exact empirical gradient may require expensive evaluation of  $F(x, \xi_i)$  for all the samples. In general, stochastic optimization can also be applied to problems where one might be able to access values of the objective function and its gradient from some physical sensor devices in physical simulations. The measured results may be noisy and depend on the unknown  $\xi_n$  every time we attempt to measure  $F(x, \xi)$  or its gradient.

In this paper, we mainly focus in machine learning problems mentioned above. Moreover, the stochastic gradient, denoted as  $g(x, \Xi)$  is an unbiased estimator of  $\nabla f(x)$ , i.e.,  $\mathbb{E}_{\Xi}[g(x, \Xi)] = \nabla f(x)$ , where the expectation is taken with respect to  $\Xi$ . We assume that we can access the gradient via explicit evaluation from the training data (or some physical sensor devices in physical simulations for general stochastic optimization). In addition, we assume that  $f$  is continuously differentiable and the gradient of  $f$  is Lipschitz continuous:

$$\|\nabla f(x) - \nabla f(y)\| \leq L_f \|x - y\|, \quad (2)$$

with Lipschitz constant  $L_f > 0$ .

In classical deterministic quasi-Newton methods, at iteration  $k$ , the update of current iterate is given by:

$$x_{k+1} = x_k - \eta_k B_k^{-1} \nabla f(x_k), \quad (3)$$

where  $B_k$  is an approximation to the Hessians of the objective function  $\nabla^2 f(x_k)$ , since evaluating  $\nabla^2 f(x_k)$  is computationally intensive. Various Hessian approximation methods have been proposed which include, e.g., Broyden, Fletcher, Goldfarb, and Shanno (BFGS); Davidon, Fletcher, and Powell (DFP) and symmetric rank-1 (SR1) updates. In this paper, we mainly focus on the following BFGS update as it is one of the most popular quasi-Newton algorithms:

$$B_{k+1} = B_k + \frac{y_k y_k^T}{s_k^T y_k} - \frac{B_k s_k s_k^T B_k}{s_k^T B_k s_k}, \quad (4)$$

where the correction pairs are  $s_k = x_{k+1} - x_k$  and  $y_k = \nabla f(x_{k+1}) - \nabla f(x_k)$  respectively. It can be shown that (4) satisfies the secant equation, i.e.,  $B_k s_{k+1} = y_k$ . To show that the resultant matrix is positive definite, one can rewrite (4) by letting  $s = s_k, y = y_k, B = B_k, B_{k+1} = B^+$  for notational convenience, which yields:

$$B^+ = \frac{y y^T}{s^T y} + B^{\frac{1}{2}} \left( I - \frac{B^{\frac{1}{2}} s s^T B^{\frac{1}{2}}}{s^T B^{\frac{1}{2}} B^{\frac{1}{2}} s} \right) B^{\frac{1}{2}}. \quad (5)$$

Moreover, it can be shown by induction that with the condition  $s_k^T y_k > 0$ , and an initial positive definite Hessian approximation  $B_0 \succ 0$ ,  $B_k$  is updated recursively and remains positive definite in subsequent iterations. In fact, the condition  $s_k^T y_k > 0$  to preserve the positive definiteness of the Hessian approximation update via (4) is always satisfied for strongly or strictly convex objective functions. This is due to the monotonic gradient mapping property [37]. To be specific, if the objective function  $f$  is strongly or strictly convex, for any  $x, y \in \mathbb{R}^n$ ,  $(\nabla f(x) - \nabla f(y))^T (x - y) > 0$ . Hence, by letting  $x = x_{k+1}$  and  $y = x_k$ , we can see that the condition  $s_k^T y_k > 0$  is satisfied.

To migrate the classical quasi-Newton method to the stochastic regime, the main ingredient is to adopt the stochastic approximation for the exact gradient, which forms the general framework of the SQN method. More precisely, at iteration  $k$ , we subsample a mini-batch  $m_k$  of data so as to compute the stochastic gradient evaluated at the current solution  $x_k$ , which we shall refer to as  $\nabla F(x_k, \xi_{k,i})$  with  $i = 1, \dots, m_k$ . The SA based on this mini-batch estimate can be obtained by the following ensemble average of  $\nabla F(x_k, \xi_{k,i})$  with  $i = 1, \dots, m_k$ :  $\bar{g}(x, \xi_k) = \frac{1}{m_k} \sum_{i=1}^{m_k} \nabla F(x_k, \xi_{k,i})$ . By combining (3) and (4), one gets the desired SQN iterate as follows:

$$x_{k+1} = x_k - \eta_k B_k^{-1} \bar{g}(x_k, \xi_k), \quad (6)$$

where the following stochastic gradient difference is employed in BFGS update (4):

$$y_k = \frac{1}{m_k} \sum_{i=1}^{m_k} \nabla F(x_{k+1}, \xi_{k,i}) - \nabla F(x_k, \xi_{k,i}). \quad (7)$$

**Remark.** It should be noted from the first term in (7) that the gradient of  $F$  at  $x_{k+1}$  is generated using the same subsampling process conducted at current iteration. This implies that at each iteration, the stochastic gradient is evaluated twice. There are two advantages: i). For strongly convex function  $F(\cdot)$ , using (7) guarantees the condition  $s_k^T y_k > 0$ . Moreover, we suggest to adopt the first-order Taylor approximation to reduce the computational complexity, i.e.,  $y_k \approx 1/m_k \sum_{i=1}^{m_k} \nabla^2 F(x_k, \xi_{k,i}) s_k$ , where  $\nabla^2 F(x_k, \xi_{k,i}) s_k$  is a product of the matrix and the vector, which can be obtained with low complexity [10]; ii). It ensures that the BFGS Hessian approximations are uniformly bounded below and above.

### III. THE PROPOSED ALGORITHM

#### A. The Proposed Damped SQN Method

In nonconvex optimization problems, the positivity condition  $s_k^T y_k > 0$  of the correction pairs may not be maintained. This may lead to non-positive definite BFGS matrix. To remedy this problem, [13] has proposed a damped QN method to preserve the positive definiteness of BFGS matrix in nonconvex optimization. Here, we shall extend it to stochastic regime. Specifically,  $y_k$  is modified to  $\bar{y}_k := \theta_k y_k + (1 - \theta_k) B_k s_k$  (thus  $y_k$  in (7) will be modified), where  $\theta_k$  is the damped parameter satisfying:

$$\theta_k = \begin{cases} \frac{0.8 s_k^T B_k s_k}{s_k^T B_k s_k - s_k^T y_k}, & \text{if } s_k^T y_k \leq 0.2 s_k^T B_k s_k, \\ 1, & \text{otherwise.} \end{cases} \quad (8)$$

It can be easily verified that  $B_k \succ 0$  and  $0 < \theta_k \leq 1$  with an initial positive definite Hessian approximation  $B_0 \succ 0$ . Note that when  $\theta_k = 1$ , which is often the case in practice, the BFGS matrix update reduces to the classical formula in (4). For other values of  $\theta_k$ , such modification prevents the determinant of  $B_{k+1}$  from being less than 0.2 of the determinant of  $B_k$  [13]. In addition, since:

$$s_k^T \bar{y}_k = \begin{cases} 0.2 s_k^T B_k s_k, & \text{if } s_k^T y_k \leq 0.2 s_k^T B_k s_k, \\ s_k^T y_k, & \text{otherwise,} \end{cases} \quad (9)$$

it implies that if  $B_k \succ 0$ , then  $s_k^T \tilde{y}_k \geq 0.2s_k^T B_k s_k > 0$ , and the damped quasi-Newton method ensures the positive definiteness of the BFGS update  $B_{k+1}$ .

For nonconvex optimization problems, even the stochastic damped BFGS method guarantees all the subsequent  $B_{k+1}$  obtained via (4) be positive definite, it is possible for the smallest eigenvalue of  $B_{k+1}$  to be arbitrarily close to zero, and hence, the Hessian approximation matrix  $B_k$  will be nearly singular [3]. To remedy the problem, we shall propose a generalized RES scheme for nonconvex optimization using novel damped QN method. We shall first introduce briefly the regularized stochastic quasi-Newton method (RES) for strongly convex optimization problems in [3]. Then, the proposed generalized RES scheme will be described.

Recall  $B_{k+1}$  in (4) is obtained by solving the following semidefinite programming problems:

$$\begin{aligned} \min_Z \quad & \text{Tr}[B_k^{-1}Z] - \log\det[B_k^{-1}Z] - n \\ \text{s.t.} \quad & Zs_k = y_k, Z \succeq 0, \end{aligned} \quad (10)$$

where the optimal solution to (10) is  $Z^* = B_{k+1}$ , obtained by nulling the gradient of the Lagrangian duality function  $\varphi(Z(\nu), \nu) = \inf_{Z \succeq 0} \mathcal{L}(Z, \nu)$  with respect to  $\nu$ , in which  $\mathcal{L}(Z, \nu) = \text{Tr}[B_k^{-1}Z] - \log\det[B_k^{-1}Z] - n + \nu^T(Zs_k - y_k)$ . A simple interpretation to (10) is to minimize the Gaussian differential entropy between the Gaussian distributions  $\mathcal{N}(0, B_k)$  and  $\mathcal{N}(0, Z)$  with the constraint of the secant equation and positive semidefinite solution. For the RES strategy, the following modification of the optimization problem (10) is solved:

$$\begin{aligned} \min_Z \quad & \text{Tr}[B_k^{-1}(Z - \gamma I)] - \log\det[B_k^{-1}(Z - \gamma I)] - n \\ \text{s.t.} \quad & Zs_k = y_k, Z \succeq 0. \end{aligned} \quad (11)$$

By setting  $\tilde{Z} = Z - \gamma I$  and  $\tilde{y}_k = y_k - \gamma s_k$ , the following regularized BFGS update is obtained by using the related Lagrangian duality function:

$$B_{k+1} = B_k + \frac{\tilde{y}_k \tilde{y}_k^T}{s_k^T \tilde{y}_k} - \frac{B_k s_k s_k^T B_k}{s_k^T B_k s_k} + \gamma I. \quad (12)$$

Under the condition  $s_k^T \tilde{y}_k > 0$  with an initial positive semidefinite  $B_0 \succeq 0$ , the subsequent Hessian approximations will have the smallest eigenvalue exceeding a given desired level  $\gamma$ . Comparing (4) and (12), one can see that not only is  $y_k$  being modified to  $\tilde{y}_k$ , an additional regularization term  $\gamma I$  is also introduced to avoid possible ill-conditioning.

However, it can be verified that RES cannot be adopted to the damped QN method for nonconvex optimization problems by simply applying (8) to modify  $y_k$  in (12). We briefly illustrate this below. Consider  $\tilde{y}_k$ , which is the modified version of  $y_k$  by employing (8). It follows that  $s_k^T \tilde{y}_k$  can be calculated as follows:

$$s_k^T \tilde{y}_k = \begin{cases} 0.2s_k^T B_k s_k - \gamma s_k^T s_k, & \text{if } s_k^T y_k \leq 0.2s_k^T B_k s_k, \\ s_k^T y_k - \gamma s_k^T s_k, & \text{otherwise,} \end{cases} \quad (13)$$

Hence, the positivity of  $s_k^T \tilde{y}_k$  cannot be guaranteed. Moreover, even in strongly convex functions  $F(\cdot)$  with convexity parameter  $\underline{m}$  (i.e.,  $\nabla^2 F \succeq \underline{m}I$ ), if the given level  $\gamma$  is chosen to be

greater than  $\underline{m}$ , which results in  $s_k^T \tilde{y}_k < 0$ ,  $B_{k+1}$  can still be near singular or negative positive.

To remedy the problem, we now propose a novel damped SQN method. To start with, the following stochastic gradient difference  $\hat{y}_k$  is proposed to modify  $y_k$ :

$$\hat{y}_k = \bar{\theta}_k y_k + (1 - \bar{\theta}_k)(B_k + \delta I)s_k, \quad (14)$$

where  $\delta$  is a given positive constant that satisfies specific condition (see Lemma 1). Furthermore, we propose to update the damped parameter as follows:

$$\bar{\theta}_k = \begin{cases} \frac{0.8s_k^T(B_k + \delta I)s_k - \gamma s_k^T s_k}{s_k^T(B_k + \delta I)s_k - s_k^T y_k}, & \text{if } s_k^T y_k \leq 0.2s_k^T(B_k + \delta I)s_k + \gamma s_k^T s_k, \\ 1, & \text{otherwise.} \end{cases} \quad (15)$$

Substituting  $\tilde{y}_k := \hat{y}_k - \gamma s_k$  into (12) with the parameter  $\bar{\theta}_k$  defined in (15) yields our proposed Hessian approximation updating scheme:

$$B_{k+1} = B_k + \frac{\tilde{y}_k \tilde{y}_k^T}{s_k^T \tilde{y}_k} - \frac{B_k s_k s_k^T B_k}{s_k^T B_k s_k} + \gamma I. \quad (16)$$

The following lemma shows that by recursively updating  $B_k$  via (16), our proposed method maintains the positive definiteness of the Hessian approximation matrix at each iteration.

*Lemma 1.* For  $\hat{y}_k$  defined in (14) and  $\delta$  is chosen to satisfy  $0.8\delta \geq \gamma$ , then  $0 < \bar{\theta}_k \leq 1$  and  $s_k^T \tilde{y}_k \geq 0.2s_k^T(B_k + \delta I)s_k$ . Moreover, if  $B_k \succ 0$ , then  $B_{k+1}$  generated by the proposed damped BFGS update (16) are positive definite with the smallest eigenvalue exceeding the given desired level  $\gamma$ .

*Proof.* Note from (15) that, if  $s_k^T y_k \leq 0.2s_k^T(B_k + \delta I)s_k + \gamma s_k^T s_k$ , then  $\bar{\theta}_k = 1$ ; for  $s_k^T y_k > 0.2s_k^T(B_k + \delta I)s_k + \gamma s_k^T s_k$ , by substituting the inequality into  $\bar{\theta}_k$ , we get the following inequality:

$$\begin{aligned} \bar{\theta}_k &= \frac{0.8s_k^T(B_k + \delta I)s_k - \gamma s_k^T s_k}{s_k^T(B_k + \delta I)s_k - s_k^T y_k} \\ &\leq \frac{0.8s_k^T(B_k + \delta I)s_k - \gamma s_k^T s_k}{s_k^T(B_k + \delta I)s_k - [0.2s_k^T(B_k + \delta I)s_k + \gamma s_k^T s_k]} = 1. \end{aligned} \quad (17)$$

Moreover, the numerator of (15) satisfies  $0.8s_k^T B_k s_k + (0.8\delta - \gamma)s_k^T s_k \geq 0.8s_k^T B_k s_k > 0$  with the conditions  $0.8\delta \geq \gamma$  and  $B_k \succ 0$ . Similarly from the denominator in (15), we have:

$$s_k^T(B_k + \delta I)s_k - s_k^T y_k \geq 0.8s_k^T(B_k + \delta I)s_k - \gamma s_k^T s_k > 0. \quad (18)$$

Subsequently, both the numerator and denominator of (15) are positive and its maximum value is one, i.e.,  $0 < \bar{\theta}_k \leq 1$ . Moreover, from (14) and (15),  $s_k^T \tilde{y}_k$  can be calculated as follows:

$$\begin{aligned} s_k^T \tilde{y}_k &= s_k^T(B_k + \delta I)s_k - \gamma s_k^T s_k - \bar{\theta}_k[s_k^T(B_k + \delta I)s_k - s_k^T y_k] \\ &= \begin{cases} 0.2s_k^T(B_k + \delta I)s_k, & \text{if } s_k^T y_k \leq 0.2s_k^T(B_k + \delta I)s_k + \gamma s_k^T s_k, \\ s_k^T y_k - \gamma s_k^T s_k, & \text{otherwise.} \end{cases} \end{aligned} \quad (19)$$

From (19), we can see that  $s_k^T \tilde{y}_k \geq 0.2s_k^T(B_k + \delta I)s_k$ . Therefore, if  $B_k$  is positive definite, it follows that  $s_k^T \tilde{y}_k > 0$ . Consequently, as in (5), the first three terms in the right hand

side of the proposed BFGS update scheme (16) is a positive definite matrix.

**Remark.** From the inequality  $s_k^T \tilde{y}_k \geq 0.2s_k^T (B_k + \delta I)s_k$ , we further have  $s_k^T \tilde{y}_k \geq 0.2[\lambda(B_k)_{\min} + \delta]s_k^T s_k$ , where  $\lambda(B_k)_{\min}$  is the smallest eigenvalue of  $B_k$ . Next, we shall extend the proposed BFGS update to a limited memory version.

### B. The Proposed Algorithms for Limited Memory

The limited-memory quasi-Newton method [32], which approximates the Hessian approximation from a limited number of vectors attained from recent iterations, is useful in large scale applications to reduce the large memory storage of the Hessian approximation matrices. As this method requires modest storage and possesses good convergence speed, it is generally considered to be superior to the steepest descent method for deterministic optimization [10]. Interested readers are referred to [14] for more information. In recent years, stochastic limited-memory BFGS (L-BFGS) methods have been studied for strongly convex optimization problems [33] [32] [10]. In this subsection, we propose a stochastic damped and regularized L-BFGS (Sd-REG-LBFGS) method for non-convex optimization problems.

For robustness in implementation and to amortize the cost, one of the strategies is to update the BFGS Hessian approximation at spaced intervals using the average of the iterate points instead of at each iteration [10]. Motivated by this strategy, we compute the correction pairs  $\{s_j, y_j\}$  based on the average of the iterates in the specified interval. The BFGS Hessian approximations are subsequently calculated. In particular, all the modifications are based on our proposed damped BFGS method in (14)-(16). Specifically, we assume that the length of the aforementioned interval of iterations is  $L$ . Suppose we have a memory with size  $M$ . It stores the sequence of correction pairs  $\{s_j, y_j\}$  for  $j = t - (M - 1) - 1, \dots, t - 1$ , where  $t := \frac{k+1}{L}$  and the iteration  $k$  satisfies  $(k + 1) \bmod L = 0$  and  $k \geq M(L - 1) - 1$ . We further define  $s_j$  as the difference of two average iterates with respect to the two most recent disjoint intervals, i.e.,:

$$s_j = \bar{x}_{j+1} - \bar{x}_j, \text{ where } \bar{x}_j = \begin{cases} \frac{1}{L} \sum_{k=(j-1)L}^{jL-1} x_k, & \text{if } j \geq 1, \\ x_0, & \text{if } j = 0. \end{cases} \quad (20)$$

Subsequently, the gradient difference is evaluated at  $\bar{x}_{j+1}$  and  $\bar{x}_j$  as follows:

$$y_j = \frac{1}{m_j} \sum_{l=1}^{m_j} \nabla F(\bar{x}_{j+1}, \xi_{j,l}) - \nabla F(\bar{x}_j, \xi_{j,l}). \quad (21)$$

Recall that we only update BFGS matrix at the end of each interval, to reduce the memory of storing  $B_t$ , we can further approximate it using the L-BFGS method, where a sequence of correction pairs in (20) and (21) are stored. Based on the stochastic damped and regularized BFGS method proposed in

(16), we define a new vector  $\tilde{y}_j := \hat{\theta}_j y_j + (1 - \hat{\theta}_j)(\hat{B}_{j+1}^{(0)} + \delta I)s_j - \gamma s_j$ , with  $\hat{\theta}_j$  given by:

$$\hat{\theta}_j = \begin{cases} \frac{0.8s_j^T (\hat{B}_{j+1}^{(0)} + \delta I)s_j - \gamma s_j^T s_j}{s_j^T (\hat{B}_{j+1}^{(0)} + \delta I)s_j - s_j^T y_j}, & \text{if } s_j^T y_j \leq \gamma s_j^T s_j + 0.2s_j^T (\hat{B}_{j+1}^{(0)} + \delta I)s_j, \\ 1, & \text{otherwise,} \end{cases} \quad (22)$$

where  $\hat{B}_{j+1}^{(0)}$  is an initial estimate of the Hessian matrix and a typical value of  $\hat{B}_{j+1}^{(0)}$  in standard L-BFGS is  $\frac{y_j^T y_j}{s_j^T y_j} I$ . As the denominator  $s_j^T y_j$  may not be positive for nonconvex problems, we propose the following initial value of  $\hat{B}_{j+1}^{(0)}$ :

$$\hat{B}_{j+1}^{(0)} = \tau_{j+1} I, \text{ where } \tau_{j+1} = \max \left\{ \frac{y_j^T y_j}{s_j^T y_j} + \gamma, \beta \right\}, \quad (23)$$

where  $\beta$  is a given positive constant and is also the lower bound on  $\tau_j$ , i.e.,  $\tau_j > \beta$ . Therefore, at the end of the  $t$ -th interval, we define the Sd-REG-LBFGS formula from the past correction pairs  $(s_j, \tilde{y}_j)$  via the following inner iterations:

$$\hat{B}_t^{(i+1)} = \hat{B}_t^{(i)} + \frac{\tilde{y}_j \tilde{y}_j^T}{s_j^T \tilde{y}_j} - \frac{\hat{B}_t^{(i)} s_j s_j^T \hat{B}_t^{(i)}}{s_j^T \hat{B}_t^{(i)} s_j} + \gamma I \quad (24)$$

for  $i = 0, \dots, M - 1$  and  $j = t - (M - 1) + i - 1$ . It follows from Lemma 1 that  $s_j^T \tilde{y}_j \geq 0.2s_j^T (B_{j+1}^{(0)} + \delta I)s_j$ . Therefore, starting with the positive definite matrix  $\hat{B}_t^{(0)}$  given in (23) and a constant  $\delta$  satisfying  $0.8\delta > \gamma$ , the positive definite matrix  $\hat{B}_t = \hat{B}_t^{(M)} \succ \gamma I$  can be updated by the inner iteration of the proposed Sd-REG-LBFGS formula in (24). Furthermore, as the gradient is stochastic and the exact evaluation of the objective function is expensive at each iteration, the Wolfe condition based on the incomplete stochastic gradient may lead to premature condition for convergence or oscillation and prevent the algorithm further progressing. Therefore, we choose the step size to satisfy the well-known condition [22] for the step size choice in stochastic optimization, namely:

$$\sum_{k=1}^{\infty} \eta_k = \infty, \quad \sum_{k=1}^{\infty} \eta_k^2 < \infty. \quad (25)$$

A popular choice is  $\eta_k = \frac{r}{k}$ , for  $r > 0$  [3], [8], [10]. The proposed Sd-REG-LBFGS algorithm is summarized in Algorithm 1.

### C. Convergence Result

For the convergence result of our proposed algorithm, one significant condition is that the norm of the resulting  $\hat{B}_t^{(i+1)}$  from (24) is uniformly bounded above, and uniformly bounded below from zero. Moreover, the following assumption is useful for the derivation of the upper and lower bound:

*Assumption 1* [8]. The random function  $F(x, \Xi)$  is twice continuously differentiable, where the second-order derivative with respect to  $x$  is denoted as  $\nabla^2 F(x, \Xi)$ . Moreover, there exists a positive constant  $\rho$  such that  $\|\nabla^2 F(x, \Xi)\| \leq \rho$ .

Note that the above assumption implies that  $-\rho I \prec \nabla^2 F(x, \Xi) \prec \rho I$ , rather than the strong convexity assumption  $0 \prec \underline{\rho} I \prec \nabla^2 F(x, \Xi) \prec \bar{\rho} I$  in [10] [3]. The following lemma

**Algorithm 1** Sd-REG-LBFGS

**Input:** initial optimization variable  $x_0$ , memory size  $M$ , interval length  $L$ , step length  $\eta_k$  and gradient sample batch size  $m_k$ , choose the constant  $\delta$  and  $\gamma$  satisfying  $0.8\delta > \gamma$

- 1: Set  $t = 0$  and generate  $m_0$  samples  $\{\xi_{0,l}\}_{l=1}^{m_0}$
- 2: **for**  $k = 0, 1, \dots$  **do**
- 3: Randomly choose  $m_k$  samples  $\xi_k = \{\xi_{k,1}, \dots, \xi_{k,m_k}\}$
- 4: Calculate stochastic gradient  $\bar{g}(x_k, \xi_k) = \frac{1}{m_k} \sum_{l=1}^{m_k} \nabla F(x_k, \xi_{k,l})$ ,
- 5: **if**  $t < 2$  **then**
- 6:  $x_{k+1} = x_k - \eta_k \bar{g}(x_k, \xi_k)$
- 7: **else**
- 8:  $x_{k+1} = x_k - \eta_k \hat{B}_t^{-1} \cdot \bar{g}(x_k, \xi_k)$
- 9: **end if**
- 10: **if**  $(k+1) \bmod L = 0$  **then**
- 11: Calculate and store the correction pairs:  $s_t$  and  $y_t$  according to (20) and (21) respectively
- 12: Set  $t = t + 1$
- 13: Generate  $m_t$  samples  $\{\xi_{t,l}\}_{l=1}^{m_t}$
- 14: **if**  $t > 1$  **then**
- 15: Set  $\tilde{M} = \min\{t, M\}$ , draw the sequence of correction pairs  $\{s_j, y_j\}_{j=t-\tilde{M}}^{t-1}$  from the memory.
- 16: Set the initial matrix  $\hat{B}_t^{(0)} = \tau_t I$ , where  $\tau_t = \max\left\{\frac{y_{t-1}^T y_{t-1}}{s_{t-1}^T s_{t-1}} + \gamma, \beta\right\}$
- 17: **for**  $i = 0, \dots, \tilde{M} - 1$  **do**
- 18: Set  $j = t - \tilde{M} + i$  and apply Sd-REG-LBFGS formula according to (24)
- 19: **end for**
- 20: Set  $\hat{B}_t = \hat{B}_t^{(\tilde{M})}$ .
- 21: **end if**
- 22: **end if**
- 23: **end for**

shows that the norm of the matrix  $\hat{B}_t^{\tilde{M}}$  generated by the Sd-REG-LBFGS formula (24) is uniformly bounded above.

*Lemma 2.* Given the positive definite matrix  $\hat{B}_t^{(0)}$  defined by (23), suppose  $\hat{B}_t^{(i+1)}$  is updated through L-BFGS computation step in the  $t$ -th interval of Algorithm 1, then with Assumption 1, the norm of  $\hat{B}_t^{(\tilde{M})}$  is bounded above, i.e.,

$$\left\| \hat{B}_t^{(\tilde{M})} \right\| \leq Q_U, \quad (26)$$

where  $Q_U = \beta + \rho + \gamma + \tilde{M}(Q + 5\rho + \gamma)$ ,  $\tilde{M} = \min\{t, M\}$  and  $Q$  is defined as follows:

$$Q = \max \left\{ \frac{5(\rho + \gamma)^2}{\beta + \delta} + 5(\beta + \delta), \frac{5(\rho + \gamma)^2}{\beta + \rho + \gamma + \delta} + \frac{5(\rho + \gamma)^2}{5(\beta + \rho + \gamma + \delta)} \right\}. \quad (27)$$

*Proof.* Recall from the Sd-REG-LBFGS formula that according to Lemma 1, each generated matrix satisfies  $\hat{B}_t^{(i+1)} \succ \gamma I$ . Note from the third term on the right hand side in (24) that the matrix term  $\frac{\hat{B}_t^{(i)} s_j s_j^T \hat{B}_t^{(i)}}{s_j^T \hat{B}_t^{(i)} s_j}$  is positive definite. Therefore, we

have:

$$\hat{B}_t^{(i+1)} \preceq \hat{B}_t^{(i)} + \frac{\tilde{y}_j \tilde{y}_j^T}{s_j^T \tilde{y}_j} + \gamma I. \quad (28)$$

Taking matrix norm on both sides and using triangle inequality of norm leads to:

$$\begin{aligned} \left\| \hat{B}_t^{(i+1)} \right\| &\leq \left\| \hat{B}_t^{(i)} + \frac{\tilde{y}_j \tilde{y}_j^T}{s_j^T \tilde{y}_j} + \gamma I \right\| \leq \left\| \hat{B}_t^{(i)} \right\| + \left\| \frac{\tilde{y}_j \tilde{y}_j^T}{s_j^T \tilde{y}_j} \right\| + \gamma \\ &= \left\| \hat{B}_t^{(i)} \right\| + \frac{\tilde{y}_j^T \tilde{y}_j}{s_j^T \tilde{y}_j} + \gamma, \end{aligned} \quad (29)$$

from the definition  $\tilde{y}_j := \hat{\theta}_j y_j + (1 - \hat{\theta}_j)(\hat{B}_{j+1}^{(0)} + \delta I)s_j - \gamma s_j$  with  $\hat{\theta}_j$  given in (22), it follows from Lemma 1 that inequalities  $s_j^T \tilde{y}_j \geq 0.2s_j^T (B_{j+1}^{(0)} + \delta I)s_j > 0$  hold. This yields:

$$\begin{aligned} \frac{\tilde{y}_j^T \tilde{y}_j}{s_j^T \tilde{y}_j} &\leq \frac{\left\| \hat{\theta}_j y_j + (1 - \hat{\theta}_j)(\hat{B}_{j+1}^{(0)} + \delta I)s_j - \gamma s_j \right\|^2}{0.2s_j^T (B_{j+1}^{(0)} + \delta I)s_j} \\ &= \frac{1}{0.2s_j^T (B_{j+1}^{(0)} + \delta I)s_j} \{ \hat{\theta}_j^2 y_j^T y_j + (1 - \hat{\theta}_j)^2 \\ &\quad \cdot s_j^T (\hat{B}_{j+1}^{(0)} + \delta I)^2 s_j + 2\hat{\theta}_j(1 - \hat{\theta}_j) \\ &\quad \cdot y_j^T (\hat{B}_{j+1}^{(0)} + \delta I)s_j + \gamma^2 s_j^T s_j \\ &\quad - 2\gamma s_j^T [\hat{\theta}_j y_j + (1 - \hat{\theta}_j)(\hat{B}_{j+1}^{(0)} + \delta I)s_j] \}. \end{aligned} \quad (30)$$

From the definition  $y_j = \frac{1}{m_j} \sum_{l=1}^{m_j} \nabla F(\bar{x}_{j+1}, \xi_{j,l}) - \nabla F(\bar{x}_j, \xi_{j,l})$ , and using the first-order Taylor approximation at  $\bar{x}_j$ , we have  $y_j = \frac{1}{m_j} \sum_{l=1}^{m_j} \nabla^2 F(\bar{x}_j + \vartheta s_j, \xi_{j,l}) s_j$ , where  $0 < \vartheta < 1$ . Thus,  $y_j^T y_j = \frac{1}{m_j^2} s_j^T \{ \sum_{l=1}^{m_j} \sum_{r=1}^{m_j} \nabla^2 F(\bar{x}_j + \vartheta s_j, \xi_{j,r}) \nabla^2 F(\bar{x}_j + \vartheta s_j, \xi_{j,l}) \} s_j$ . With Assumption 1 that  $\|\nabla^2 F(x, \xi)\| \leq \rho$ , which implies  $-\rho I \prec \nabla^2 F(x, \xi) \prec \rho I$ . We further have  $y_j^T y_j \leq \rho^2 s_j^T s_j$ . Next, we consider the product  $y_j^T s_j$ . Since  $y_j^T s_j = \frac{1}{m_j} \sum_{l=1}^{m_j} s_j^T \nabla^2 F(\bar{x}_j + \vartheta s_j, \xi_{j,l}) s_j$ , it follows that  $-\rho s_j^T s_j \leq y_j^T s_j \leq \rho s_j^T s_j$ . Substituting the above inequality into (30), with  $\hat{B}_{j+1}^{(0)} = \tau_{j+1} I$ , we get:

$$\begin{aligned} \frac{\tilde{y}_j^T \tilde{y}_j}{s_j^T \tilde{y}_j} &\leq \frac{1}{0.2(\tau_{j+1} + \delta)} \{ \hat{\theta}_j^2 \rho^2 + (1 - \hat{\theta}_j)^2 (\tau_{j+1} + \delta)^2 \\ &\quad + 2\rho \hat{\theta}_j (1 - \hat{\theta}_j) (\tau_{j+1} + \delta) + \gamma^2 \\ &\quad + 2\gamma \hat{\theta}_j \rho - 2(1 - \hat{\theta}_j) (\tau_{j+1} + \delta) \gamma \} \\ &= \frac{5(\hat{\theta}_j \rho + \gamma)^2}{\tau_{j+1} + \delta} + 5(1 - \hat{\theta}_j)^2 (\tau_{j+1} + \delta) \\ &\quad + 10\hat{\theta}_j (1 - \hat{\theta}_j) \rho - 10\gamma (1 - \hat{\theta}_j). \end{aligned} \quad (31)$$

By using  $\tau_{j+1} = \max\left\{\frac{y_j^T y_j}{s_j^T y_j} + \gamma, \beta\right\}$ , we have  $\beta + \delta \leq \tau_{j+1} + \delta \leq \beta + \rho + \gamma + \delta$ . Furthermore,  $10\hat{\theta}_j (1 - \hat{\theta}_j) \rho \leq 5\rho(1 - \hat{\theta}_j^2)$  holds true as  $0 < \hat{\theta}_j \leq 1$ . By using the property of the function  $\varphi(x) = ax + \frac{b}{x}$ ,  $a > 0, b > 0$ , we obtain the following result:

$$\begin{aligned} \frac{\tilde{y}_j^T \tilde{y}_j}{s_j^T \tilde{y}_j} &\leq Q + 10\hat{\theta}_j (1 - \hat{\theta}_j) \rho - 10\gamma (1 - \hat{\theta}_j) \\ &\leq Q + 5\rho(1 - \hat{\theta}_j^2) - 10\gamma (1 - \hat{\theta}_j) \leq Q + 5\rho, \end{aligned} \quad (32)$$

where  $Q$  is defined in (27). Therefore, by substituting the results in (32) into (29), one gets  $\|\hat{B}_t^{(i+1)}\| \leq \|\hat{B}_t^{(i)}\| + Q + 5\rho + \gamma$ . By induction, we then obtain the desired result:

$$\|\hat{B}_t^{(\bar{M})}\| \leq \|\hat{B}_t^{(0)}\| + \bar{M}(Q + 5\rho + \gamma) \leq \beta + \rho + \gamma + \bar{M}(Q + 5\rho + \gamma). \quad (33)$$

Thus, we have proved the upper bound on the norm of the matrix  $\hat{B}_t^{(\bar{M})}$ , the next lemma gives for a more accurate lower bound rather than just  $\hat{B}_t^{(\bar{M})} \succeq \gamma I$ .

*Lemma 3.* Given the initial positive definite matrix  $\hat{B}_t^{(0)}$  defined by (23), and suppose  $\hat{B}_t^{(i+1)}$  is updated via L-BFGS step of Algorithm 1, then with Assumption 1, all eigenvalues of  $\hat{B}_t^{(\bar{M})}$  satisfies

$$\lambda(\hat{B}_t^{(\bar{M})}) \geq Q_L, \quad (34)$$

where  $Q_L = \max\{\tilde{Q}^{-1}, \gamma^{-1}\}$  and

$$\tilde{Q} = \frac{w^{2\bar{M}} - 1}{Q + 5\rho + 2\sqrt{0.2(Q + 5\rho)(\beta + \delta)}} + \beta^{-1}w^{2\bar{M}}, \quad (35)$$

with  $w := \sqrt{\frac{Q + 5\rho}{0.2(\beta + \delta)}} + 1$ .

*Proof.* From (24), we have:

$$\hat{B}_t^{(i+1)} \succeq \hat{B}_t^{(i)} + \frac{\tilde{y}_j \tilde{y}_j^T}{s_j^T \tilde{y}_j} - \frac{\hat{B}_t^{(i)} s_j s_j^T \hat{B}_t^{(i)}}{s_j^T \hat{B}_t^{(i)} s_j}. \quad (36)$$

Since both sides of the inequality (36) are positive definite matrices, taking matrix inversion and using the Sherman–Morrison–Woodbury formula yields:

$$\begin{aligned} \hat{H}_t^{(i+1)} &\preceq \left( I - \frac{s_j \tilde{y}_j^T}{s_j^T \tilde{y}_j} \right) \hat{H}_t^{(i)} \left( I - \frac{\tilde{y}_j s_j^T}{s_j^T \tilde{y}_j} \right) + \frac{s_j s_j^T}{s_j^T \tilde{y}_j} \\ &= \hat{H}_t^{(i)} - \frac{1}{s_j^T \tilde{y}_j} (s_j \tilde{y}_j^T \hat{H}_t^{(i)} + \hat{H}_t^{(i)} \tilde{y}_j s_j^T) + \frac{\tilde{y}_j^T \hat{H}_t^{(i)} \tilde{y}_j}{(s_j^T \tilde{y}_j)^2} \\ &\quad \cdot s_j s_j^T + \frac{s_j s_j^T}{s_j^T \tilde{y}_j}, \end{aligned} \quad (37)$$

where  $\hat{H}_t^{(i)}$  is the inverse matrix of  $\hat{B}_t^{(i)}$ , i.e.,  $\hat{H}_t^{(i)} := \hat{B}_t^{(i)^{-1}}$ . By taking the matrix norm on both sides of (37) and using the triangle inequality, we get:

$$\begin{aligned} \|\hat{H}_t^{(i+1)}\| &\leq \|\hat{H}_t^{(i)}\| + \frac{2\|\hat{H}_t^{(i)}\| \cdot \|s_j\| \cdot \|\tilde{y}_j\|}{s_j^T \tilde{y}_j} + \frac{\tilde{y}_j^T \tilde{y}_j}{s_j^T \tilde{y}_j} \cdot \frac{s_j^T s_j}{s_j^T \tilde{y}_j} \\ &\quad \cdot \|\hat{H}_t^{(i)}\| + \frac{s_j^T s_j}{s_j^T \tilde{y}_j}. \end{aligned} \quad (38)$$

Recall from the proof of Lemma 2 that  $\frac{\tilde{y}_j^T \tilde{y}_j}{s_j^T \tilde{y}_j} \leq Q + 5\rho$ . Moreover, according to Lemma 1, we have  $\frac{s_j^T s_j}{s_j^T \tilde{y}_j} \leq$

$\frac{s_j^T s_j}{0.2s_j^T (\hat{B}_{j+1}^{(0)} + \delta I) s_j} = \frac{1}{0.2(\tau_{j+1} + \delta)}$  and hence

$$\frac{\|s_j\| \cdot \|\tilde{y}_j\|}{s_j^T \tilde{y}_j} = \left( \frac{s_j^T s_j}{s_j^T \tilde{y}_j} \cdot \frac{\tilde{y}_j^T \tilde{y}_j}{s_j^T \tilde{y}_j} \right)^{1/2} \leq \sqrt{\frac{Q + 5\rho}{0.2(\tau_{j+1} + \delta)}}. \quad (39)$$

Substituting the above results into (38) and noting the fact  $\tau_{j+1} \geq \beta$ , (38) can be simplified to

$$\|\hat{H}_t^{(i+1)}\| \leq w^2 \|\hat{H}_t^{(i)}\| + \frac{1}{0.2(\beta + \delta)}. \quad (40)$$

By induction with  $\hat{H}_t^{(0)} \preceq \beta^{-1}I$ , we obtain the desired result.

Based on the above uniformly upper bound and lower bound on the resultant L-BFGS matrix, we now derive the convergence result of our proposed algorithm. Moreover, the following assumption is required.

*Assumption 2.* For any iteration, the variance of the gradient conditioned on current iterate is bounded above:

$$\mathbb{E}(\|\nabla F(x_k, \xi_k) - \nabla f(x_k)\|^2 | x_k) \leq \sigma^2. \quad (41)$$

Moreover, the norm square of the gradient is expected to be bounded above by a positive constant  $D$  [3], [10], [15]:

$$\mathbb{E}(\|\nabla F(x_k, \xi_k)\|^2 | x_k) \leq D. \quad (42)$$

With Assumption 2, we introduce the following lemma:

*Lemma 4* [3], [8]. Suppose Assumption 2 holds, and the sequence  $\{x_k\}$  for  $k = 1, \dots$ , is generated with the initial value  $x_0$  and using a specific constant batch size  $m_k = m$ . Then there exists a positive constant  $M_f$  such that  $\mathbb{E}[f(x_k)] \leq M_f$ . Moreover, the sequence almost surely converges to a stationary point, i.e.,  $\lim_{k \rightarrow \infty} \|\nabla f(x_k)\| = 0$ , with probability 1.

We are now ready to proceed to show the convergence of our proposed algorithm under the given assumptions, which is summarized in the following theorem. Without loss of generality, the interval length is assumed to be unity.

*Theorem 1.* Suppose the iterations of the Sd-REG-LBFGS algorithm satisfies Assumption 2, and the sequence  $\{x_k\}$  for  $k = 1, \dots, N-1$  is generated with initial value  $x_0$ . Given the constant batch size  $m_k = m$  and in particular the following step size:

$$\eta_k = \frac{\eta_0 Q_U^{-1}}{k^v + (L_f/2)\eta_0 Q_L^{-2}}, \quad (43)$$

with  $0.5 < v < 1$ , the following inequality holds:

$$\begin{aligned} \frac{1}{N} \sum_{k=0}^{N-1} \mathbb{E}(\|\nabla f(x_k)\|^2) &\leq \frac{[(N-1)^v + (L_f/2)\eta_0 Q_L^{-2}]^2}{\eta_0 Q_U^{-2} (N-1)^v N} \\ &\quad \cdot (M_f - f^l) + \frac{L_f Q_L^{-2} \sigma^2 \eta_0 [(N-1)^{1-v} - 1]}{2m(1-v)N}, \end{aligned} \quad (44)$$

where  $f^l := \min\{f(x_0), \dots, f(x_{N-1})\}$  and  $N$  is the iteration number. Furthermore, given a constant  $0 < \epsilon < 1$ , the iteration number  $N$  needed to ensure  $\frac{1}{N} \sum_{k=0}^{N-1} \mathbb{E}(\|\nabla f(x_k)\|^2) \leq \epsilon$  is at most  $O(\epsilon^{-\frac{1}{1-v}})$ .

*Proof.* Recall that the gradient of  $f(\cdot)$  is Lipschitz continuous with constant  $L_f$ , therefore, using second-order Taylor expansion at iteration  $k$  leads to:

$$\begin{aligned} f(x_{k+1}) &\leq f(x_k) + \nabla f(x_k)^T (x_{k+1} - x_k) + \frac{L_f}{2} \|x_{k+1} - x_k\|^2 \\ &= f(x_k) + \nabla f(x_k)^T (-\eta_k \hat{B}_k^{-1} \bar{g}_k) + \frac{L_f}{2} \eta_k^2 \|\hat{B}_k^{-1} \bar{g}_k\|^2 \\ &\leq f(x_k) - \eta_k \nabla f(x_k)^T \hat{B}_k^{-1} \bar{g}_k + \frac{L_f}{2} \eta_k^2 \|\hat{B}_k^{-1}\|^2 \cdot \|\bar{g}_k\|^2, \end{aligned} \quad (45)$$

where for notational convenience, we denote  $\bar{g}_k = \bar{g}_k(x_k, \xi_k)$ . From Lemma 2 and Lemma 3, we have  $Q_U^{-1}I \preceq \hat{B}_k^{-1} \preceq Q_L^{-1}I$ . Substituting it into (45) results in:

$$f(x_{k+1}) \leq f(x_k) - \eta_k Q_U^{-1} \nabla f(x_k)^T \bar{g}_k + \frac{L_f}{2} \eta_k^2 Q_L^{-2} \|\bar{g}_k\|^2. \quad (46)$$

To evaluate the expectation of (46), we shall first take the expectation conditioned on  $x_k$  on both sides and then the expectation with respect to  $x_k$ . We shall make use of the fact that  $\mathbb{E}_B[\mathbb{E}_A(A|B)] = \mathbb{E}(A)$  for random variables  $A$  and  $B$ . Consequently, with Assumption 2, we get:

$$\begin{aligned} \mathbb{E}[f(x_{k+1})] &\leq \mathbb{E}[f(x_k)] - \eta_k Q_U^{-1} \mathbb{E}[\nabla f(x_k)^T \mathbb{E}(\bar{g}_k | x_k)] \\ &\quad + \frac{L_f}{2} \eta_k^2 Q_L^{-2} \mathbb{E}[\mathbb{E}(\|\bar{g}_k\|^2 | x_k)]. \end{aligned} \quad (47)$$

Furthermore, we have:

$$\mathbb{E}(\|\bar{g}_k - \nabla f(x_k)\|^2 | x_k) = \mathbb{E}(\|\bar{g}_k\|^2 | x_k) - \|\nabla f(x_k)\|^2, \quad (48)$$

and it further yields  $\mathbb{E}(\|\bar{g}_k\|^2 | x_k) = \sigma^2/m + \|\nabla f(x_k)\|^2$ . Substituting the result into (47), we have:

$$\begin{aligned} \mathbb{E}[f(x_{k+1})] &\leq \mathbb{E}[f(x_k)] - \left( \eta_k Q_U^{-1} - \frac{L_f}{2} \eta_k^2 Q_L^{-2} \right) \\ &\quad \cdot \mathbb{E}(\|\nabla f(x_k)\|^2) + \frac{L_f \eta_k^2 Q_L^{-2} \sigma^2}{2m}. \end{aligned} \quad (49)$$

By summing (49) for  $k = 0, \dots, N-1$ , the following result is obtained:

$$\begin{aligned} \sum_{k=0}^{N-1} \mathbb{E}(\|\nabla f(x_k)\|^2) &\leq \sum_{k=0}^{N-1} \frac{\mathbb{E}[f(x_k)] - \mathbb{E}[f(x_{k+1})]}{\eta_k Q_U^{-1} - (L_f/2) \eta_k^2 Q_L^{-2}} \\ &\quad + \sum_{k=0}^{N-1} \frac{L_f \eta_k Q_L^{-2} \sigma^2}{2m [Q_U^{-1} - (L_f/2) \eta_k Q_L^{-2}]}. \end{aligned} \quad (50)$$

Furthermore, from (43), we have  $\frac{\eta_k}{Q_U^{-1} - (L_f/2) \eta_k Q_L^{-2}} = \eta_0 k^{-v}$ . Substituting it into (50), we obtain the simplified inequality:

$$\begin{aligned} \sum_{k=0}^{N-1} \mathbb{E}(\|\nabla f(x_k)\|^2) &\leq \sum_{k=0}^{N-1} \frac{\eta_0 k^{-v}}{\eta_k^2} (\mathbb{E}[f(x_k)] \\ &\quad - \mathbb{E}[f(x_{k+1})]) + \frac{L_f Q_L^{-2} \sigma^2 \eta_0}{2m} \sum_{k=0}^{N-1} k^{-v}. \end{aligned} \quad (51)$$

By utilizing the result in Lemma 3 that  $\mathbb{E}[f(x_k)] \leq M_f$ , we

further have:

$$\begin{aligned} \sum_{k=0}^{N-1} \mathbb{E}(\|\nabla f(x_k)\|^2) &\leq \sum_{k=1}^{N-1} \left( \frac{\eta_0 k^{-v}}{\eta_k^2} - \frac{\eta_0 (k-1)^{-v}}{\eta_{k-1}^2} \right) \mathbb{E}[f(x_k)] \\ &\quad - \frac{\eta_0 (N-1)^{-v}}{\eta_{N-1}^2} \mathbb{E}[f(x_N)] + \frac{L_f Q_L^{-2} \sigma^2 \eta_0}{2m} \sum_{k=0}^{N-1} k^{-v} \\ &\leq M_f \sum_{k=1}^{N-1} \left( \frac{\eta_0 k^{-v}}{\eta_k^2} - \frac{\eta_0 (k-1)^{-v}}{\eta_{k-1}^2} \right) \\ &\quad - \frac{\eta_0 (N-1)^{-v}}{\eta_{N-1}^2} f^l + \frac{L_f Q_L^{-2} \sigma^2 \eta_0}{2m} \sum_{k=0}^{N-1} k^{-v} \\ &= \frac{\eta_0 (M_f - f^l) (N-1)^{-v}}{\eta_{N-1}^2} + \frac{L_f Q_L^{-2} \sigma^2 \eta_0}{2m} \sum_{k=0}^{N-1} k^{-v} \\ &= \frac{[(N-1)^v + (L_f/2) \eta_0 Q_L^{-2}]^2 (M_f - f^l)}{\eta_0 Q_U^{-2} (N-1)^v} \\ &\quad + \frac{L_f Q_L^{-2} \sigma^2 \eta_0}{2m} \sum_{k=0}^{N-1} k^{-v}. \end{aligned} \quad (52)$$

By applying following inequality:

$$k^{-v} \leq \frac{k^{1-v} - (k-1)^{1-v}}{1-v}, \text{ for } k \geq 1, \quad (53)$$

to (52), we obtain the desired result in (44). For a given constant  $\epsilon$  satisfying  $0 < \epsilon < 1$ , the iteration number needed to guarantee  $\frac{1}{N} \sum_{k=0}^{N-1} \mathbb{E}(\|\nabla f(x_k)\|^2) < \epsilon$  satisfies:

$$\begin{aligned} &\frac{[(N-1)^v + (L_f/2) \eta_0 Q_L^{-2}]^2 (M_f - f^l)}{\eta_0 Q_U^{-2} (N-1)^v N} + \\ &\frac{L_f Q_L^{-2} \sigma^2 \eta_0 [(N-1)^{1-v} - 1]}{2m(1-v)N} < \epsilon. \end{aligned} \quad (54)$$

Therefore, for  $0.5 < v < 1$ , the iteration number is at most  $O(\epsilon^{-\frac{1}{1-v}})$  to reach  $\frac{1}{N} \sum_{k=0}^{N-1} \mathbb{E}(\|\nabla f(x_k)\|^2) < \epsilon$ .

#### IV. EMPIRICAL STUDY

We have studied the theoretical properties and the convergence of the proposed quasi-Newton method in the previous section. In this section, we will apply the proposed method to solve several optimization problems in machine learning. Specifically, two machine learning problems will be studied, namely logistic regression and Bayesian logistic regression for binary classification. To carry out the optimization, the gradient required by the Sd-REG-LBFGS method is obtained analytically. In a general fashion, we mainly focus on nonconjugate exponential models under stochastic regime, in which Bayesian logistic regression is a particular example.

##### A. Logistic Regression

We first consider the logistic regression problem. The objective function is given as follows [17]:

$$f(\theta) = -\frac{1}{N} \sum_{n=1}^N z_n \log \sigma(\theta^T x_n) + (1 - z_n) \log \sigma(-\theta^T x_n), \quad (55)$$

where  $\sigma(\cdot)$  is the sigmoid function given by  $\sigma(x) = 1/(1 + \exp(-x))$ ,  $x_n$  is the feature vector and  $z_n$  is its label.

### B. Sd-REG-LBFGS for VBI

Variational Bayesian inference (VBI) is an efficient method for approximating the a posteriori probability distributions for making inference. The main ingredient is to convert inference problems into optimization problems with the KL-divergence as the objective function. Another popular scheme for making inference is Markov chain Monte Carlo (MCMC) sampling method. It can be easily parallelized for multiple processors to reduce the computational cost for high dimension problems. In this section, we illustrate the application of the proposed Sd-REG-LBFGS to the delta VBI scheme for nonconjugate models proposed in [18]. The resultant algorithm is denoted by SDVBI. In addition, interested readers can refer to [23], [29], [35] for applications of optimization methods in VBI.

Suppose  $x_{1:N}$  are observations,  $z_{1:N}$  are local hidden variables and  $\theta$  is global hidden variable. Furthermore,  $\theta$  is the nonconjugate variable and  $z_{1:N}$  are conjugate variables. Consider the nonconjugate model in [18] as follows:

$$p(x, z, \theta) = p(\theta) \cdot \prod_{n=1}^N p(x_n|z_n)p(z_n|\theta), \quad (56)$$

where  $p(z_n|\theta) = h(z_n)\exp\{\eta_{g_n}(\theta)^T t(z_n) - a(\eta_{g_n}(\theta))\}$  and  $p(x_n|z_n) = h(x_n)\exp\{t(z_n)^T [t(x_n)^T, 1]^T\}$ . The goal of variational inference is to approximate the posterior distribution by finding a member of a specific family  $\mathcal{Q}$  to minimize its KL-divergence to the true a posteriori distribution:

$$q^*(z, \theta) = \operatorname{argmin}_{q \in \mathcal{Q}} \operatorname{KL}(q(z, \theta) \| p(z, \theta|x)). \quad (57)$$

For the MFVI framework, the statistical independence between hidden variables with a fully factorized variational distribution family are assumed, i.e.,

$$q(z, \theta) = q(\theta|\lambda) \cdot \prod_{n=1}^N q(z_n|\varphi_n), \quad (58)$$

where  $q(z_n|\varphi_n) = h(z_n)\exp\{\eta_l(\varphi_n)^T t(z_n) - a(\eta_l(\varphi_n))\}$ , and Gaussian distribution has been adopted to approximate its variational distribution, i.e.,  $q(\theta|\lambda) = \mathcal{N}(\mu, S)$ , with  $\lambda$  being the parameter pair  $(\mu, S)$ . The following can be obtained by substituting the results into (57):

$$\begin{aligned} \operatorname{KL}(q \| p) &= \mathbb{E}_q[\log q(z_{1:N}, \theta)] - \mathbb{E}_q[\log p(z_{1:N}, \theta|x_{1:N})] \\ &= \mathbb{E}_q[\log q(z_{1:N}, \theta)] - \mathbb{E}_q[\log p(x_{1:N}, z_{1:N}, \theta)] + \operatorname{const}. \\ &:= \mathcal{L}(q). \end{aligned} \quad (59)$$

First, for nonconjugate variable  $\theta$ , the objective function of delta VBI has been derived based on second-order Taylor approximation of the variational objective function in [18] as follows:

$$\begin{aligned} \mathcal{L}(\lambda) &= \mathbb{E}_{q(\theta|\lambda)}[\log q(\theta|\lambda)] - \sum_{n=1}^N \mathbb{E}_{q(\theta, z_n)}[\log p(z_n|\theta)] \\ &\approx d(\mu) + \frac{1}{2}(\operatorname{Tr}\{\nabla^2 d(\mu)S\} - \log \det S) + \operatorname{const}. \end{aligned} \quad (60)$$

where  $d(\theta) := -\eta_g(\theta)^T \cdot \sum_{n=1}^N \nabla_{\eta_l} a(\eta_l(\varphi_n)) + Na(\eta_g(\theta)) - \log p(\theta)$ , the optimization problem becomes:

$$\lambda^* = \operatorname{argmin}_{\lambda \in \mathbb{R}^d} \{\mathcal{L}(\lambda) = -\frac{1}{2} \log \det S + \mathbb{E}_{q(\theta|\lambda)} d(\theta)\}. \quad (61)$$

We notice that (60) contains large summation term, which makes the gradient evaluation computationally rather expensive. Next, we randomly sample a subset  $\mathcal{S}$  from  $\{1, \dots, N\}$  to form an unbiased stochastic gradient, which is denoted as  $\nabla_{\lambda} \mathcal{L}(\lambda; \mathcal{S})$ . We omit the reduplicative and tedious derivation, as the full gradient can be found in Appendix C of [18].

For the conjugate variable  $z_n$  updating, the variational objective function  $\mathcal{L}(\varphi_n)$  from the KL-divergence in (59) in [18] is as follows:

$$\begin{aligned} \mathcal{L}(\varphi_n) &= \mathbb{E}_{q(z_n)}[\log q(z_n)] - \mathbb{E}_{q(z_n)}[\log p(x_n|z_n)] \\ &\quad - \mathbb{E}_{q(z_n, \theta)}[\log p(z_n|\theta)] + \operatorname{const} \\ &= \{\eta_l(\varphi_n)^T - [t(x_n)^T, 1] - \mathbb{E}_{q(\theta)}[\eta_g(\theta)^T]\} \\ &\quad \cdot \nabla_{\eta_l} a(\eta_l(\varphi_n)) - a(\eta_l(\varphi_n)) + \operatorname{const}, \end{aligned} \quad (62)$$

where the last equality in (62) follows from the basic property of the exponential family. To derive the update for  $\varphi_n$ , we take the gradient of  $\mathcal{L}(\varphi_n)$ :

$$\begin{aligned} \nabla_{\varphi_n} \mathcal{L}(\varphi_n) &= \mathcal{D}_{\varphi_n} \eta_l(\varphi_n)^T \cdot \nabla_{\eta_l}^2 a(\eta_l(\varphi_n)) \{\eta_l(\varphi_n) \\ &\quad - [t(x_n)^T, 1]^T - \mathbb{E}_{q(\theta)}[\eta_g(\theta)]\}, \end{aligned} \quad (63)$$

where  $\mathcal{D}_{\varphi_n} \eta_l(\varphi_n)$  is the Jacobian matrix of  $\eta_l(\cdot)$  with respect to  $\varphi_n$ . Therefore, by using the gradient for optimization or by simply setting the gradient to zero, i.e.,  $\nabla_{\varphi_n} \mathcal{L}(\varphi_n) = 0$ , we obtain the conjugate variable update. With the above stochastic gradients derived, we have shown the application of the proposed method.

In particular, with the following settings [18]:

$$\begin{aligned} h(z_n) &= 1, \quad t(z_n) = [z_n, 1 - z_n]^T, \quad a(\eta_g(\theta)) = 0, \\ \eta_{g_n}(\theta) &= [\log \sigma(\theta^T x_n), \log \sigma(-\theta^T x_n)]^T, \quad n = 1, \dots, N, \end{aligned} \quad (64)$$

one recovers Bayesian logistic regression. Here, it should be noted that VBI is only considered for the nonconjugate variable  $\theta$ . However, for the settings of correlated topic model, VBI is considered for both  $\theta$  and  $z_n$ . As the applications of Sd-REG-LBFGS are similar, we shall consider Bayesian logistic regression for numerical experiments for simplicity.

## V. NUMERICAL RESULTS

In this section, the numerical experiments are performed on our proposed Sd-REG-LBFGS algorithm. Two applications are considered in machine learning, which are logistic regression and SDVBI for Bayesian logistic regression. Moreover, in this paper, we only consider binary classification problems. We also employed a synthetic dataset and several real datasets [38], [41]–[45] for the performance evaluation. In particular for the parameter studies, we use a synthetic dataset and a real scene dataset [38] (available at <http://mulan.sourceforge.net/datasets-mlc.html>), which can be categorized as the following 4 scenarios:

S1. Solving logistic regression (LR) using synthetic dataset, which is presented in Section V-A;

S2. Solving Bayesian logistic regression (BLR) using synthetic dataset, which is presented in Section V-A;

S3. Solving LR using *scene* dataset in [38]. Due to page limitation, the results are presented in Section II of the supplementary material;

S4. Solving BLR using *scene* dataset in [38]. Due to page limitation, the results are presented in Section II of the supplementary material.

The following algorithms are considered for evaluation:

(A) Proposed Sd-REG-LBFGS: The proposed stochastic damped regularized L-BFGS as described in Algorithm 1;

(B) SdLBFGS: Stochastic damped regularized L-BFGS without regularization in [8];

(C) SGD: Stochastic gradient descent is adopted;

(D) SAA: Stochastic approximation averaging in [39] is applied;

(E) RSA: Robust stochastic approximation in [20] is employed.

(F) Adam: Adam [40] is employed.

Here, we summarize again some key parameters and variables that are involved in the numerical experiments.

1.  $d$ : the dimension of the optimization variable, e.g.,  $\theta \in \mathbb{R}^d$ .

2.  $N$ : the number of training points in the dataset.

3.  $m$ : the batch size used for stochastic approximation of the gradient. In the numerical experiment, we use constant batch size at each iteration, e.g.,  $m = |\mathcal{S}|$  for  $\nabla_{\theta}\mathcal{L}(\theta; \mathcal{S})$ .

4.  $M$ : the memory size used for the Sd-REG-LBFGS algorithm to store the correction pairs  $(s_t, y_t)$  calculated by (20) and (21).

5.  $L$ : the interval length. Every  $L$  iterations, we perform averaging on the iterate points, which is used to calculate the correction pairs by (20) and (21).

6.  $\gamma$ : the regularized parameter for BFGS update given in (24), which prevent the L-BFGS matrix from being close to singularity.

7.  $\eta_k$ : the step size for SGD, SdLBFGS and Sd-REG-LBFGS optimization schemes. In the numerical experiment, we adopt the diminishing step size  $\eta_k = r/k$  with a positive constant  $r$  at each iteration.

In general, for the regression problems, one needs to include a constant bias term. This can be implemented by concatenating a unity element at the beginning or the end of each input vector, i.e., if the unity is put at the beginning,  $\theta_0 + \theta^T x_n = [\theta_0, \theta^T][1, x_n^T]$ . For notational convenience, we omit the bias term here. The performance of various algorithms will be evaluated in terms of the norm of the gradient (NOG) and the classification accuracy (ACC). The NOG for LR is defined as follows:

$$\text{NOG} = \left\| \frac{1}{N} \sum_{n=1}^N [z_n - \sigma(\theta^T x_n)] x_n \right\|. \quad (65)$$

Moreover, the exact gradient of the objective function in BLR

can be calculated as follows [18]:

$$\begin{aligned} \nabla_{\theta}\mathcal{L}(\theta) = & \frac{1}{N} \sum_{n=1}^N \{ [z_n - \sigma(\theta^T x_n)] x_n + \frac{1}{2} \sigma(\theta^T x_n) \\ & \cdot \sigma(-\theta^T x_n) [1 - 2\sigma(\theta^T x_n)] x_n x_n^T S x_n \} + S_0^{-1}. \end{aligned} \quad (66)$$

Hence, the NOG for BLR is defined as  $\text{NOG} = \|\nabla_{\theta}\mathcal{L}\|$ .

Lower NOG indicates the better convergence of an algorithm to a stationary point. The classification accuracy is given as

$$\text{ACC} = \frac{TP + TN}{TP + FN + FP + FN} \quad (67)$$

where TP, TN, FP, and FN denote true positives, true negatives, false positives and false negatives, respectively. The decision rules for class prediction are given as

$$\text{if } \sigma(\hat{\theta}^T x_n) \geq 0.5, \text{ then } z_n = 1, \text{ else } z_n = 0, \quad (68)$$

for logistic regression and Bayesian logistic regression respectively, where  $\hat{\theta}$  is the estimated value of  $\theta$ . We have also conducted a sensitivity analysis to study the effects of different batch sizes, memory sizes and regularization parameters with our proposed method. Due to page limitation, the details are omitted here. Interested readers are referred to Section II of the supplementary material for details.

#### A. Performance comparisons with different real datasets

In this subsection, we first study the effectiveness of our proposed method using the same settings of each algorithms for their common parameters with different real datasets. Specifically, for all schemes, batch size is set to a relatively small value to show that our proposed method is particularly effective. Moreover, the real datasets are described as follows:

1. *Banknote Authentication Dataset* (BNA) [45] (available at UCI Machine Learning Repository): we use 1,370 samples, which has 4 variables. Considering 5-fold cross validation, there are 1,096 data points for training and 274 samples for testing.

2. *Wireless Indoor Localization Dataset* (WINL) [42], [43] (available at UCI Machine Learning Repository): 2,000 samples with 7 variables are used for the performance evaluation. For 5-fold cross-validation, there are 1,600 data points for training and 400 samples for testing.

3. *Ionosphere Dataset* (IONO) [44] (available at UCI Machine Learning Repository): we use 350 samples with 33 variables for performance evaluation. There are 280 data points for training and 70 samples for testing according to 5-fold cross validation.

4. *Electrical Grid Stability Simulated Dataset* (ELEG) [41] (available at UCI Machine Learning Repository): 10,000 samples of the dataset with 14 variables is used for performance evaluation, of which 8,000 are for training and 2,000 are for testing according to 5-fold cross validation.

Moreover, the batch size and step size for each algorithm are set to  $m = 20$  and  $\eta_k = 7/k$ , respectively. For our proposed method and SdLBFGS, the memory is set to the same value  $M = 10$ . We set the regularization parameters for our proposed method to  $\gamma = 10^{-4}$  and  $\delta = 1.25 + 0.01$ ,

Table I: The NOG and ACC performances of various algorithms averaged over 50 Monte Carlo simulations and 5-fold cross validation with different datasets for logistic regression and Bayesian logistic regression.

Dataset	Algorithms	NOG (LR)	ACC (LR)	NOG (BLR)	ACC (BLR)
BNA	Sd-REG-LBFGS	<b>0.0288</b>	<b>95.27%</b>	<b>0.0294</b>	<b>95.47%</b>
	SdLBFGS	0.0313	95.17%	0.3351	91.67%
	RSA	0.0317	95.11%	0.3306	90.36%
	SAA	1.7341	95.11%	1.6801	90.35%
	SGD	0.0318	95.10%	0.3371	90.35%
	Adam	0.2592	92.23%	0.1349	94.52%
WINL	Sd-REG-LBFGS	<b>0.010</b>	97.11%	<b>0.0073</b>	91.42%
	SdLBFGS	0.012	97.31%	0.0077	91.43%
	RSA	0.0654	95.89%	0.023	91.12%
	SAA	1.36	95.89%	0.595	91.12%
	SGD	0.0653	95.90%	0.023	91.12%
	Adam	0.54	80.74%	0.10	86.46%
IONO	Sd-REG-LBFGS	<b>0.013</b>	<b>87.33%</b>	<b>0.0188</b>	<b>88.21%</b>
	SdLBFGS	0.017	86.98%	0.0741	86.82%
	RSA	0.087	85.70%	0.096	84.59%
	SAA	0.795	85.70%	0.795	84.87%
	SGD	0.087	85.70%	0.099	84.87%
	Adam	0.19	79.13%	0.207	80.80%
ELEG	Sd-REG-LBFGS	<b>0.017</b>	87.55%	<b>0.016</b>	87.56%
	SdLBFGS	0.020	87.68%	0.02	87.67%
	RSA	0.043	86.72%	0.024	87.27%
	SAA	0.502	86.72%	0.502	87.27%
	SGD	0.042	86.72%	0.0241	87.27%
	Adam	0.489	52.69%	0.493	64.53%

respectively. For each NOG and ACC value, it is computed via the average of 5-fold cross validation and 50 Monte Carlo runs. The results are shown in Table I. It can be seen that our proposed method performs the best obviously in terms of NOG performance. For ACC performance evaluation, our proposed method is generally better than other methods, except that the proposed method is slightly worse than SdLBFGS for WINL and ELEG. This is due to the bias that our method has introduced. However, Sd-REG-LBFGS is more robust as SdLBFGS has resulted in ill-conditioning problems during the experiments.

Next, we will consider the synthetic dataset and the real dataset *scene* [38] to extensively study the effects of different parameter settings.

### B. Numerical results using synthetic dataset

In this subsection, we conduct the numerical experiments using synthetic dataset. For the binary classification schemes, we initialize the parameter to be optimized as  $\theta_0$ , which is

generated from a Gaussian distribution  $\mathcal{N}(0, I)$ . For SDVBI, the initial values of mean  $\mu$  and the covariance matrix  $S$  are set to  $\theta_0$  and identity matrix  $S_0 = I$ , respectively. We generate 5000 synthetic data points for 5-fold cross validation and 50 Monte Carlo runs in the following manner. Each of the sample  $x_n$  is of dimension  $d = 50$  and is randomly drawn from a uniform distribution  $[0, 1]^d$ . The desired parameter  $\bar{\theta}$  generated from the uniform distribution  $[-1, 1]^d$  is used to generate the true class labels  $z_n = \mathbb{I}(\bar{\theta}^T x_n > 0)$  for each of the sample  $x_n$ . Using the synthetic dataset, we first consider the logistic regression problem and the objective function given in (55).

1) *Logistic Regression*: In Figs. 1(a) and 1(b), we illustrate the effect of batch size on the Sd-REG-LBFGS algorithm in terms of NOG and ACC, respectively. The regularized parameter  $\gamma$  is set to  $\gamma = 10^{-4}$  and  $\delta$  to  $\delta = 1.25\gamma + 0.01$  correspondingly. Fig. 1(a) shows that the proposed approach consistently performs better than the SdLBFGS, SGD, RSA, SAA and Adam algorithms in NOG. Larger batch size generally leads to better performance for all algorithms. This is due to less variance of the stochastic gradient with larger batch size.

Figs. 1(a) and (b) show that the proposed approach and the SdLBFGS consistently performs better than SGD, RSA, SAA and Adam algorithms in terms of NOG and ACC, respectively. Moreover, the proposed algorithm performs consistently well for different batch sizes, which suggests that the incorporation of regularization helps to reduce estimation variance and hence it is more robust to the variations of batch sizes.

In Figs. 2(a) and 2(b), we report the effect of various memory sizes on the performance of the proposed Sd-REG-LBFGS. We set the step size constant to  $r = 7$  and the batch size  $m = 100$  for the proposed approach and SdLBFGS. The regularized parameters of the proposed approach are set to  $\gamma = 10^{-4}$  and  $\delta$  to  $\delta = 1.25\gamma + 0.01$ , respectively, which satisfies the condition  $0.8\delta > \gamma$ . Furthermore, the iteration interval length is set to  $L = 10$ . From the figures, we can see that the proposed approach and the SdLBFGS give better NOG and ACC performance than the SGD, RSA, SAA and Adam. Moreover, a larger memory size generally lead to more accurate approximation of the Hessian matrix and hence a better performance.

In Figs. 3(a) and 3(b), we study the effect of the regularization parameter on Sd-REG-LBFGS in terms of NOG and ACC, respectively. The following values of  $\gamma = 10^{-2}, 10^{-3}, 10^{-4}$  are employed. We can see that the proposed approach performs better in terms of NOG and ACC. We notice the small amount of regularization imposed in the proposed Sd-REG-LBFGS method generally lead to better NOG than the SdLBFGS while its ACC is similar to SdLBFGS.

Overall, we find that the proposed approach performs better than other conventional algorithms in terms of NOG and ACC. This may be attributed by the small amount of regularization applied to the proposed approach, which improves the numerical stability and hence it converges closer to the stationary point (lower NOG). Meanwhile, we notice that the ACC of the proposed Sd-REG-LBFGS and the SdLBFGS algorithms are quite similar under this setting. We shall compare these algorithms more formally using a statistical test on their

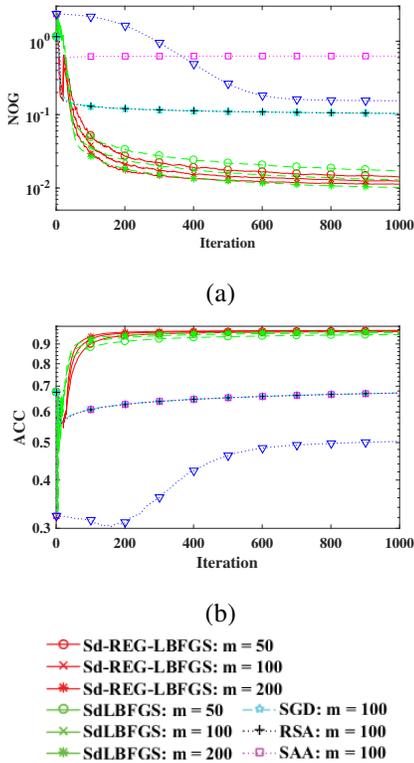


Figure 1: The (a) Norm of Gradient (NOG) and (b) Classification Accuracy (ACC) of logistic regression solved using various algorithms with different batch sizes averaged over 50 Monte Carlo simulations. The synthetic dataset is used.

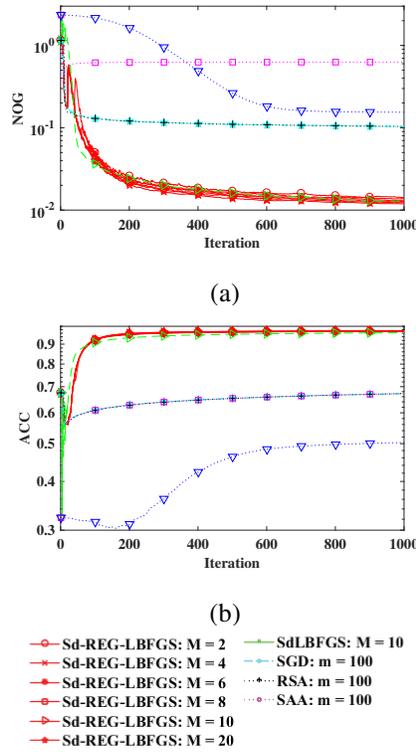


Figure 2: The (a) NOG and (b) ACC of logistic regression solved using various algorithms with different memory sizes averaged over 50 Monte Carlo simulations. For comparison, SdLBFGS, SGD, RSA, SAA and Adam are implemented. The synthetic dataset is used.

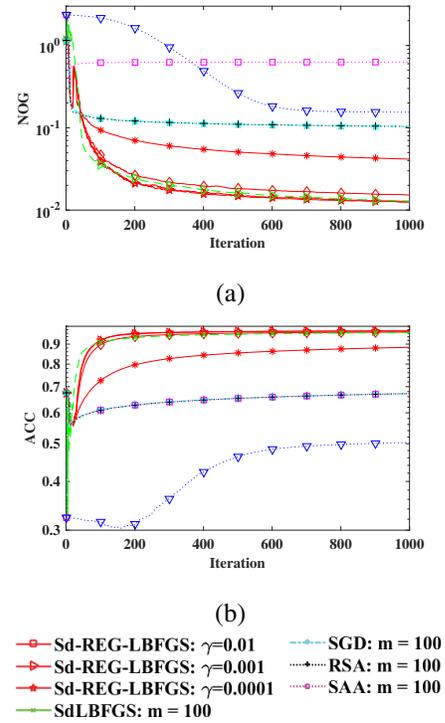


Figure 3: The effect of regularized parameter  $\gamma$  on the (a) NOG and (b) ACC of logistic regression solved using the proposed Sd-REG-LBFGS. For comparison, SdLBFGS, SGD, RSA, SAA and Adam are implemented. The synthetic dataset is used.

average classification accuracies at different settings in Section V-B.

2) *SDVBI for Bayesian Logistic Regression*: In this subsection, numerical experiments are performed on SDVBI for Bayesian logistic regression using synthetic dataset. We will study various values of the batch size  $m$ , the memory size  $M$  and the regularization parameter  $\gamma$ , under which Sd-REG-LBFGS is performed for the optimization.

Figs 4(a) and 4(b) show the NOG and ACC of SDVBI, respectively, solved using different algorithms with various batch sizes. In this experiment, we fix the regularization parameters to  $\gamma = 10^{-4}$  and  $\delta = 1.25\gamma + 0.01$  respectively. Moreover, we set the memory size to  $M = 10$  and the interval length to  $L = 10$ .

From the figures, it can be seen that the proposed method generally outperforms the Sd-LBFGS, SGD, RSA, SAA and Adam algorithms with all batch sizes studied. Moreover, the proposed algorithm performs consistently well for different batch size, which suggests the incorporation of regularization helps to reduce estimation variance and hence it is more robust to the variations of batch sizes. This enables us to choose a smaller batch size so that it could reduce computational cost without sacrificing much classification performance of the SDVBI in Bayesian logistic regression.

Figs. 5(a) and 5(b) show the effect of memory size on the NOG and classification performances of SDVBI in Bayesian

logistic regression using the proposed Sd-REG-LBFGS. The Sd-LBFGS, SGD, RSA, SAA and Adam are also included as benchmarks. Similar to previous sub-sections, we fix the regularized parameter to  $\gamma = 10^{-4}$  and  $\delta = 1.25\gamma + 0.01$  respectively. Moreover, the batch size is set to  $m = 100$  and the interval length for Sd-REG-LBFGS is set to  $L = 10$ .

From the figures, we find that the NOG and ACC performance of the proposed approach is generally better than other approaches. Thus, we can choose a relatively small memory size to reduce the computational cost without sacrificing performance.

In Figs 6(a) and 6(b), we report the effect of regularization parameter  $\gamma$  on Sd-REG-LBFGS for SDVBI. In general, smaller  $\gamma$  value yields better performance in terms of NOG. For the Sd-REG-LBFGS with  $\gamma = 10^{-2}, 10^{-3}, 10^{-4}$ , the improvement in classification performance decreases when decreases. We notice the small amount of regularization imposed in the proposed Sd-REG-LBFGS method generally lead to better NOG than the SdLBFGS while its ACC is similar to SdLBFGS.

Overall, the proposed Sd-REG-LBFGS performs better than the SdLBFGS, SGD, RSA, SAA and Adam algorithms in terms of NOG and classification accuracy. Moreover, the classification performance of the proposed Sd-REG-LBFGS algorithm is less vulnerable to insufficient samples caused by small batch size as regularization is imposed to avoid

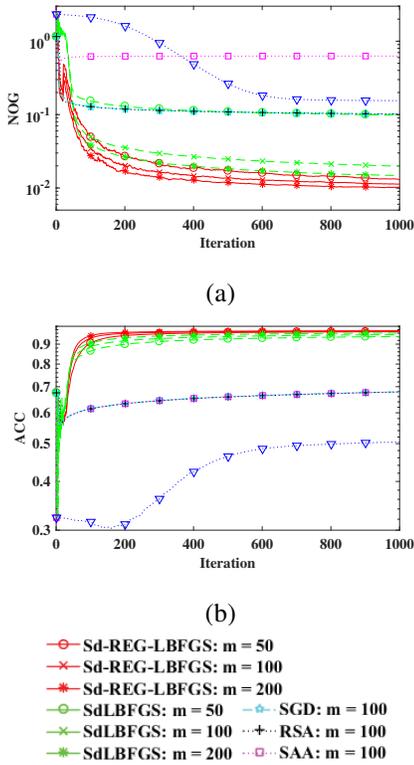


Figure 4: The (a) NOG and (b) ACC of different algorithms with various batch sizes in solving SDVBI in Bayesian logistic regression. For comparison, SdLBFGS, SGD, RSA, SAA and Adam are included. The synthetic dataset is used.

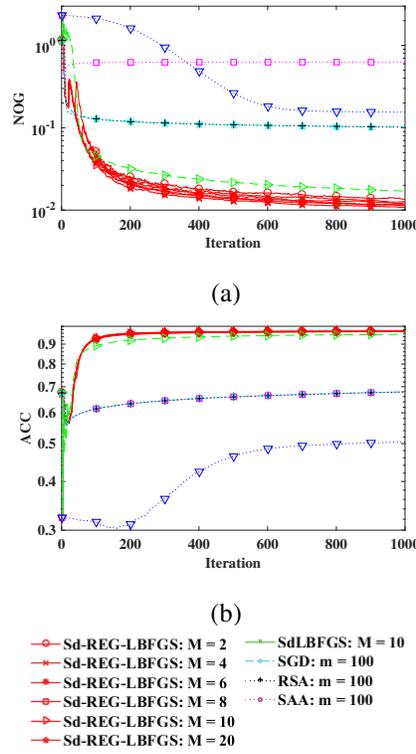


Figure 5: The (a) NOG and (b) ACC of SDVBI in Bayesian logistic regression solved using various algorithms with different memory sizes averaged over 50 Monte Carlo simulations. The synthetic dataset is used.

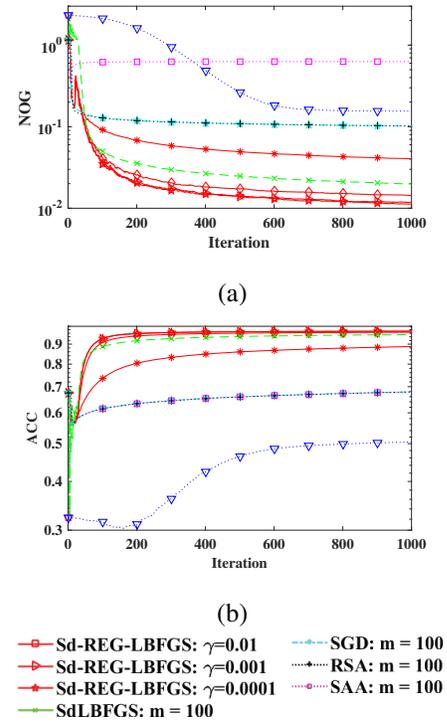


Figure 6: The effect of regularized parameter  $\gamma$  on the (a) NOG and (b) ACC of SDVBI in Bayesian logistic regression solved using the proposed Sd-REG-LBFGS. For comparison, SdLBFGS, SGD, RSA, SAA and Adam are implemented. The synthetic dataset is used.

ill-conditioning of the Hessian update. On the other hand, the proposed Sd-REG-LBFGS and SdLBFGS gives similar performance when the number of samples is large.

Regarding to the choice of the algorithmic parameters including the batch size  $m$ , memory size  $M$  and the regularization parameter  $\gamma$ , we observe a choice of  $m = 100$  yields the best performance for most algorithms under the datasets we have considered. For the proposed Sd-REG-LBFGS method and the SdLBFGS method, a memory size of  $M = 8$  will suffice. Beyond these values, the performance improvement is not so significant. Moreover, the complexity and computational time increases with the two parameters and hence it is desirable to keep them as small as possible. A small amount of regularization, such as  $\gamma = 10^{-4}$ , is adequate to reduce the fluctuations under sufficient samples.

We notice that the proposed Sd-REG-LBFGS and SdLBFGS algorithms gives similar performance with sufficient large number of samples. We shall further compare the two algorithms more rigorously using a statistical test under different settings in Section V-C.

### C. Comparison of classification performance of various algorithms using Statistical Significance Testing

In this section, we employ nonparametric statistical tests - *sign test* and *Wilcoxon paired-difference test* for the evaluation of the statistical significance of whether the proposed algo-

gorithm performs significantly better than the SGD, RSA, SAA and Adam algorithms on average, or vice versa, in terms of classification accuracy. It should be noted that the  $t$ -test may not be a proper choice as classification accuracies (ACC) are bounded and hence they are not normally distributed [48], [49]. First, Table II shows the average classification accuracy for each algorithm over all batch sizes, and the parameters are set for each algorithm as follows:

1. Batch size:  $m = 5, 10, 30, 50, 100, 200$ , for synthetic dataset scenario, and  $m = 5, 10, 20, 30, 50, 100$ , for *scene* dataset scenario;
2. Memory size:  $M = 10$  for Sd-REG-LBFGS and SdLBFGS;
3. Step size constant  $r$ :  $r = 7$  for all algorithms;
4. Regularization parameters:  $\gamma = 10^{-4}$  and  $\delta = 1.25\gamma + 0.01$ .

Here, we abbreviate Sd-REG-LBFGS and SdLBFGS as SRL and SDL for convenience respectively. More precisely, the *sign* test tests the following hypotheses:

$$H_0 : \mu_X - \mu_Y = 0 \text{ vs } H_1 : \mu_X - \mu_Y > 0, \quad (69)$$

where  $\mu_X$  and  $\mu_Y$  are the median classification accuracies of algorithms A and B, respectively. The test statistic of the sign test is given as

$$T_S : \text{number of times that } x_i - y_i > 0, \quad (70)$$

Table II: The average classification accuracy of each algorithm over all batch sizes.

ACC	SRL	SDL	SGD	RSA	SAA	Adam
S1	95.14%	89.67%	66.69%	66.58%	66.70%	50.25%
S2	95.25%	89.78%	67.26%	67.23%	67.26%	50.35%
S3	80.80%	80.48%	77.32%	77.36%	77.32%	65.72%
S4	80.90%	79.12%	76.38%	76.50%	76.38%	66.27%

where  $x_i$  and  $y_i$  are the classification accuracies of algorithms A and B for the  $i$ -th experiment, respectively. The one-sided  $p$ -value can be obtained by a binomial test as  $P = \Pr(T_S \geq t_S | H_0) = \sum_{i=t_S}^n \binom{n}{i} 0.5^n$ , where  $t_S$  is the observed number of times that  $x_i - y_i > 0$ .  $n$  is the total number of experiments performed.

For *Wilcoxon paired-difference test*, the following hypotheses are considered

$$H_0 : |x_i - y_i| \text{ follows a symmetric distribution around zero,} \quad (71)$$

$$H_1 : |x_i - y_i| \text{ does not follow a symmetric distribution around zero.} \quad (72)$$

The test statistic is given as  $T_W = \sum_{i=1}^{n_R} \text{sign}(x_i > y_i) R_i$ , where  $\text{sign}(x > y)$  is defined to be

$$\text{sign}(x > y) = \begin{cases} +1, & \text{if } (x > y) \\ -1, & \text{otherwise,} \end{cases} \quad (73)$$

and  $R_i$  is the rank order of  $|x_i - y_i|$ .  $n_R$  is the number of experiments after excluding those with  $|x_i - y_i| = 0$ . For  $n_R < 20$ , the exact distribution is used. For  $n_R \geq 20$ , a  $z$ -score can be calculated as  $z = T_w / \sigma_W$ , where  $\sigma_W = \sqrt{n_R(n_R + 1)(2n_R + 1)/6}$ . The right-sided  $p$ -value for  $x_i > y_i$  is  $P = \Pr(T_W \geq t_w | H_0)$ , where  $t_w$  is the observed sum of rank. The  $p$ -values are obtained using MATLAB function *signrank*. The batch size is set to  $m = 5$  as our proposed method is robust and efficient in particular for small batch sizes.

The results of *sign test* are shown in Table III. The log  $p$ -values for *Wilcoxon paired-difference test* are shown in Table IV. The batch size is set to  $m = 5$  as our proposed method is robust and efficient in particular for small batch sizes. From the tables, we can see that the proposed approach obtains the highest ACC with statistical significance and the mean difference in ACC between the proposed approach and other algorithms is statistically significant for  $\log p < -1.3$ , (a.k.a.  $p < 0.05$ ). A key observation is that we find that the proposed approach performs much better than the SdLBFGS under small batch size. This is possibly attributed to the incorporation of the proposed regularization scheme, which is useful to improve numerical stability under small sample size. For sufficient samples, the performance of our algorithm is similar to SdLBFGS. Such observations can be found in the sensitivity study of the different parameters, which is omitted here due to page limitation. Interested readers are referred to Section III of the supplementary material for details.

Table III: Right-sided log  $p$  values obtained from *sign test* on mean classification accuracy of various algorithms averaged over 50 Monte Carlo simulations

log $p$	SRL vs SDL	SRL vs SGD	SRL vs RSA	SRL vs SAA	SRL vs Adam
S1	-15.05	-15.05	-15.05	-15.05	-15.05
S2	-15.05	-15.05	-15.05	-15.05	-15.05
S3	-15.05	-15.05	-15.05	-15.05	-15.05
S4	-10.732	-15.05	-15.05	-15.05	-15.05

Table IV: *Wilcoxon paired-difference test* on mean classification accuracy of various algorithms averaged over 50 Monte Carlo simulations

log $p$	SRL vs SDL	SRL vs SGD	SRL vs RSA	SRL vs SAA	SRL vs Adam
S1	-9.40	-9.40	-9.40	-9.40	-9.40
S2	-9.40	-9.41	-9.41	-9.40	-9.40
S3	-9.41	-9.41	-9.41	-9.41	-9.41
S4	-9.02	-9.41	-9.42	-9.41	-9.42

## VI. CONCLUSION

A novel Sd-REG-LBGS method for solving nonconvex and ill-conditioned stochastic optimization problems has been presented. The convergence of the proposed method is established under reasonable assumptions. The effectiveness of the proposed method is studied via the logistic regression and Bayesian logistic regression problems in machine learning for both synthetic and real datasets. The effect of using different algorithmic parameters is also studied. Experimental results show that the proposed Sd-REG-LBFGS method generally outperforms SdLBFGS and exhibits superior performance for problems with small sample sizes. Moreover, the proposed method is less sensitive to the variations of the batch size and memory size than the SdLBFGS method. For future work, we shall consider the extension of our method to distributed optimization [4], [26]–[28] and asynchronous distributed optimization [46], [47].

## REFERENCES

- [1] R. M. Gower, D. Goldfarb, and P. Richtarik, "Stochastic block BFGS: Squeezing more curvature out of data," in *33rd Proc. Int. Conf. Mach. Learn.*, 1869-1878, June 19-24, 2016.
- [2] A. Bordes, L. Bottou, and P. Gallinari, "SGD-QN: Careful Quasi-Newton Stochastic Gradient Descent," *J. Mach. Learn. Res.*, 10, pp. 1737 – 1754, Jul. 2009.
- [3] A. Mokhtari and A. Ribeiro, "RES: Regularized Stochastic BFGS Algorithm," *IEEE Trans. Signal Process.*, vol. 62, no. 23, pp. 6089 – 6104, Dec. 1, 2014.
- [4] M. Eisen, A. Mokhtari and A. Ribeiro, "Decentralized Quasi-Newton Methods," *IEEE Trans. Signal Process.*, vol. 65, no. 10, pp. 2613 – 2628, May, 2017.
- [5] M. Neely, "Distributed Stochastic Optimization via Correlated Scheduling," *IEEE/ACM Trans. Netw.*, vol. 24, no. 2, pp. 759 – 772, April, 2016.
- [6] P. Si, J. Yang, S. Chen, and H. Xi, "Smoothness Constraint Based Stochastic Optimization for Wireless Scalable Video Streaming," *IEEE Commun. Lett.*, vol. 19, no. 5, pp. 759 – 762, May, 2015.
- [7] A. Ribeiro, "Ergodic Stochastic Optimization Algorithms for Wireless Communication and Networking," *IEEE Trans. Signal Process.*, vol. 58, no. 12, pp. 6369 – 6386, Dec, 2010.
- [8] X. Wang, S. Ma, D. Goldfarb, and W. Liu, "Stochastic Quasi-Newton Methods for Nonconvex Stochastic Optimization," *SIAM J. Optim.*, vol. 27, no. 2, pp. 927 – 956, 2017.
- [9] A. Mokhtari, M. Eisen and A. Ribeiro, "IQN: An Incremental Quasi-Newton Method with Local Superlinear Convergence Rate," *SIAM J. Optim.*, vol. 28, no. 2, pp. 1670 – 1698, 2018.

- [10] R. H. Byrd, S. L. Hansen, J. Nocedal, and Y. Singer, "A Stochastic Quasi-Newton Method for Large-Scale Optimization," *SIAM J. Optim.*, vol. 26, no. 2, pp. 1008 - 1031, 2016.
- [11] R. H. Byrd, G. M. Chin, W. Neveitt, and J. Nocedal, "On the Use of Stochastic Hessian Information in Optimization Methods for Machine Learning," *SIAM J. Optim.*, vol. 21, no. 3, pp. 977 - 995, Jan. 2011.
- [12] L. Bottou, F. E. Curtis and J. Nocedal, "Optimization Methods for Large-Scale Machine Learning," *SIAM Rev.*, vol. 60, no. 2, pp. 223 - 311, 2018.
- [13] M. J. D. Powell, "Algorithms for nonlinear constraints that use lagrangian functions," *Math. Programming*, vol. 14, no. 1, pp. 224 - 248, Dec., 1978.
- [14] J. Nocedal, S. J. Wright, *Numerical Optimization*, New York:Springer-Verlag, 1999.
- [15] A. Mokhtari, and A. Ribeiro, "Global Convergence of Online Limited Memory BFGS," *J. Mach. Learn. Res.*, vol. 16, no. 1, pp. 3151 - 3181, Jan., 2015.
- [16] N. Schraudolph, J. Yu, and S. Gunter, "A stochastic quasi-Newton method for online convex optimization," in *Proc. 11th Int. Conf. Artif. Intell. Statist.*, pp. 433 - 440, 2007.
- [17] C. Bishop, *Pattern Recognition and Machine Learning*, Springer New York., 2006.
- [18] C. Wang, and D. M. Blei, "Variational Inference in Nonconjugate Models," *J. Mach. Learn. Res.*, vol. 14, no. 1, pp. 1005 - 1031, Jan., 2013.
- [19] S. Ghadimi and G. Lan, "Stochastic First- and Zeroth-Order Methods for Nonconvex Stochastic Programming," *SIAM J. Optim.*, vol. 23, no. 4, pp. 2341 - 2368, 2013.
- [20] A. Nemirovski and A. Juditsky and G. Lan and A. Shapiro, "Robust Stochastic Approximation Approach to Stochastic Programming," *SIAM J. Optim.*, vol. 19, no. 4, pp. 1574 - 1609, 2009.
- [21] C. Dang and G. Lan, "Stochastic Block Mirror Descent Methods for Nonsmooth and Stochastic Optimization," *SIAM J. Optim.*, vol. 25, no. 2, pp. 856 - 881, 2015.
- [22] H. Robbins and S. Monro, "A Stochastic Approximation Method," *Ann. Math. Statist.*, vol. 22, no. 3, pp. 400 - 407, 1951.
- [23] M. D. Hoffman, D. M. Blei, C. Wang and J. Paisley, "Stochastic Variational Inference," *J. Mach. Learn. Res.*, vol. 14, no. 1, pp. 1303 - 1347, Jan., 2013.
- [24] Z. Ghahramani, "Probabilistic machine learning and artificial intelligence," *Nature*, vol. 521, no. 7553, pp. 452 - 459, 2013.
- [25] D. M. Blei, A. Kucukelbir and J. D. McAuliffe, "Variational Inference: A Review for Statisticians," *J. Am. Statist. Assoc.*, vol. 112, no. 518, pp. 859-877, 2017.
- [26] L. Zhang, H. C. Wu, C. H. Ho, S. C. Chan, "A Multi-Laplacian Prior and Augmented Lagrangian Approach to the Exploratory Analysis of Time-Varying Gene and Transcriptional Regulatory Networks for Gene Microarray Data , to appear in " *IEEE/ACM Trans. Comput. Biol. Bioinf.*
- [27] S. C. Chan, L. Zhang, H. C. Wu, and K. M. Tsui, "A maximum a posteriori probability and time-varying approach for inferring gene regulatory networks from time course gene microarray data," *IEEE/ACM Trans. Comput. Biol. Bioinf.*, vol. 12, no. 1, pp. 123-135, 2015.
- [28] S. C. Chan, H. C. Wu, C. H. Ho and L. Zhang, "An Augmented Lagrangian Approach for Distributed Robust Estimation in Large-Scale Systems, to appear in " *IEEE Systems Journal*.
- [29] J. Paisley, D. M. Blei and M. I. Jordan, "Variational Bayesian Inference with Stochastic Search," in *29th Proc. Int. Conf. Mach. Learn.*, vol. 14, pp. 1363 - 1370, 2012.
- [30] Sun Yi, D. Wierstra, T. Schaul and J. Schmidhuber, "Stochastic search using the natural gradient," in *26th Proc. Int. Conf. Mach. Learn.*, vol. 382, pp. 1161 - 1168, 2009.
- [31] R. Johnson and T. Zhang, "Accelerating Stochastic Gradient Descent Using Predictive Variance Reduction," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 1, pp. 315 - 323, 2013.
- [32] D. C. Liu and J. Nocedal, "On the limited memory BFGS method for large scale optimization," *Math. Programming*, vol. 45, no. 1, pp. 503 - 528, Aug., 1989.
- [33] P. Moritz and R. Nishihara and M. I. Jordan, "A Linearly-Convergent Stochastic L-BFGS Algorithm," in *Proc. 19th Int. Conf. Artif. Intell. Statist.*, vol. 51, pp. 249 - 258, 2016.
- [34] J. Taghia and A. Leijon, "Variational Inference for Watson Mixture Model," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 9, pp. 1886 - 1900, Sept., 2016.
- [35] A. Honkela, T. Raiko, M. Kuusela, M. Tornio and Juha Karhunen, "Approximate Riemannian Conjugate Gradient Learning for Fixed-Form Variational Bayes," *J. Mach. Learn. Res.*, vol. 11, pp. 3235 - 3268, Dec., 2010.
- [36] S. Amari, *Information Geometry and Its Applications*, Springer Japan, 2016.
- [37] S. Boyd, and L. Vandenberghe, *Convex Optimization*, Cambridge University Press New York., 2004.
- [38] M.R. Boutell, J. Luo, X. Shen, and C.M. Brown, "Learning multi-label scene classification," *Pattern Recognition*, vol. 37, no. 9, pp. 1757-1771, 2004.
- [39] B. T. Polyak and A. B. Juditsky, "Acceleration of stochastic approximation by averaging," *SIAM J. Control Optim.*, vol. 30, no. 4, pp. 838-855, 1992.
- [40] D. Kingma and J. Ba, "Adam: A method for stochastic optimization", in *3rd International Conference for Learning Representations, 2015*.
- [41] V. Arzamasov, K. Böhm and P. Jochem, "Towards Concise Models of Grid Stability", *IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids*, Oct. 9-31, 2018.
- [42] R. Bhatt, "Fuzzy-Rough Approaches for Pattern Classification: Hybrid measures, Mathematical analysis, Feature selection algorithms, Decision tree algorithms, Neural learning, and Applications", Amazon Books.
- [43] J. Rohra, B. Perumal, S. Narayanan, P. Thakur and R. Bhatt, "User Localization in an Indoor Environment Using Fuzzy Hybrid of Particle Swarm Optimization & Gravitational Search Algorithm with Neural Networks", in *Proceedings of Sixth International Conference on Soft Computing for Problem Solving*, pp. 286-295, 2017.
- [44] V. G. Sigillito, S. P. Wing, L. V. Hutton, and K. B. Baker, "Classification of radar returns from the ionosphere using neural networks. Johns Hopkins APL Technical Digest", Johns Hopkins APL Technical Digest, 10, 262-266, 1989.
- [45] V. Lohweg and H. Doerksen, "Banknote authentication data set", submitted.
- [46] R. Zhang and T. Kwok, "Asynchronous distributed ADMM for consensus optimization", in *Proc. of the 31st Int. Conf. Mach. Learn.*, vol. 32, pp. 1701-1709, June, 2014.
- [47] R. Zhu, D. Niu and Z. Li, "A Block-wise, Asynchronous and Distributed ADMM Algorithm for General Form Consensus Optimization", in *arXiv:1802.08882*, Feb. 2018.
- [48] T. G. Dietterich, "Approximate statistical tests for comparing supervised classification learning algorithms", *Neural Computation*, vol. 10, pp. 1895-1924, Oct. 1998.
- [49] J. Demšar, "Statistical Comparisons of Classifiers over Multiple Data Sets", *J. Mach. Learn. Res.*, vol. 7, pp. 1- 30, Jan. 2006.
- [50] M. Chen, B. Amos, L. Watson, J. Tyson, Y. Cao, C. Shaffer, M. Trosset, C. Oguz, and G. Kakoti, "Quasi-Newton Stochastic Optimization Algorithm for Parameter Estimation of a Stochastic Model of the Budding Yeast Cell Cycle," *IEEE/ACM Trans. Comput. Biol. Bioinf.*, vol. 16, no. 1, pp. 301-311, Nov. 2017.
- [51] S. Huang, Y. Sun, and Q. Wu, "Stochastic Economic Dispatch With Wind Using Versatile Probability Distribution and L-BFGS-B Based Dual Decomposition," *IEEE Trans. Power Syst.*, vol. 33, no. 6, pp. 6254-6263, Nov. 2018.
- [52] J. Rafati and R. Marcia, "Improving L-BFGS Initialization For Trust-Region Methods In Deep Learning," in *17th IEEE Int. Conf. Mach. Learn. App.*, Dec. 27-20, 2018.
- [53] S. Scardapane and P. Lorenzo, "Stochastic Training of Neural Networks via Successive Convex Approximations," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 10, Oct. 2018.
- [54] A. Jalilzadeh, A. Nedić, U. Shanbhag and F. Yousefian, "A Variable Sample-size Stochastic Quasi-Newton Method for Smooth and Nonsmooth Stochastic Convex Optimization," *IEEE Conference on Decision and Control*, Dec. 17-19 2018.

This figure "HCWu.png" is available in "png" format from:

<http://arxiv.org/ps/1912.04456v1>

This figure "WHLAM\_photo\_1.jpg" is available in "jpg" format from:

<http://arxiv.org/ps/1912.04456v1>

This figure "hmchen.png" is available in "png" format from:

<http://arxiv.org/ps/1912.04456v1>

This figure "scchan.png" is available in "png" format from:

<http://arxiv.org/ps/1912.04456v1>