This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2019.2935900, IEEE Access

Digital Object Identifier

On the Convergence Analysis of Cubic Regularized Symmetric Rank-1 Quasi-Newton Method and the Incremental Version in the Application of Large-Scale Problems

HUIMING CHEN[1], WONG-HING LAM[1], (SENIOR MEMBER, IEEE), AND SHING-CHOW CHAN[1], (Member, IEEE)
[1]Department of Electrical and Electronic Engineering, The University of Hong Kong, Hong Kong SAR, China.
Corresponding author: Huiming Chen (e-mail: hmchen@eee.hku.hk).

ABSTRACT This paper provides a detailed study on the convergence properties of the cubic regularized symmetric rank-1 (SR1) method (CuREG-SR1) proposed in [2]. The main advantage of incorporating cubic regularization technique to SR1 is to alleviate the problem of indefinite resulting matrix in SR1. However, its convergence under the line search framework is less studied. Here, we first show that CuREG-SR1 converges to a first-order critical point. Moreover, we give a novel result that provided the uniformly independent assumption, the difference between approximated Hessian matrix generated by CuREG-SR1 and the true Hessian is bounded. In addition, we show that for a problem with dimension d, CuREG-SR1 generates $q - d$ superlinear steps every q iterations. We also propose a novel incremental CuREG-SR1 (ICuREG-SR1) algorithm to adapt SR1 and CuREG-SR1 efficiently for solving large scale problems. The basic idea is to incorporate incremental optimization scheme, which updates progressively information for objective function involving a sum of individual functions, which are frequently encountered in large-scale machine learning. Numerical experiments on several machine learning problems show that the proposed algorithm offers superior performance in terms of the gradient magnitude than other conventional algorithms tested.

INDEX TERMS quasi-Newton method, symmetric rank-1, superlinear convergence rate, cubic regularization, incremental optimization.

I. INTRODUCTION

MANY practical engineering problems can be formulated as the solution of unconstrained or constrained optimization problems, e.g., computational biology [18], [19], wireless communications [16], [17] and machine learning [35], [41], etc. In this paper, we are concerned with quasi-Newton methods for the unconstrained optimization problems:

$$\min_{x\in\mathbb{R}^n} f(x), \tag{1}$$

where $f(x)$ is the objective function and $x \in \mathbb{R}^n$ is the optimizing variable with dimension n.

A variety of quasi-Newton methods solves the problem (1) by using a quadratic model of (1). To be specific, at each iteration k, the objective function $f(x)$ is approximated at current iterate x_k by applying the second-order Taylor series with an approximation of the Hessian matrix in lieu of true Hessian matrix. A classical form of the quasi-Newton method is as follows:

$$x_{k+1} = x_k - \lambda_k B_k^{-1} \nabla f(x_k), \tag{2}$$

where B_k is the approximated Hessian matrix and λ_k is the stepsize chosen by standard line search algorithm.

To obtain a descent direction of the objective funtion in terms of the approximated quadratic model, it requires to solve the corresponding linear system at each iteration. If the true Hessian matrix is used, the second-order derivatives need to be evaluated in addition to the solution of the linear system and matrix inversion, which can be computationally expensive especially when the objective function

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2019.2935900, IEEE Access

is complicated to evaluate, say via physical situation and vice versa. Therefore, various recursive update schemes for the approximated Hessian matrix have been proposed to alleviate the intensive computation: These include symmetric rank-one (SR1) update [1], [3], [5] and rank-two variants such as the Davidon-Fletcher-Powell (DFP) scheme [6], [7] and the Broyden-Fletcher-Goldfarb-Shanno (BFGS) update scheme [4], [8]. For the rank-two methods in which B_{k+1} is obtained by adding a matrix of rank at most two, the DFP update has been shown (under mild condition) to generate a sequence $\{x_k\}$ which will converge to a local optimal point x^* at Q-superlinear convergence rate when implemented with stepsize chosen by standard line search criterion. Moreover, the Hessian approximation matrix sequence $\{B_k\}$ generated by the DFP update does not have the property of converging to the true Hessian matrix, which is a standard technique for proving a method with Q-superlinear convergence rate. In fact, the consistency condition that $\{B_k\}$ converges to $\nabla^2 f(x^*)$ is sufficient but not necessary. The DFP updating formula is quite effective and it has been applied recently to training of complex-valued neural networks for pattern recognition and signal processing [9]. However, the DFP method was soon superseded by the BFGS formula [23], which is considered the most commonly applied quasi-Newton method. A good property of BFGS method is that the generated sequence $\{B_k\}$remains positive definite in strongly convex problems, which guarantees a descent direction. The convergence properties which is similar to DFP method can be found in [4].

For rank-one update, the SR1 scheme requires less computation at each iteration. The recursion formula is given by:

$$B_{k+1} = B_k + \frac{(y_k - B_k s_k)(y_k - B_k s_k)^T}{s_k^T (y_k - B_k s_k)}, \qquad (3)$$

where $s_k = x_{k+1} - x_k$ and $y_k = \nabla f(x_{k+1}) - \nabla f(x_k)$. Computational experiments have shown that the SR1 is very competitive compared to BFGS method [3], and it appears to be substantially more efficient in the trust region framework than any other quasi-Newton method tested [5]. However, to the best of our knowledge, very few existing works have studied the convergence analysis of SR1 method. On the other hand, [5] studied the convergence of SR1 in the framework of both trust region and line search. The proof technique is based on the assumption of uniformly independence of the sequence $\{s_k\}$. The result shows that the sequence $\{B_k\}$ generated by SR1 method converges to explicit Hessian matrix. Nevertheless, the condition of uniform linear independence of $\{s_k\}$ is too strong to hold in many practical problems. Thus, the result that the sequence $\{B_k\}$converges to the true Hessian matrix may not hold in general. [3] further studies the convergence properties of SR1 without assuming that the sequence $\{s_k\}$ uniformly independent and the result shows that the SR1 update with standard line search framework exhibits $(n+1)$-step Q-superlinear and $2n$-step quadratic convergence rate. Unlike BFGS update, one of the drawbacks of the SR1 update is that it can generate indefinite matrix B_k even in strongly convex problems, which may result in a non-descent direction.

In the big data era, large scale optimization problems with large number measurements and variables will be increasingly popular. Since the gradient evaluation is proportional to the number of measurements, the computational cost can be huge. Moreover, the storage of the gradients of the loss functions can cost large amount of memory. Thus, there is a growing interest in developing efficient stochastic optimization algorithms. The main motivation is to obtain simple and yet unbiased estimator of the full gradient [36].Stochastic BFGS methods have been among the most popular stochastic quasi-Newton (SQN) methods and adapted to solve large scale problems [13]–[15], [37]. In strongly convex problems, the sequence $\{B_k\}$ generated by the stochastic BFGS method is guaranteed to be positive definite.

In [15], a SQN method is proposed to solve large scale strongly convex problems. Since the quality of the curvature estimate can be difficult to control under the stochastic regime, an efficient subsampled Hessian-vector product is proposed based on the limited memory BFGS (LBFGS) method, which artfully avoids double evaluating the gradients. In [13], a general framework of SQN methods is proposed to solve nonconvex optimization problems. Since the problem of how to preserve positive definiteness of the Hessian approximation in nonconvex problem is challenging, a stochastic damped BFGS has been proposed to remedy the problem, in which the damped BFGS is based on [21]. Moreover, it has been shown that the Hessian approximation matrix can be singular or near singular and thus the norm of the Hessian inverse approximation are not uniformly bounded, which harms the convergence.

In [12], a regularized stochastic BFGS (RES) method is proposed to alleviate the problem. It modifies the proximity condition of BFGS to ensure that the norm of the Hessian approximation is above a specified level and thus uniformly bounded. For the class of problems aiming to minimize the objective function written in a large sum of strongly convex functions, an incremental quasi-Newton methodology (IQN) [14] is proposed to alleviate the high computational cost at each iteration. In lieu of random selection of a single function, incremental methods choose a single function for efficient implementation via both the BFGS method and iterately update in a cyclic fashion. Therefore, computational cost at each iteration is substantially reduced at the expenses of slower convergence speed. The aggregated gradients of all functions are able to reduce the noise of gradient approximation, which makes the IQN method local convergence at a superlinear rate.

Most studies on the quasi-Newton methods for solving large scale problems are based on the BFGS method and its efficient variants. However, to the best of our knowledge, few works have been developed for the SR1 method for solving large scale problems. This may attribute to the two major drawbacks of SR1 update formula: (i) The denominator of SR1 recursion formula in (3) may vanish; (ii) The result-

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2019.2935900, IEEE Access

ing Hessian approximation matrix can be indefinite even in strongly convex problems, which leads to non-descent direction of the step. To remedy problem (i), a standard technique is to skip the SR1 update if

$$|s_k^T(y_k - B_k s_k)| < \epsilon \|s_k\| \|y_k - B_k s_k\|, \tag{4}$$

where the value of the constant ϵ is normally chosen as 10^{-8}. Provided the Hessian approximation matrix is positive definite, skipping SR1 update is reasonable in practice since the current positive definite B_k still results in sufficient reduction in objective function. For the purpose of alleviating problem (ii), [2] has proposed a cubic regularized SR1 (CuREG-SR1) method. The main strategy is to calculate the gradient difference based on cubic approximation of the objective function. By choosing a suitable cubic parameter which satisfies the specified condition, SR1 recursion formula can lead to positive definite Hessian approximation matrix. Moreover, cubic regularization technique has been studied in [10], [11]. Since cubic regularized method has avoided the problem of indefinite Hessian approximation matrix, it has improved performance over standard line search criterion. However, the current study on the convergence properties of cubic regularized SR1 method is limited. Besides, in the big data era, how to develop efficient algorithms to solve large scale problem with massive data has attracted much attention recently. Hence, in this paper, we mainly focus on two problems, the first is the convergence analysis of CuREG-SR1 algorithm, and the second is to develop efficient algorithm based on SR1 to solve large scale problems.

Our main contributions are as follows:

- Based on the line search framework and the motivation of CuREG-SR1 [2], we show that CuREG-SR1 converges to a first-order critical point (see *Theorem 1*).
- Novel results on the convergence rate of the algorithm are derived. We first propose *Lemma 1* which shows that under specific assumptions, the quantity $\|y_j - B_i s_j\|$ for $i \leq j-1$ is bounded above. Based on *Lemma 1* and using the assumption that the sequence $\{s_k\}$ is uniformly independent, we obtain a novel result that the difference between the approximated Hessian matrix and the true Hessian is bounded above (see *Theorem 2*).
- Since the uniformly independent assumption is too strong in practice, we show that the quantity $\frac{\|[B_{k_p} - \nabla^2 f(x^*)]s_{k_p}\|}{\|s_{k_p}\|}$, which is closely related to superlinear convergence rate, is small with other reasonable assumptions (see *Lemma 3*). Based on the above result, we show that for every $q \geq d+1$ iterations, the CuREG-SR1 algorithm generates at least $q-d$ superlinear steps (see *Theorem 3*).
- A novel incremental optimization method based on the SR1 and CuREG-SR1 is proposed. The main idea is to update the information of a selected set of individual functions in the objective function involving a large sum of component functions at each iteration, while others remain intact as in previous iteration. Numerical experiments show that our proposed algorithm offers superior performance in terms of the gradient magnitude than other conventional algorithms tested.

The rest of the paper is organized as follows: Section II reviews the general CuREG-SR1 algorithm based on the line search framework. In Section III, we provide a detailed convergence analysis of CuREG-SR1 algorithm. In Section IV, we propose a novel ICuREG-SR1 algorithm ICuREG-SR1. Numerical experiments are conducted to evaluate the performance of the proposed ICuREG-SR1 algorithm in Section V. The conclusion is drawn in Section VI. Moreover, we have included the comparison of our proposed method with the state-of-art approaches Adam [52] and RMSProp [53] in supplementary material.

Mathematical Notation: we use $\|a\|$ to denote the Euclidean norm of vector a and $\|A\| := \max\{\frac{\|Ax\|}{\|x\|}\}$ to denote the matrix norm of a matrix A. $A \succeq B$ indicates the matrix $A - B$ is positive semidefinite. The identity matrix with appropriate dimension is signified as I.

II. ALGORITHM DEFINITION

In this paper, we consider the following unconstrained optimization problems: $x^* = \text{argmin}_{x\in\mathbb{R}^d} f(x)$, where d is the dimension of the variable x and $f : \mathbb{R}^d \to \mathbb{R}$ is the objective function. In this section, we first review the conventional quasi-Newton methods using the line search framework. Specifically, we shall consider the derivation of SR1 update and provide the cubically regularized SR1 method. Then, the cubic regularization technique is briefly introduced. Incorporation of the cubic regularization technique to SR1 method leads to CuREG-SR1. We will also discuss the condition for Hessian approximation update resulted from CuREG-SR1 to be positive definite.

A. LINE SEARCH FRAMEWORK

In classical line search method, a search direction $p_k \in \mathbb{R}^d$ that can sufficiently reduce the objective function is computed at each iteration via the following update:

$$x_{k+1} = x_k + \lambda_k p_k, \tag{5}$$

where λ_k is the stepsize that decides how far to move along the search direction. An ideal choice of the stepsize is to solve the sub problem: $\lambda_k^* = \text{argmin}_{\lambda\in\mathbb{R}} \varphi(\lambda) := f(x_k + \lambda p_k)$, However, it is computationally expensive in general to seek the exact one dimensional minimum, since it may require many evaluations of the objective function f and the gradient ∇f at each iteration. In practice, various termination criteria for inexact line search have been proposed, which aims to achieve a satisfactory reduction in the objective function, without spending too much effort in computation. The inexact line search based on the following Wolfe condition is widely used:

$$f(x_k + \lambda p_k) < f(x_k) + c_1 \lambda \nabla f(x_k)^T p_k, \tag{6}$$

$$\nabla f(x_k + \lambda p_k)^T p_k \geq c_2 \nabla f(x_k)^T p_k, \tag{7}$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2019.2935900, IEEE Access

where $c_1 \in (0,1)$ and $c_2 \in (c_1,1)$ are constants. The inequality in (6) is also known as Armijo condition. Intuitively, inequality (6) means that the updated objective function $f(x_k+\lambda p_k)$ lies below the linear function $l(\lambda) := f(x_k) + c_1\lambda\nabla f(x_k)^T p_k$, which has a negative slope $c_1\nabla f(x_k)^T p_k$ with respect to λ to ensure that the function will decrease at least at a certain rate (since p_k is a decrease direction and c_1 is a positive constant, $c_1\nabla f(x_k)^T p_k < 0$ holds). Moreover, since $f(x_k+\lambda p_k) < f(x_k)$, it indicates that for small λ, the inequality $\varphi(\lambda) < l(\lambda)$ holds, i.e., the Armijo condition is always satisfied at small stepsize, this may cause insufficient reduction when the resultant stepsize is small. The Armijo condition is thus insufficient to guarantee a reasonable progress. Therefore, the curvature condition (7) is introduced, which is called curvature condition. Note the left hand side of (7) is the derivative of the function $\varphi(\cdot)$ at λ, i.e., $\varphi'(\lambda) = \nabla f(x_k+\lambda p_k)^T p_k$. Similarly for the right hand side, $\varphi'(0) = \nabla f(x_k)^T p_k$. Consequently, for a stepsize chosen to ensure significant decrease along the search direction, it implies that the corresponding slope $\varphi'(\lambda)$ should be intuitively less slightly negative than $\varphi'(0)$. It makes sense since we have the intuition that at a point that results in sufficient reduction, the related slope is negatively flatter. Thus, combining the Armijo condition and curvature condition, the algorithm can make reasonable progress.

B. SR1 QUASI-NEWTON METHODS

For Newton's method, the search direction p_k is obtained by solving the system: $\nabla^2 f(x_k)p_k = -\nabla f(x_k)$. However, two drawbacks make Newton's method impractical: (i). evaluation of second-order derivative at each iteration is generally too expensive; (ii). solving the system takes much time and effort, and matrix inversion or factorization increase the arithmetic complexity, which can be computationally huge for large scale problem. Therefore, we consider the quasi-Newton methods, which seek for an approximate Hessian matrix B_k. We now turn our attention to SR1 update (3). It has been shown that if the second-order derivative of the objective function is Lipschitz continuous and the sequence $\{\frac{s_k}{\|s_k\|}\}$ (recall s_k is defined as $s_k := x_{k+1} - x_k$) is uniformly linearly independent, the sequence $\{B_k\}$ generated by SR1 formula converges to the true Hessian matrix at optimal point, i.e., $\lim_{k\to\infty} \left\|B_k - \nabla^2 f(x^*)\right\| = 0$ [3], [5]. The convergence results generally involve the following assumptions.

Assumption1: The objective function $f(x)$ is twice continuously differentiable.

Assumption2: The first-order derivative $\nabla f(x)$ is Lipschitz continuous with $L' > 0$, i.e., for all $x, y \in \mathbb{R}^d$, the following inequality holds:

$$\|\nabla f(x) - \nabla f(y)\| < L' \|x - y\|. \tag{8}$$

Assumption3: The second-order derivative $\nabla^2 f(x)$ is Lipschitz continuous with $L'' > 0$, i.e., for all $x, y \in \mathbb{R}^d$, the following inequality holds:

$$\left\|\nabla^2 f(x) - \nabla^2 f(y)\right\| < L'' \|x - y\|. \tag{9}$$

Assumption4: A sequence $\{s_k\}$ in $\mathbb{R}^d$ is defined to be uniformly linearly independent, if there exists a positive constant $\tau > 0$, an integer k_0 and $m \geq d$ such that for each $k \geq k_0$, one can choose d distinct indices between k and $k+m$, namely $k \leq k_1 < \cdots < k_d \leq k+m$, such that $\sigma_{\min}(S_k) \geq \tau$, where $\sigma_{\min}(S_k)$ is the minimum singular value of the matrix

$$S_k = [\frac{s_{k_1}}{\|s_{k_1}\|}, \cdots, \frac{s_{k_d}}{\|s_{k_d}\|}]. \tag{10}$$

Next, we first follow the development in [22] to briefly review the derivation of the SR1 formula, which motivates the CuREG-SR1 formula. Recall y_k is defined as:

$$y_k := \nabla f(x_{k+1}) - \nabla f(x_k), \tag{11}$$

By using first-order Taylor expansion, we can further obtain:

$$y_k = \nabla^2 f(x_k)s_k + o(\|s_k\|). \tag{12}$$

For a quadratic function $f(x) = \frac{1}{2}x^T Ax + b^T x + c$, where A is positive definite, $b, c \in \mathbb{R}^d$, it can be verified straightforwardly that the higher order term is zero. In this case, we can set $B_k = A$ and obtain the secant equation $y_k \approx B_k s_k$. Though the secant equation does not hold true for the general nonlinear functions due to the higher order term, it serves as a useful condition for updating the Hessian approximation matrix given s_k and y_k at each iteration:

$$y_k = B_{k+1}s_k. \tag{13}$$

For rank one update, the recursion formula has the following form, and the resulting matrix is symmetric:

$$B_{k+1} = B_k + \rho v v^T, \tag{14}$$

Combining the secant equation in (13), it follows that:

$$y_k = B_k s_k + \rho v v^T s_k \tag{15}$$

Note from the right hand side in (15) that $v^T s_k$ is a scalar, hence the vector v has the same direction with $(y_k - B_k s_k)$ for some scalar ϑ. Therefore, we can simply set $v = \vartheta(y_k - B_k s_k)$. It subsequently leads to the following equation:

$$y_k - B_k s_k = \rho\vartheta^2[s_k^T(y_k - B_k s_k)](y_k - B_k s_k), \tag{16}$$

where $s_k^T(y_k - B_k s_k) \neq 0$. Comparing the LHS and RHS, there only exists two possible scenarios: (i). $s_k^T(y_k - B_k s_k) < 0$, then $\rho = -1$ and $\vartheta^2 = [-s_k^T(y_k - B_k s_k)]^{-1}$; (ii). $s_k^T(y_k - B_k s_k) > 0$, then $\rho = 1$ and $\vartheta^2 = [s_k^T(y_k - B_k s_k)]^{-1}$. Substituting the results, we obtain the SR1 formula as follows:

$$B_{k+1} = B_k + \frac{(y_k - B_k s_k)(y_k - B_k s_k)^T}{(y_k - B_k s_k)^T s_k}. \tag{17}$$

Subsequently, the search direction p_k can be calculated by solving the system $B_k p_k = -\nabla f(x_k)$. Moreover, if the inverse Hessian approximation matrix denoted by H_k is positive definite, it can also be updated by employing the Sherman-Morrison formula

$$H_{k+1} = H_k + \frac{(s_k - H_k y_k)(s_k - H_k y_k)^T}{(s_k - H_k y_k)^T y_k}. \tag{18}$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2019.2935900, IEEE Access

Therefore, instead of solving the system $B_k p_k = -\nabla f(x_k)$ by matrix inverse operation, we can simply obtain the search direction $p_k = -H_k \nabla f(x_k)$ through efficient matrix-vector product, making the algorithm computationally very attractive. As mentioned above, when the sequence $\{x_k\}$ converges to the optimal point with Assumption 3, the Hessian approximation sequence generated by (17) will converge to the true Hessian matrix. However, the denominator tends to zero if $\{x_k\}$ converges to x^*, which implies that all the future updates will be dominated by the update matrix and violate the uniform linearly independent assumption [2]. Hence, to alleviate both the theoretical and practical problems, the Hessian approximation matrix skips the update whenever the denominator is too small:

$$|(y_k - B_k s_k)^T s_k| < \epsilon \|y_k - B_k s_k\| \|s_k\|, \qquad (19)$$

i.e., at each iteration, set $B_{k+1} = B_k$ whenever (19) is satisfied while otherwise, B_{k+1} is calculated via (17). The skipping scenario has been shown to be effective since we can still get the descent direction if B_k is positive definite.

C. CUBIC REGULARIZATION TECHNIQUE

It should be noted that even if B_k is positive definite, B_{k+1} can still be indefinite since the denominator in (17) can be negative. In [2], the cubic regularized technique has been proposed to address this issue. We shall first review the cubic regularization technique [10], [11].

Recall that the search direction in quasi-Newton method for problem (1) is obtained by solving $p_k = \operatorname{argmin}_{p\in\mathbb{R}^d} f(x_k) + \nabla f(x_k)^T p + \frac{1}{2} p^T B_k p$. Similarly in cubic model for the objective function $f(x)$ with Lipschitz continuous $\nabla^2 f(x)$, the search direction p is calculated via minimizing the cubic model $m_k^c(p)$:

$$m_k^c(p) := f(x_k) + \nabla f(x_k)^T p + \frac{1}{2} p^T B_k p + \frac{M_k}{6} \|p\|^3. \qquad (20)$$

In general, M_k is chosen to satisfy $M_k \leq L''$. It should be noted that minimizing the cubic model is a non-convex problem and it can have local minima [11]. Here, the search direction $p_k \in \operatorname{Argmin}_{p\in\mathbb{R}^d} m_k^c(p)$ means that p_k is a global minimizer of the cubic model $m_k^c(p)$. It has been shown in [10] that p_k^* is a global minimizer of the problem if and only if $\nabla m_k^c(p_k^*) = 0$ and $B_k + \frac{1}{2} M_k \|p_k^*\| \succeq 0$. The necessary and sufficient condition has provided us a way to compute a global minimizer of the cubic model. However, from a computational point of view, doing so can be prohibitively sophisticated. In fact, an approximate solution to the global minimizer can make the algorithm progress well with less computational complexity. Specifically, for the framework of adaptive regularization using cubics (ARC) proposed in [10], the search direction p_k is only required to ensure the decrease in the cubic model at least as good as that produced by the corresponding Cauchy point. Because in this way and with the condition that $\nabla f(x)$ is uniformly continuous on the sequence $\{x_k\}$, the ARC algorithm has been shown to converge to the first-order critical point, i.e., $\lim_{k\to\infty} \|\nabla f(x_k)\| = 0$. Hence, the more efficient Krylov method can be applied to well approximate the global solution.

D. CUBIC REGULARIZED SR1 METHOD

We have briefly introduced the cubic regularization technique. Let us turn our attention to SR1 formula (17). As we mentioned above, one of the drawbacks that SR1 formula is that the updated Hessian approximation matrix can be indefinite. Cubic regularization technique incorporated into SR1 has been proved theoretically and computationally effective to alleviate such concern with light extra calculation. In fact, if B_k is not positive definite, the well-known Levenberg-Marquardt regularization as suggested in [24], [25] can also overcome the problem. The main ingredient is to regularize B_k with a unit matrix times a specified scalar β and the resultant iteration is:

$$x_{k+1} = x_k - (B_k + \beta I)^{-1} \nabla f(x_k). \qquad (21)$$

To be specific, when the smallest eigenvalue $\lambda_{min}(B_k)$ of the matrix B_k is non-positive, the regularized parameter β should be chosen to satisfy $\beta > |\lambda_{min}(B_k)|$. Doing so implies complicated computation since we need to factor a dense matrix B_k at each iteration.

For incorporating cubic regularization technique into SR1, we start with the secant equation (13), namely $y_k = B_{k+1} s_k$. Recall the definition of $y_k := \nabla f(x_{k+1}) - \nabla f(x_k)$, replace $f(x_{k+1})$ with the cubic model

$$m_k^c(x_{k+1} - x_k) = f(x_k) + \nabla f(x_k)^T (x_{k+1} - x_k) + \frac{1}{2}(x_{k+1} - x_k)^T B_k (x_{k+1} - x_k) + \frac{1}{6} M_k \|x_{k+1} - x_k\|^3, \qquad (22)$$

it follows that:

$$\begin{aligned} y_k^c &= \nabla m_k^c(x_{k+1} - x_k) - \nabla m_k^c(0) \\ &= B_k(x_{k+1} - x_k) + \frac{M_k}{2} \|s_k\| s_k \\ &= B_k s_k + \frac{1}{2} \|s_k\| s_k \\ &\approx y_k + \frac{M_k}{2} \|s_k\| s_k \end{aligned} \qquad (23)$$

Therefore, the secant equation is modified to

$$y_k = B_{k+1} s_k - \frac{M_k}{2} \|s_k\| s_k. \qquad (24)$$

By substituting the above equation into the rank-1 update $B_{k+1} = B_k + \rho v v^T$, we can obtain $v = \vartheta[y_k + (\frac{M_k}{2}\|s_k\| I - B_k)s_k]$ for some scalar ϑ. Thus, there exists two possible scenarios (i). if $s_k^T[y_k + (\frac{M_k}{2}\|s_k\| I - B_k)s_k] < 0$, then $\rho = -1$ and $\vartheta^2 = \{-s_k^T[y_k + (\frac{M_k}{2}\|s_k\| I - B_k)s_k]\}^{-1}$; (ii). if $s_k^T[y_k + (\frac{M_k}{2}\|s_k\| I - B_k)s_k] > 0$, then $\rho = 1$ and $\vartheta^2 = \{s_k^T[y_k + (\frac{M_k}{2}\|s_k\| I - B_k)s_k]\}^{-1}$. According to the above discussion, the CuREG-SR1 update formula can be derived as follows:

$$B_{k+1} = B_k + \frac{[y_k + (\frac{M_k}{2}\|s_k\| I - B_k)s_k]}{[y_k + (\frac{M_k}{2}\|s_k\| I - B_k)s_k]^T s_k} \cdot [y_k + (\frac{M_k}{2}\|s_k\| I - B_k)s_k]^T. \qquad (25)$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2019.2935900, IEEE Access

Since it is necessary to solve a linear system $B_k p_k = -\nabla f(x_k)$ at each iteration to obtain the search direction, the inverse Hessian approximation matrix can be updated via Sherman-Morrison formula as follows

$$H_{k+1} = H_k + \frac{[s_k - H_k(y_k + \frac{M_k}{2}\|s_k\| s_k)]}{[s_k - H_k(y_k + \frac{M_k}{2}\|s_k\| s_k)]^T(y_k + \frac{M_k}{2}\|s_k\| s_k)} \cdot [s_k - H_k(y_k + \frac{M_k}{2}\|s_k\| s_k)]^T. \tag{26}$$

In this way, direct matrix inverse operation is avoided and the computation is substantially reduced.

Recall that the main purpose to incorporate cubic regularization technique into SR1 formula is to efficiently avoid the indefinite update. Let us consider the formula (26) and assume H_k is positive definite. Since the numerator is the product of a vector and its transpose, the resultant matrix is positive semidefinite with rank one. Hence, the only way for the updated H_{k+1} to be indefinite is that the denominator is non-positive, i.e., $[s_k - H_k(y_k + \frac{M_k}{2}\|s_k\| s_k)]^T(y_k + \frac{M_k}{2}\|s_k\| s_k) < 0$. Subsequently, by suitably choosing the cubic regularized parameter M_k, the denominator can be positive and the resultant update matrix H_{k+1} is ensured positive definite. Even when the denominator is slightly negative which still results in positive definite update, we enforce the denominator to be positive:

$$[s_k - H_k(y_k + \frac{M_k}{2}\|s_k\| s_k)]^T(y_k + \frac{M_k}{2}\|s_k\| s_k) > 0. \tag{27}$$

By straightforward calculation and regrouping terms, we obtain a quadratic inequality with respect to M_k: $aM_k^2 + bM_k + c < 0$, where

$$a = \frac{s_k^T H_k s_k}{4}\|s_k\|^2, \tag{28}$$

$$b = s_k^T H_k y_k \|s_k\| - \frac{\|s_k\|^3}{2} \tag{29}$$

$$c = -(s_k - H_k y_k)^T y_k. \tag{30}$$

Since H_k is positive definite, it follows that $a > 0$. For b, since $\nabla f(x)$ is Lipschitz continuous with L', using triangle inequality, we have

$$\begin{aligned} b &= s_k^T H_k y_k \|s_k\| - \frac{\|s_k\|^3}{2} \\ &\geq (L'\mu_H - \frac{1}{2})\|s_k\|^3 \end{aligned} \tag{31}$$

where $\mu_H > 0$. The inequality follows from the Assumption 2 and H_k is positive definite with $\|H_k\| \geq \mu_H$. Thus, we cannot make sure of the sign of b. For c, it should be noted that c is actually the negative denominator in (18). As the algorithm is designed to incorporate cubic regularized technique whenever the denominator in (18) is negative, c is strictly positive.Moreover, provided the discriminant is larger than zero, i.e., $b^2 - 4ac \geq 0$, the roots of the quadratic function is $M_k = \frac{-b \pm \sqrt{b^2-4ac}}{2a}$. According to the discussion above, there are three cases for the root of quadratic function $aM_k^2 + bM_k + c$ depending on both the coefficients and discriminant $b^2 - 4ac$, which are summarized as follows:

Case I: if $b^2 - 4ac < 0$, there are no real roots for the quadratic function and the inequality $aM_k^2 + bM_k + c < 0$ cannot be satisfied. It indicates that there is no M_k to guarantee the matrix updated via (25) or (26) is positive definite. Whenever case I happens, we suggest to skip the Hessian approximation update or adopt rank-two quasi-Newton update. It should be noted that $b = 0$ belongs to this case and thus we do not need to consider $b = 0$ for the other two cases.

Case II: if $b^2 - 4ac > 0$ and $b < 0$. Obviously $-b + \sqrt{b^2 - 4ac} > 0$, moreover, we have $-b - \sqrt{b^2 - 4ac} > -b - |b| = 0$. Hence, the roots of the quadratic function are both positive. Theoretically we can choose one value between the two roots to ensure positive definiteness of the updated matrix. However, numerical tests have shown that setting M_k with a value between the smaller root and $\frac{-b}{2a}$ works better. This matches ARC method which allows for a broader step once the algorithm works sufficient well. Intuitively for ARC, smaller cubic regularized parameter indicates broader step for the next iteration.

Case III: if $b^2 - 4ac > 0$ and $b > 0$. Obviously the smaller root is negative. Moreover, we have $-b + \sqrt{b^2 - 4ac} < -b + b = 0$. Thus, we obtain two negative roots for this case. Since we cannot choose a positive value to satisfy the inequality $aM_k^2 + bM_k + c < 0$, we deal this case with the same strategy in Case I.

According to the above discussions, the CuREG-SR1 based algorithm is summarized in *Algorithm 1*. Note from step 3 to step 8 in *Algorithm 1*, we apply backtracking line search scheme to dispense with the extra condition (7) and use just the Armijo condition for sufficient reduction to terminate the line search procedure [23]. In the step 11-17, as the generated Hessian approximation matrix may have large eigenvalue and thus ill-conditioned when the denominator is small, a simple and effective way to control the Hessian approximation matrix is adopted [3] [5], namely: skip the Hessian approximation matrix update when the denominator is small.

III. CONVERGENCE RESULTS

The idea of incorporating cubic regularized technique into SR1 quasi-Newton methods has been proposed by [2]. However, limited convergence results have been studied in [2]. Hence, in this section, we aim to further study the convergence results of CuREG-SR1. To be specific, by applying the technique of CuREG-SR1, we can obtain a positive definite update for the Hessian approximation matrix. Therefore, we assume that there is a positive constant $m, M > 0$ such that $mI \preceq H_k \preceq MI$. Following Theorem 3.2 in [23], we have the result that the algorithm converges to a first-order critical point:

Theorem 1: Consider Algorithm 1 with Assumption 2, we assume that the generated H_k is positive definite and its maximum eigenvalue is bounded above by $M > 0$, i.e., $mI \preceq H_k \preceq MI$. We further assume that the stepsize λ_k is

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2019.2935900, IEEE Access

Algorithm 1 CuREG-SR1

Input: initial optimization variable x_0 randomly generated from the uniform distribution $[-1,1]^d$, the initial inverse Hessian approximation matrix $H_0 = I$, $c_1 \in (0,1)$ and $c_2 \in (c_1,1)$, $\epsilon = 10^{-8}$ and $\alpha \in (0,1)$, the desired iteration number K

Output: x_K

for $k = 0, 1, \ldots, K$ **do**
  Set the search direction $p_k = -H_k \nabla f(x_k)$
  Set stepsize $\lambda_k = 1$
  if $f(x_k + \lambda p_k) < f(x_k) + c_1 \lambda \nabla f(x_k)^T p_k$, **then**
    Goes to the step 9.
  else
    Set stepsize $\lambda_k \leftarrow \alpha \lambda_k$ and goes back to step 4.
  end if
  Update the iterate $x_{k+1} = x_k - \lambda_k H_k \nabla f(x_k)$
  Set $H_{k+1} = H_k$,
  if $|(y_k - B_k s_k)^T s_k| > \epsilon \|y_k - B_k s_k\| \|s_k\|$ **then**
    if $(s_k - H_k y_k)^T y_k > 0$ **then**
      Calculate H_{k+1} via (18),
    else
      Compute a,b and c according (28)(29) and (30) respectively,
      if $b^2 - 4ac > 0$ and $b < 0$ **then**
        Set $M_k = -\frac{b}{2a}$ and calculate H_{k+1} according to (26),
      end if
    end if
  end if
end for

chosen to satisfy the Wolfe condition (6) and (7). Moreover, the objective function $f(x)$ is bounded below in $\mathbb{R}^d$, then it follows:

$$\lim_{k\to\infty} \|\nabla f(x_k)\| = 0. \tag{32}$$

Proof: From the Armijo condition that $f(x_{k+1}) < f(x_k) + c_1 \lambda_k \nabla f(x_k)^T p_k$, take summation of both sides from 0 to k, we have

$$f(x_{k+1}) < f(x_0) + c_1 \sum_{i=0}^{k} \lambda_k \nabla f(x_k)^T p_k. \tag{33}$$

Observe from (33), since we have assumed that the function f is bounded below and Algorithm 1 generates descent direction at each iteration, we have $-\sum_{i=0}^{k} \lambda_k \nabla f(x_k)^T p_k$ is bounded above by a positive scalar. Next, we derive the result that the stepsize is bounded below. Since the gradient of f is Lipschitz continuous in $\mathbb{R}^d$, we have $\|\nabla f(x_{k+1}) - \nabla f(x_k)\| \le L' \|x_{k+1} - x_k\| = \lambda_k L' \|p_k\|$, which indicates:

$$[\nabla f(x_{k+1}) - \nabla f(x_k)]^T p_k \le \lambda_k L' \|p_k\|^2. \tag{34}$$

On the other hand, from the curvature condition that $\nabla f(x_{k+1})^T p_k \ge c_2 \nabla f(x_k)^T p_k$, we have the LHS of (34) is bounded below:

$$[\nabla f(x_{k+1}) - \nabla f(x_k)]^T p_k \ge (c_2 - 1) \nabla f(x_k)^T p_k \tag{35}$$

Hence, observe from (34) and (35), it follows that the stepsize is bounded below:

$$\lambda_k \ge \frac{(c_2 - 1) \nabla f(x_k)^T p_k}{L' \|p_k\|^2}. \tag{36}$$

Substituting the above result into (33), we have

$$\sum_{i=0}^{k} \frac{[\nabla f(x_k)^T p_k]^2}{\|p_k\|^2} < \gamma, \tag{37}$$

where $\gamma = \frac{f(x_0) - f_{\min}}{L'(1-c_2)} > 0$ with $f_{\min}$ is the minimum value of the sequence $\{f(x_k)\}_{k=0}^{\infty}$. By applying the triangle inequality to the denominator, it follows that the summand in (37) satisfies:

$$\frac{[\nabla f(x_k)^T p_k]^2}{\|p_k\|^2} = \frac{[\nabla f(x_k)^T H_k \nabla f(x_k)]^2}{\nabla f(x_k)^T H_k \nabla f(x_k)} \ge \frac{m^2}{M^2} \|\nabla f(x_k)\|^2 \tag{38}$$

The second inequality in (38) follows from $mI \preceq H_k \preceq MI$. Hence, combining (37) and (38) and taking the limits, we get:

$$\sum_{k=0}^{\infty} \|\nabla f(x_k)\|^2 < \frac{\gamma M^2}{m^2} < \infty. \tag{39}$$

The result in (39) implies that $\lim_{k\to\infty} \|\nabla f(x_k)\| = 0$ holds.

We have shown that Algorithm 1 has converged to a first-order critical point. Next, out aim is to obtain the convergence rate of the algorithm. Note from (27), when Case II happens, we have

$$[s_k - H_k(y_k + \frac{M_k}{2} \|s_k\| s_k)]^T (y_k + \frac{M_k}{2} \|s_k\| s_k) > \epsilon' \tag{40}$$

for some positive constant $\epsilon' > 0$.

Thus, similar to SR1 that $|s_k^T (y_k - B_k s_k)| \ge \epsilon \|s_k\| \|y_k - B_k s_k\|$, we update B_k whenever the following holds:

$$\begin{aligned} v|[y_k + (\frac{M_k}{2} \|s_k\| I - B_k) s_k]^T s_k| \ge \\ \epsilon \left\| [y_k + (\frac{M_k}{2} \|s_k\| I - B_k) s_k \right\| \|s_k\| \end{aligned} \tag{41}$$

where ϵ is normally chosen to be 10^{-8}. Now we proceed to introduce the following assumption, based on which we derive Lemma 1.

Assumption 5: At the critical point x^*, the second-order derivative is positive definite $\nabla^2 f(x^*) \succeq \delta I$, for some positive constant $\delta > 0$.

Lemma 1: Suppose that *Assumptions 1-3* and *5* hold. The Hessian approximation matrix is assumed to be updated through CuREG-SR1 at each iteration, and M_k is obtained via Case II, i.e., $b^2 - 4ac > 0$ and $b < 0$, which happens with a, b and c calculated in (28), (29) and (30) respectively. Moreover, (41) holds at each iteration, the generated

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2019.2935900, IEEE Access

sequence $\{x_k\}$ is sufficient close to the critical point x^* and the sequence $\{H_k\}$ satisfies $mI \preceq H_k \preceq MI$. Then the following inequality holds:

$$\|y_j - B_i s_j\| \leq \left(\frac{\zeta + 2L''\eta_{i,j}}{2\epsilon}\right)\left(\frac{2}{\epsilon}+1\right)^{i-j-2}\|s_j\|, \tag{42}$$

for $j \leq i-1$, where ζ is a constant defined by $\zeta := \frac{1-2m\delta}{m}$ and $\eta_{i,j}$ is defined by

$$\eta_{i,j} := \max\{\|x_s - x_l\| : j \leq s \leq l \leq i\}. \tag{43}$$

Proof: we proof (42) by induction. Suppose (42) holds for $i = j+1, \ldots, k$ and $k \geq j+1$. By using the inductive assumption, we have for $i = k+1$,

$$\begin{aligned}\|y_j - B_{k+1}s_j\| &\leq \|y_j - B_k s_j\| + \\ &\left\|\frac{[y_k + (\frac{M_k}{2}\|s_k\| I - B_k)s_k][y_k + (\frac{M_k}{2}\|s_k\| I - B_k)s_k]^T s_j}{[y_k + (\frac{M_k}{2}\|s_k\| I - B_k)s_k]^T s_k}\right\| \\ &\leq \|y_j - B_k s_j\| + \frac{|[y_k + \frac{M_k}{2}\|s_k\| s_k - B_k s_k]^T s_j|}{\epsilon\|s_k\|}\end{aligned} \tag{44}$$

where the first inequality follows from the triangle inequality the second follows from the Cauchy-Schwarz inequality and (41). Now, let us consider the second term $\frac{|[y_k+\frac{M_k}{2}\|s_k\|s_k - B_k s_k]^T s_j|}{\epsilon\|s_k\|}$, it follows that

$$\begin{aligned}\frac{|[y_k + \frac{M_k}{2}\|s_k\| s_k - B_k s_k]^T s_j|}{\epsilon\|s_k\|} &\leq \frac{|y_k^T s_j - s_k^T B_k s_j|}{\epsilon\|s_k\|} \\ &+ \frac{\frac{M_k}{2}\|s_k\|\|s_j\|}{\epsilon}\end{aligned} \tag{45}$$

By adding and abstract $s_k^T y_j$ to the nominator of the first term on the LHS, it leads to

$$\begin{aligned}|y_k^T s_j - s_k^T B_k s_j| &\leq |y_k^T s_j - s_k^T y_j| + \|s_k\|\|y_j - B_k s_j\| \\ &\leq |y_k^T s_j - s_k^T y_j| + \left(\frac{\zeta + 2L''\eta_{k,j}}{2\epsilon}\right)\left(\frac{2}{\epsilon}+1\right)^{k-j-2}\|s_j\|\|s_k\|\end{aligned} \tag{46}$$

where for first inequality, we have used the triangle inequality and the Cauchy-Schwarz inequality, and the result in (42) for the second inequality. Note $y_k = \nabla f(x_{k+1}) - f(x_k)$, by using mean value theorem, we can obtain:

$$y_k = \nabla^2 f(x_k + t_k s_k)s_k, \text{ for } t_k \in (0,1). \tag{47}$$

Similarly we can obtain $y_j = \nabla^2 f(x_j + t_j s_j)s_j$ for $t_j \in (0,1)$. Substituting the above results into $|y_k^T s_j - s_k^T y_j|$, we have

$$\begin{aligned}|y_k^T s_j - s_k^T y_j| &= |s_k^T(\nabla^2 f(x_k + t_k s_k) - \nabla^2 f(x_j + t_j s_j))s_j| \\ &\leq \|s_k\|\left\|\nabla^2 f(x_k + t_k s_k) - \nabla^2 f(x_j + t_j s_j)\right\|\|s_j\| \\ &\leq L''\eta_{k+1,j}\|s_k\|\|s_j\|,\end{aligned} \tag{48}$$

where we have used the assumption that the Hessian of f is Lipschitz continuous. Now, let us consider $M_k\|s_k\|$ in the RHS of (45). Recall that M_k can have various values between the two roots in Case II. However, for simplicity, we assume M_k is chosen to be $M_k = \frac{-b}{2a}$. Hence, it follows:

$$\begin{aligned}M_k\|s_k\| = \frac{-b}{2a}\|s_k\| &= \frac{\|s_k\|^2 - 2s_k^T H_k y_k}{s_k^T H_k s_k} \\ &= \frac{\|s_k\|^2 - 2s_k^T H_k \nabla^2 f(x_k + t_k s_k)s_k}{s_k^T H_k s_k} \\ &= \frac{\|s_k\|^2 - 2s_k^T H_k \nabla^2 f(x_k + t_k s_k)s_k}{s_k^T H_k s_k}\end{aligned} \tag{49}$$

Since $\{x_k\}$ is sufficiently close to x^*, using the continuity of f, we have $\nabla^2 f(x_k + t_k s_k) \succeq \delta I$. Moreover, combining $H_k \succeq mI$ leads to

$$M_k\|s_k\| \leq \frac{1-2m\delta}{m} = \zeta. \tag{50}$$

Combining (45)-(50) and substituting the results into (44), we obtain

$$\begin{aligned}&\|y_j - B_{k+1}s_j\| \\ &\leq \left(\frac{1}{\epsilon}+1\right)\|y_j - B_k s_j\| + \left(\frac{\zeta + 2L''\eta_{k+1,j}}{2\epsilon}\right)\|s_j\| \\ &\leq \left(\frac{1}{\epsilon}+1\right)\left(\frac{\zeta + 2L''\eta_{k+1,j}}{2\epsilon}\right)\left(\frac{2}{\epsilon}+1\right)^{k-j-2}\|s_j\| \\ &\quad + \left(\frac{\zeta + 2L''\eta_{k+1,j}}{2\epsilon}\right)\|s_j\| := \mathcal{L}_{k+1}\end{aligned} \tag{51}$$

where we have used the simple inequality $\eta_{k+1,j} \geq \eta_{k,j}$. Now, our work left is to compare the last inequality in (51) with $\mathcal{B}_{k+1} := \left(\frac{\zeta+2L''\eta_{k+1,j}}{2\epsilon}\right)\left(\frac{2}{\epsilon}+1\right)^{k-j-1}\|s_j\|$. By straightforwardly calculating $\mathcal{B}_{k+1} - \mathcal{L}_{k+1}$, we have

$$\begin{aligned}\mathcal{B}_{k+1} - \mathcal{L}_{k+1} = &\left[\frac{1}{\epsilon}\left(\frac{2}{\epsilon}+1\right)^{k-j-2} - 1\right] \\ &\cdot \left(\frac{\zeta + 2L''\eta_{k+1,j}}{2\epsilon}\right).\end{aligned} \tag{52}$$

Hence, $\mathcal{B}_{k+1} > \mathcal{L}_{k+1}$ holds by taking into account $\epsilon \in (0,1)$, which gives (42) for $i = k+1$.

As mentioned above, the sequence $\{B_k\}$ generated by classical SR1 formula in (17) converges to the true Hessian, provided *Assumption 4* holds true. However, for CuREG-SR1 formulas (25) and (26), the sequence $\{B_k\}$ generated by CuREG-SR1 formula (25) no longer converges to the true Hessian since we have incorporated cubic regularized technique with the parameter M_k. Nonetheless, we can still show that their difference is bounded above based on *Lemma 1* and *Assumption 4* in the next theorem.

Theorem 2. Assume the conditions in *Lemma 1* and *Assumption 4* hold. $\{x_k\}$ is the generated sequence. Then, there exit positive constants $c_1, c_2 > 0$, an integer k_0 and $m \geq d$ such that for any $k \geq k_0$,

$$\left\|B_{k+m+1} - \nabla^2 f(x^*)\right\| \leq c_1 + c_2\varepsilon_k \tag{53}$$

where $\varepsilon_k = \max\{\|x_s - x^*\| \,|k \leq s \leq k+m+1\}$.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2019.2935900, IEEE Access

Proof: We have derived the bounded result for $\|y_j - B_k s_j\|$ in (42). Observe from (53), it implies that our next step is to derive the bounded result for $\|y_j - \nabla^s f(x^*)s_j\|$. Then, we can further use the triangle inequality and *Assumption 4* to show that the sequence $\{s_k\}$ is uniformly independent. To start with, we apply the mean value theorem for $k \leq j \leq k+m$, and it yields:

$$\begin{aligned}\left\|y_j - \nabla^2 f(x^*)s_j\right\| &= \left\|(\nabla^2 f(x_j + t_j s_j) - \nabla^2 f(x^*))s_j\right\| \\ &\leq \left\|\nabla^2 f(x_j + t_j s_j) - \nabla^2 f(x^*)\right\| \|s_j\| \\ &\leq L''\varepsilon_k \|s_j\|.\end{aligned} \tag{54}$$

Now, we consider $\|y_j - B_{k+m+1}s_j\|$. By using *Lemma 1*, we have:

$$\begin{aligned}\|y_j - B_{k+m+1}s_j\| &\leq \left(\frac{\zeta + 2L''\eta_{k+m+1,j}}{2\epsilon}\right) \\ &\quad \cdot \left(\frac{2}{\epsilon}+1\right)^{k+m-j-1} \|s_j\| \\ &\leq \left(\frac{\zeta + 4L''\varepsilon_k}{2\epsilon}\right)\left(\frac{2}{\epsilon}+1\right)^{m-1}\|s_j\|,\end{aligned} \tag{55}$$

where we have used the triangle inequality

$$\begin{aligned}\eta_{k+m+1,j} \leq \eta_{k+m+1,k} &= \max\{\|x_s - x_l\| : j \leq s \leq\ l \leq i\} \\ &\leq \max\{\|x_s - x^*\| + \|x_l - x^*\| : j \leq s \leq\ l \leq i\} \\ &\leq 2\varepsilon_k.\end{aligned} \tag{56}$$

Hence, it follows by combining (54) and (55) that

$$\begin{aligned}&\left\|[B_{k+m+1} - \nabla^2 f(x^*)]s_j\right\| \\ &\leq \left\|y_j - \nabla^2 f(x^*)s_j\right\| + \|y_j - B_{k+m+1}s_j\| \\ &= \left\{\left(\frac{\zeta + 4L''\varepsilon_k}{2\epsilon}\right)\left(\frac{2}{\epsilon}+1\right)^{m-1} + L''\varepsilon_k\right\}\|s_j\|.\end{aligned} \tag{57}$$

Subsequently, for $k \leq j \leq k+m$, according to *Assumption 4*, one can choose d distinct indices between k and $k+m$ for the sequence $\{s_k\}$ to formulate the matrix S_k defined in (10). Thus, for any $j \in [k, k+m]$, it yields:

$$\left\|[B_{k+m+1} - \nabla^2 f(x^*)]S_k\right\| \leq \sqrt{d}\left\|[B_{k+m+1} - \nabla^2 f(x^*)]s_j\right\|. \tag{58}$$

Moreover, because of *Assumption 4*, the minimum eigenvalue of the matrix S_k is positive, i.e., $\sigma_{\min}(S_k) > 0$, it indicates:

$$\left\|[B_{k+m+1} - \nabla^2 f(x^*)]S_k\right\| \geq \tau\left\|B_{k+m+1} - \nabla^2 f(x^*)\right\|, \tag{59}$$

for some $\tau > 0$ satisfying $\sigma_{\min}(S_k) \geq \tau$. Combining (57)-(59), we obtain the desired result in (53) with c_1 and c_2 given respectively as follows:

$$c_1 = \frac{\sqrt{d}\zeta}{2\epsilon\tau}\left(\frac{2}{\epsilon}+1\right)^{m-1}, \tag{60}$$

$$c_2 = \frac{\sqrt{d}}{\tau}\left[\frac{2L''}{\epsilon}\left(\frac{2}{\epsilon}+1\right)^{m-1} + L''\right] \tag{61}$$

The result in *Theorem 2* shows that the difference between the approximate Hessian matrix updated via (25) and the true Hessian is bounded above. Moreover, as the iteration sequence converges to a critical point by *Theorem 1*, we have

$$\lim_{k\to\infty}\left\|B_{k+m+1} - \nabla^2 f(x^*)\right\| \leq c_1. \tag{62}$$

However, for the SR1 update in (17) with the similar conditions, it will converge to the true Hessian [5]. The difference is due to the use of cubic regularization parameter M_k, which leads to the shift $c_1 > 0$. While the condition of $\{B_k\}$ converging to the true Hessian indicates superlinear convergence result for a class of quasi-Newton methods [4], [6], conversely for an algorithm to reach superlinear convergence rate, it does not require $\{B_k\}$ converging to $\nabla^2 f(x^*)$. For *Algorithm 1*, before we derive the convergence rate, we need the following lemmas, in which we have adopted the proof technique of *Lemma 3.2* in [3] with different settings, based on which, we further derive a similar result for the CuREG-SR1 algorithm. In addition, we mention that it does not require the uniform independent assumption.

Lemma 2. Suppose the sequence $\{x_k\}$ converges to a first-order critical point x^*. For any set of $d+1$ steps $\psi := \{s_{k_i} : k_0 \leq k_1 \leq \cdots \leq k_{d+1}\}$ with some integer k_0, and $i = 1, \cdots, d+1$, consider the matrix $S_i \in \mathbb{R}^{d\times i}$ defined by

$$S_i = [\frac{s_{k_1}}{\|s_{k_1}\|}, \cdots, \frac{s_{k_i}}{\|s_{k_i}\|}]. \tag{63}$$

There exits an index k_p for $p \in \{2, \cdots, d+1\}$ such that $\frac{s_{k_p}}{\|s_{k_p}\|} = S_{p-1}u + \nu$, with some vectors $u \in \mathbb{R}^{p-1}$ and $\mu \in \mathbb{R}^d$ satisfying:

$$\|u\| < \frac{2}{\varepsilon_\psi^{(d-1)\kappa}}, \quad \|\nu\| < 2\varepsilon_\psi^\kappa, \tag{64}$$

where ε_ψ is defined as $\varepsilon_\psi := \max\{\|x_{k_i} - x^*\| : i = 1, \cdots, d+1\}$.

Proof: For $i = 1, \cdots, d$, let us denote σ_i as the smallest singular value of the matrix S_i and $\sigma_{d+1} := 0$. Subsequently, it follows that [3]:

$$1 \geq \sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_{d+1} = 0. \tag{65}$$

Choose $p \in \{2, \cdots, d+1\}$ to be the smallest integer such that $\frac{\sigma_p}{\sigma_{p-1}} < \varepsilon_\psi^\kappa$ with $\kappa > 0$. Since $\sigma_{p-1} < 1$, it follows that $\sigma_p < \varepsilon_\psi^\kappa$. Moreover, since $\sigma_{p-1} = \sigma_1\left(\frac{\sigma_2}{\sigma_1}\right)\cdots\left(\frac{\sigma_{p-1}}{\sigma_{p-2}}\right)$, with (65) and $p \leq d+1$, we have:

$$\sigma_{p-1} > \varepsilon_\psi^{(p-2)\kappa} > \varepsilon_\psi^{(d-1)\kappa}. \tag{66}$$

We choose a vector $z = [u^T, -1]^T$ with $u \in \mathbb{R}^{p-1}$ such that

$$\|S_p z\| = \sigma_p \|z\|. \tag{67}$$

Since $S_p = [S_{p-1}, \frac{s_{k_p}}{\|s_{k_p}\|}]$, by defining $\nu := S_p z$, it leads to $\frac{s_{k_p}}{\|s_{k_p}\|} = S_{p-1}u - \nu$. Subsequently, we have

$$\|u\| \leq \frac{1}{\sigma_{p-1}}\|S_{p-1}u\| \leq \frac{\|\nu\|+1}{\varepsilon_\psi^{(d-1)\kappa}} \tag{68}$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2019.2935900, IEEE Access

where the first inequality follows that σ_{p-1} is the smallest singular value of S_{p-1}, and the second follows from triangle inequality and (66). Now, we continue to derive the bound on $\|\nu\|$. Combining (67) and (68), we have

$$\|\nu\|^2 = \|S_p z\|^2 = \sigma_p^2 \|z\|^2 = \sigma_p^2(1 + \|u\|^2). \tag{69}$$

Together with $\sigma_p < \varepsilon_\psi^\kappa$ and using (68), it subsequently leads to

$$\begin{aligned} \|\nu\|^2 &< \varepsilon_\psi^{2\kappa} + \varepsilon_\psi^{2\kappa}(\|\nu\| + 1)^2 \\ &< 4\varepsilon_\psi^{2\kappa}(\|\nu\| + 1)^2. \end{aligned} \tag{70}$$

Furthermore, since $x_k \to x^*$, without loss of generality, we assume $\varepsilon_\psi < (\frac{1}{2})^{1/\kappa}$. Subsequently, it leads to $\|\nu\| < 2\varepsilon_\psi^\kappa$. Substituting the above results into (68), we can obtain $\|u\| < \frac{2}{\varepsilon_\psi^{(d-1)\kappa}}$.

Lemma 2 shows that $\|\nu\|$ is small, which implies that for sufficiently large k, s_{k_p} is nearly in the space spanned by S_{p-1}. In particular, when p happens to be $d+1$, s_{k_p} is exactly in the space spanned by S_{p-1}, provided that S_{p-1} has full rank and is well conditioned. With the above result, we continue to derive the following lemma. The basic idea is to show that the quantity $\left\|(B_{k_p} - \nabla^2 f(x^*))s_{k_p}\right\| / \left\|s_{k_p}\right\|$ is small for some k_p. It should be noted that the quantity has the same form with the well-known Dennis-Moré condition [4], [6], which is closely related to superlinear convergence. However, we observe from (57) that due to the existence of the shift c_1 caused by the cubic regularization parameter, such quantity is bounded above. Hence, we need reasonable condition to restrict M_k.

Lemma 3. Suppose the conditions in *Lemma 1* are satisfied for the sequence $\{x_k\}$ and $\{B_k\}$ generated by *Algorithm 1*. Assume further that there is a positive constant $c_M > 0$ such that $M_k < c_M$. Moreover, the sequence $\{x_k\}$ converges to a first-order critical point x^*. Then there exists an integer k_0 and an index k_p such that it satisfies:

$$\frac{\left\|[B_{k_p} - \nabla^2 f(x^*)]s_{k_p}\right\|}{\left\|s_{k_p}\right\|} < 2\sqrt{d}c_3\varepsilon_\psi^{1-(d-1)\kappa} + c_4\varepsilon_\psi^\kappa, \tag{71}$$

for any set of $d+1$ steps $\psi := \{s_{k_i} : k_0 \le k_1 \le \cdots \le k_{d+1}\}$ and $p \in \{2, \cdots, d+1\}$, where $c_4 = 2(m^{-1} + \left\|\nabla^2 f(x^*)\right\|)$. In particular, set $\kappa = 1/d$, the following inequality holds:

$$\frac{\left\|[B_{k_p} - \nabla^2 f(x^*)]s_{k_p}\right\|}{\left\|s_{k_p}\right\|} < c_5\varepsilon_\psi^{1/d}, \tag{72}$$

where $c_5 = 2(\sqrt{d}c_3 + m^{-1} + \left\|\nabla^2 f(x^*)\right\|)$.

Proof: We first show that the proof technique in *Lemma 1* can be applied to obtain the following result:

$$\|y_j - B_i s_j\| \le \frac{(c_M + 2L'')\eta_{i,j}}{2\epsilon}\left(\frac{2}{\epsilon} + 1\right)^{i-j-2}\|s_j\|, \tag{73}$$

with $i \ge j+1$. Suppose it is satisfied for $i = k$. Now, let us consider $i = k+1$. Note from (45) and $s_k \le \eta_{k,j}$ that

$$\begin{aligned} \frac{|[y_k + \frac{M_k}{2}\|s_k\| s_k - B_k s_k]^T s_j|}{\epsilon\|s_k\|} &\le \frac{|y_k^T s_j - s_k^T B_k s_j|}{\epsilon\|s_k\|} + \frac{c_M\eta_{k,j}\|s_j\|}{2\epsilon} \\ &\le \frac{1}{\epsilon}\|y_j - B_k s_j\| + \frac{(c_M + 2L'')\eta_{k+1,j}}{2\epsilon}\|s_j\|. \end{aligned} \tag{74}$$

Hence, it follows:

$$\begin{aligned} &\|y_j - B_{k+1}s_j\| \\ &\quad\le \left(1 + \frac{1}{\epsilon}\right)\|y_j - B_k s_j\| + \frac{(c_M + 2L'')\eta_{k+1,j}}{2\epsilon}\|s_j\|. \end{aligned} \tag{75}$$

Following the technique in *Lemma 1*, we can obtain the desired result in (73). Now, we focus on (71). First, we note that

$$\begin{aligned} \eta_{k_i,j} &= \max\{\|x_s - x_l\| : s, l \in [j, k_i]\} \\ &\le \max\{\|x_s - x^*\| + \|x_l - x^*\| : s, l \in [j, k_i]\} \\ &\le 2 \cdot \max\{\|x_{k_l} - x^*\| : l = 0, \cdots, d+1\}. \end{aligned} \tag{76}$$

Hence, by substituting (76) into (73), we get

$$\begin{aligned} \|y_j - B_{k_i}s_j\| &\le \frac{(c_M + 2L'')\eta_{k_i,j}}{2\epsilon}\left(\frac{2}{\epsilon} + 1\right)^{k_i-j-2}\|s_j\| \\ &\le \frac{(c_M + 2L'')\varepsilon_\psi}{\epsilon}\left(\frac{2}{\epsilon} + 1\right)^{k_{d+1}-k_1-2}\|s_j\|, \end{aligned} \tag{77}$$

where we set i in (73) to k_i, and the subscript i can take values in $\{2, \cdots, d+1\}$. In addition, by using (54) and (76), we can obtain:

$$\left\|y_j - \nabla^2 f(x^*)s_j\right\| \le L''\varepsilon_\psi\|s_j\|, \tag{78}$$

where the mean value theorem and the fact that the Hessian of f is Lipschitz continuous are used. Subsequently, the triangle inequality is further applied to obtain

$$\begin{aligned} \left\|(B_{k_i} - \nabla^2 f(x^*))\frac{s_j}{\|s_j\|}\right\| &\le \left\|y_j - B_{k_i}\frac{s_j}{\|s_j\|}\right\| \\ &\quad + \left\|y_j - \nabla^2 f(x^*)\frac{s_j}{\|s_j\|}\right\| \\ &\le c_3\varepsilon_\psi, \end{aligned} \tag{79}$$

where the constants c_3 is given by:

$$c_3 = \frac{c_M + 2L''}{\epsilon}\left(\frac{2}{\epsilon} + 1\right)^{k_{d+1}-k_1-2} + L''. \tag{80}$$

We proceed to complete the proof by using the results in *Lemma 2*. By setting $i = p$, it follows from (62) and (79) that

$$\left\|(B_{k_p} - \nabla^2 f(x^*))S_{p-1}\right\| \le \sqrt{d}c_3\varepsilon_\psi. \tag{81}$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2019.2935900, IEEE Access

Combining (81) and *Lemma 2* that $\frac{s_{k_p}}{\|s_{k_p}\|} = S_{p-1}u + \nu$, we have

$$\begin{aligned}\frac{\left\|(B_{k_p} - \nabla^2 f(x^*))s_{k_p}\right\|}{\|s_{k_p}\|} &= \left\|(B_{k_p} - \nabla^2 f(x^*))(S_{p-1}u + \nu)\right\| \\ &\leq \left\|(B_{k_p} - \nabla^2 f(x^*))S_{p-1}\right\| \|u\| + (\|B_{k_p}\| + \left\|\nabla^2 f(x^*)\right\|) \|\nu\| \\ &< \sqrt{d}c_3\varepsilon_\psi \frac{2}{\varepsilon_\psi^{(d-1)\kappa}} + 2(m^{-1} + \left\|\nabla^2 f(x^*)\right\|)\varepsilon_\psi^\kappa \\ &= 2\sqrt{d}c_3\varepsilon_\psi^{1-(d-1)\kappa} + 2(m^{-1} + \left\|\nabla^2 f(x^*)\right\|)\varepsilon_\psi^\kappa,\end{aligned} \tag{82}$$

where we obtain the desired result in (71). Furthermore, by setting $\kappa = 1/d$, it subsequently leads to (72).

Note that the quantity $\left\|(B_{k_p} - \nabla^2 f(x^*))s_{k_p}\right\| / \|s_{k_p}\|$ is not necessarily small if we choose $\kappa > 1/(d-1)$. Therefore, to derive the convergence rate, we set $\kappa > 1/(d-1)$. Moreover, provided the generated B_k at each step is positive definite, we show that for every q iterations, *Algorithm 1* generates at least $q - d$ superlinear steps.

Theorem 3. Suppose *Assumptions 1-3,5* are satisfied and the condition (41) holds at each iteration. Set $\kappa > 1/(d-1)$. In addition, the conditions in *Lemma 1* are required. Then, there exits an integer k_0 such that for each $k > k_0$ and $q \geq d+1$, the following inequality holds:

$$\|x_{k+q} - x^*\| \leq \alpha_k \|x_k - x^*\|, \tag{83}$$

where the sequence $\{\alpha_k\}$ satisfies $\lim_{k\to\infty} \alpha_k = 0$.

Proof: For convenience, let us denote $e_k := \|x_k - x^*\|$. Note that *Assumption 2* is equivalent to $\nabla^2 f(x^*) \preceq L'I$. Combining $\nabla^2 f(x^*) \succeq \delta I$ and $f(x_k) - f(x^*) = \frac{1}{2}(x_k - x^*)^T \nabla^2 f(\tilde{x})(x_k - x^*)$, where $\tilde{x} = x^* + t(x_k - x^*)$, it follows that

$$\frac{1}{2}\delta e_k^2 \leq f(x_k) - f(x^*) \leq \frac{1}{2}L' e_k^2. \tag{84}$$

Hence, since the algorithm results in descent step at each iteration, we have for $l > k > k_0$ that

$$e_l \leq \sqrt{\frac{L'}{\delta}} e_k. \tag{85}$$

Consider the set of $d+1$ steps $\{s_i : i = k, \cdots, k+d\}$, by applying *Lemma 3*, there exits $l_1 \in \{k+1, \cdots, k+d\}$ such that

$$\begin{aligned}\frac{\left\|(B_{l_1} - \nabla^2 f(x^*))s_{l_1}\right\|}{\|s_{l_1}\|} &< 2\sqrt{d}c_3\varepsilon_\psi^{1-(d-1)\kappa} + c_4\varepsilon_\psi^\kappa \\ &< 2\sqrt{\frac{L'd}{\delta}}c_3 e_k^{1-(d-1)\kappa} + \sqrt{\frac{L'}{\delta}}c_4 e_k^\kappa\end{aligned} \tag{86}$$

where we have used (85) that $\varepsilon_\psi \leq \sqrt{\frac{L'}{\delta}} e_k$. Additionally, for sufficient large $k > k_0$, we have from (86) that there exits a positive constant c_6 such that:

$$\frac{\left\|(B_{l_1} - \nabla^2 f(x^*))s_{l_1}\right\|}{\|s_{l_1}\|} < c_6 e_k^{\tilde{\kappa}}, \tag{87}$$

where $\tilde{\kappa} = \max\{1 - (d-1)\kappa, \kappa\}$. Note that the inequality in Lemma 3.3 [3] can be applied in our case, and since $\tilde{\kappa} < 1$, we have

$$\begin{aligned}e_{l_1+1} &\leq \left\|\nabla^2 f(x^*)^{-1}\right\| \left[2\frac{\left\|(B_{l_1} - \nabla^2 f(x^*))s_{l_1}\right\|}{\|s_{l_1}\|} e_{l_1} + \frac{L''}{2}e_{l_1}^2\right] \\ &< 4\left\|\nabla^2 f(x^*)^{-1}\right\| c_6 e_k^{\tilde{\kappa}} e_{l_1}.\end{aligned} \tag{88}$$

The above results are based on the fact that for sufficiently large $k > k_0$, e_k are small. Next, we apply the same method to the set $\{s_k, \cdots, s_{k+d}, s_{k+d+1}\} - s_{l_1}$, then we can obtain l_2 such that

$$e_{l_2+1} < 4\left\|\nabla^2 f(x^*)^{-1}\right\| c_6 e_k^{\tilde{\kappa}} e_{l_2}. \tag{89}$$

Hence, by repeating the procedure, we obtain the following inequalities with respect to the indices $l_1 \leq l_2 \leq \cdots \leq l_{q-d}$

$$e_{l_i+1} < 4\left\|\nabla^2 f(x^*)^{-1}\right\| c_6 e_k^{\tilde{\kappa}} e_{l_i} \tag{90}$$

for $i \in \{1, \cdots, q-d\}$. Define $\Delta f_k := f(x_k) - f(x^*)$, note that each step results in a reduction in f, i.e., $\Delta f_{k+1} < \Delta f_k$. Moreover, from (84) we have:

$$\begin{aligned}\Delta f_{l_i+1} \leq \frac{1}{2}L' e_{l_i+1}^2 &< 8\left\|\nabla^2 f(x^*)^{-1}\right\|^2 L' c_6^2 e_k^{2\tilde{\kappa}} e_{l_i}^2 \\ &< \frac{8L'c_6^2}{\delta}\left\|\nabla^2 f(x^*)^{-1}\right\|^2 e_k^{2\tilde{\kappa}} \Delta f_{l_i}.\end{aligned} \tag{91}$$

Subsequently, it leads to

$$\Delta f_{k+q} < \left\{\frac{8L'c_6^2}{\delta}\left\|\nabla^2 f(x^*)^{-1}\right\|^2 e_k^{2\tilde{\kappa}}\right\}^{q-d} \Delta f_k. \tag{92}$$

By applying (84), it yields

$$e_{k+q} < \sqrt{\frac{L'}{\delta}}\left\{\sqrt{\frac{8L'}{\delta}}\left\|\nabla^2 f(x^*)^{-1}\right\| c_6 e_k^{\tilde{\kappa}}\right\}^{q-d} e_k. \tag{93}$$

For convenience, let us denote

$$c_7 = \sqrt{\frac{L'}{\delta}}\left\{\sqrt{\frac{8L'}{\delta}}\left\|\nabla^2 f(x^*)^{-1}\right\| c_6\right\}^{q-d}, \tag{94}$$

it leads to $e_{k+q} < c_7 e_k^{\tilde{\kappa}(q-d)} e_k$. Hence we obtain the desired result in (83). In addition, it indicates that the sequence $\{e_k\}$ exhibits q step superlinear convergence rate.

IV. INCREMENTAL CUREG-SR1

As mentioned above, massive data may result in expensive computation for traditional algorithms in the big data era. Therefore, direct application of the algorithms to problems with large samples is rather inefficient. In this section we propose a novel and efficient method based on CuREG-SR1 and SR1 to solve large scale problems. Specifically, our proposed algorithm aims to solve a class of problems which can be written as the sum of functions. This kind of problems widely exist in different areas such as source localization in sensor networks [27], [28], machine learning problems [29], [30], [32], [33], [35], computational biology [18], [19] and robotics [26].

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2019.2935900, IEEE Access

Now let us consider the problem mentioned above, which has the form:

$$x^* = \text{argmin}_{x\in\mathbb{R}^d}\ f(x) := \frac{1}{N}\sum_{i=1}^{N} f_i(x), \tag{95}$$

where N is the sample size and there are N individual functions f_i. Obviously, when naively applying traditional quasi-Newton method, we are required to evaluate the gradient N times and compute the large sample summation, which are computationally expensive. Stochastic scheme has been widely adopted to alleviate such complexity [20], [38]–[40] by sampling a mini batch of the large samples to estimate the gradient, which is used in lieu of the exact gradient [36]. This idea has substantially reduced the computational cost and extended to stochastic quasi-Newton method based on BFGS [13], [15]. In this section, we propose an efficient algorithm based on SR1 and CuREG-SR1 by incorporating the incremental method.

Specifically, we approximate each individual function f_i by second-order Taylor expansion around its current iterate z_i^k. Subsequently, we obtain an approximation to the function f:

$$\begin{aligned} f(x) \approx \frac{1}{N}\sum_{i=1}^{N}\{f_i(z_i^k) + \nabla f(z_i^k)^T(x - z_i^k) \\ + \frac{1}{2}(x - z_i^k)^T B_i^k (x - z_i^k)\}, \end{aligned} \tag{96}$$

where the matrix B_i^k is the local Hessian approximation to $\nabla^2 f_i(z_i^k)$. Furthermore, we refer to $\{z_i^k, \nabla f(z_i^k), B_i^k\}$ as the information corresponds to the individual function f_i. Using the same strategy with quasi-Newton methods to obtain the step direction for next iteration, we minimize the RHS in (96). It subsequently yields:

$$x_{k+1} = (\tilde{B}^k)^{-1}(\tilde{z}^k - \tilde{g}^k) \tag{97}$$

where $\tilde{B}^k := \sum_{i=1}^N B_i^k$, $\tilde{z}^k = \sum_{i=1}^N B_i^k z_i^k$ and $\tilde{g}^k := \sum_{i=1}^N \nabla f(z_i^k)$. In this paper, we consider update the Hessian approximation matrix by using SR1 and CuREG-SR1.

It can be seen from (97) that it involves matrix inversion and summation arsing from large samples, which is computationally expensive. To simplify computation, we only update the information associated with one chosen individual function at each iteration, while the information corresponding to other individual functions is left intact. The function is selected by cyclically iterating through N individual functions. Without loss of generality, we start to select the first individual function f_1, then at iteration k, we update the information of i_k-th individual function, where $i_k = (k \mod N) + 1$. Specifically, we have

$$z_{i_k}^{k+1} = x_{k+1}, \quad z_i^{k+1} = z_i^k \quad \text{for } i \neq i_k. \tag{98}$$

Hence, with these settings, while the information of the selected function f_{i_k} is updated, the other terms are kept the same with their previous value. Moreover, it follows that

$$\nabla f_{i_k}(z_{i_k}^{k+1}) = \nabla f_{i_k}(x_{k+1}), \tag{99}$$

$$\nabla f_i(z_i^{k+1}) = \nabla f_i(z_i^k) \quad \text{for } i \neq i_k. \tag{100}$$

According to (98)-(100), we can derive the following for Hessian approximation update:

$$B_i^{k+1} = B_i^k + \frac{(y_i^k - B_i^k s_i^k)(y_i^k - B_i^k s_i^k)^T}{(y_i^k - B_i^k s_i^k)^T s_i^k}, \tag{101}$$

for $i = i_k$ and $B_i^{k+1} = B_i^k$ for $i \neq i_k$, where $y_i^k = \nabla f(z_i^{k+1}) - \nabla f(z_i^k)$ and $s_i^k = z_i^{k+1} - z_i^k$. Obviously, for $i \neq i_k$, it indicates $y_i^k = 0$ and $s_i^k = 0$. This leads to our efficient computation of $\{\tilde{B}^k, \tilde{z}^k, \tilde{g}^k\}$ in (97). To be specific, $\{\tilde{B}^k, \tilde{z}^k, \tilde{g}^k\}$ can be updated as follows:

$$\tilde{B}^{k+1} = \tilde{B}^k - B_{i_k}^k + B_{i_k}^{k+1}, \tag{102}$$

$$\tilde{z}^{k+1} = \tilde{z}^k - B_{i_k}^k z_{i_k}^k + B_{i_k}^{k+1} z_{i_k}^{k+1}, \tag{103}$$

$$\tilde{g}^{k+1} = \tilde{g}^k - \nabla f(z_{i_k}^k) + \nabla f(z_{i_k}^{k+1}). \tag{104}$$

Therefore, we avoid the computation of the large sample summation to update $\{\tilde{B}^k, \tilde{z}^k, \tilde{g}^k\}$. Moreover, since updating the iterate in (97) requires matrix inverse operation, it is desirable to update its inverse to avoid direct matrix inversion. Substituting (101) into (102) leads to:

$$\tilde{B}^{k+1} = \tilde{B}^k + \frac{(y_{i_k}^k - B_{i_k}^k s_{i_k}^k)(y_{i_k}^k - B_{i_k}^k s_{i_k}^k)^T}{(y_{i_k}^k - B_{i_k}^k s_{i_k}^k)^T s_{i_k}^k}. \tag{105}$$

By applying the Sherman-Morrison formula to (105), we obtain the following update of the inverse of the approximated Hessian:

$$\tilde{H}^{k+1} = \tilde{H}^k - \frac{\tilde{H}^k(y_{i_k}^k - B_{i_k}^k s_{i_k}^k)(y_{i_k}^k - B_{i_k}^k s_{i_k}^k)^T \tilde{H}^k}{(y_{i_k}^k - B_{i_k}^k s_{i_k}^k)^T s_{i_k}^k} \tag{106}$$

where for simplicity we define $\tilde{H}^k := (\tilde{B}^k)^{-1}$. The computational complexity of (106) is $\mathcal{O}(d^2)$ while the cost of direct matrix inversion is $\mathcal{O}(d^3)$, thus substantially reducing the computational complexity. Furthermore, to ensure that each update of the Hessian approximation $B_{i_k}^{k+1}$ is positive definite, we can apply CuREG-SR1 in (25) to the resulting update. To be specific, if $(y_{i_k}^k - B_{i_k}^k s_{i_k}^k)^T s_{i_k}^k < 0$ and Case II happens, we choose $M_{i_k}^k = -b/2a$ where a and b are calculated with (28) and (29) respectively. For other cases, we skip the update. In this way, $-B_{i_k}^k + B_{i_k}^{k+1}$ is positive definite, and thus $\tilde{B}^{k+1} \succeq 0$ in (102). We summarize the ICuREG-SR1 algorithm in *Algorithm 2*.

Remark. In steps 8 and 12, it should be noted from (105) that if the resultant Hessian approximation matrix $\tilde{B}^{k+1}$ is ill-conditioned, an effective regularization technique can be applied. Specifically, we have

$$\tilde{H}^{k+1} \leftarrow (\tilde{B}^{k+1} + rI)^{-1}, \tag{107}$$

with $r := \frac{\rho \text{Tr}(\tilde{B}^{k+1})}{d}$, typical values of ρ are 10^{-2} and 10^{-3} etc.

For N samples with feature dimension d, if each iteration only updates one sample, there will be N Hessian approximation matrices for the corresponding individual functions.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2019.2935900, IEEE Access

Algorithm 2 ICuREG-SR1

Input: set $z_i^0 = x_0$ for randomly generated x_0 from uniform distribution $[-1,1]^d$, $B_i^0 = I$ for $i = 1, \cdots, N$, the desired iteration number KN.

Output: $z_N^{KN} = x^{KN}$.

for $k = 0, 1, \ldots, KN-1$ **do**
  Calculate x^{k+1}: $x_{k+1} = (\tilde{B}^k)^{-1}(\tilde{z}^k - \tilde{g}^k)$.
  Set $i_k = (k \bmod N) + 1$.
  Compute $y_{i_k}^k$ and $s_{i_k}^k$ according to (98)-(100).
  Set $\tilde{H}_{k+1} = \tilde{H}_k$,
  if $|(y_{i_k}^k - B_{i_k}^k s_{i_k}^k)^T s_{i_k}^k| > \epsilon \left\| y_{i_k}^k - B_{i_k}^k s_{i_k}^k \right\| \left\| s_{i_k}^k \right\|$ **then**
    if $(y_{i_k}^k - B_{i_k}^k s_{i_k}^k)^T s_{i_k}^k > 0$ **then**
      Calculate $\tilde{H}_{k+1}$ via (106),
    else
      Compute a,b and c according (28)(29) and (30) respectively,
      if $b^2 - 4ac > 0$ and $b < 0$ **then**
        Set $M_{i_k}^k = -\frac{b}{2a}$ and calculate $\tilde{H}_{k+1}$ by substituting $\tilde{y}_{i_k}^k := y_{i_k}^k + \frac{M_{i_k}^k}{2} \left\| s_{i_k}^k \right\| s_{i_k}^k$ into (106) in lieu of $y_{i_k}^k$,
      end if
    end if
  end if
  Compute $\tilde{z}^{k+1}$ and $\tilde{g}^{k+1}$ according to (103) and (104) respectively.
end for

Without loss of generality, suppose that the incremental algorithm starts by using the first sample (corresponding to first individual function). The memory cost for this scenario will be $O(Nd^2 + Nd)$. Hence, for large scale problems, the incremental method suffers from the problem of large memory cost. However, by grouping L individual functions into a new individual function, the memory cost can be substantially reduced to $O(\frac{Nd^2+Nd}{L})$. Now we illustrate the existing technique for further reducing the memory cost. At the $(KN+1)$, the first variable denoted as z_1^{KN} has been updated K times. Consider limited memory SR1 (chapter 9.16 in [23]), it stores the correction pairs (s_1^k, y_1^k) with respect to the first variable z_1^k for $k = (K-m)\cdot N+1, \ldots, (K-1)\cdot N+1$. Similarly, for all components z_i^k, $i = 1...N$, the memory cost will be $O(2dmN)$. Moreover, if one groups L individual functions as one new individual function, the memory cost will be further reduced to $O(2dmN/L)$. Here, our main purpose is to propose a framework of incremental method. It can be a future work to further refine the implementation of the above limited memory version of our proposed ICuREG-SR1.

V. NUMERICAL RESULTS

In this section, we conduct numerical tests on our proposed ICuREG-SR1 algorithm. We will apply the algorithm to logistic regression and Bayesian logistic regression. For the latter, we adopt Laplace method [32]. We also implement the conventional SGD, SdLBFGS [13] and Sd-REG-LBFGS [42] algorithms as comparison for the above problems. Note that the latter two optimization schemes are based on limited memory BFGS scheme (LBFGS). The performance evaluation will be based on the norm of the gradient at each iteration. Furthermore, we use three datasets for the tests, namely the synthetic dataset randomly generated, the *scene* dataset and the *CIFAR-10* dataset. For fair comparison, the parameters have been tuned to yield the best performance of each stochastic algorithm in the numerical experiments.

A. LOGISTIC REGRESSION

We first consider the logistic regression for binary classification [32]. Suppose two classes denoted as $z_n = 0$ and $z_n = 1$ are to be recognized respectively. Logistic regression models the problem as $p(z_n|\theta) = \sigma(\theta^T x_n)^{z_n} \cdot (1 - \sigma(\theta^T x_n))^{1-z_n}$, where θ is the parameter to be identified, x_n is the feature vector and $\sigma(\cdot)$ is the sigmoid function given by $\sigma(x) = 1/(1 + \exp(-x))$. Given the training data $\{z_n, x_n\}$ with $x_n \in \mathbb{R}^d$ and $n = 1, \cdots, N$, the likelihood function to be maximized is $p(z_{1:N}|\theta) = \prod_{n=1}^N p(z_n|\theta)$. One can maximize the log-likelihood function or equivalently minimize the objective function: $f(\theta) = -\frac{1}{N}\sum_{n=1}^N z_n \log \sigma(\theta^T x_n) + (1 - z_n)\log \sigma(-\theta^T x_n)$. Moreover, we use the norm of gradient (NOG) for performance evaluation. The NOG for logistic regression is defined as follows:

$$\mathrm{NOG}_{LR} = \left\| \frac{1}{N} \sum_{n=1}^{N} [z_n - \sigma(\theta^T x_n)] x_n \right\|. \tag{108}$$

B. LAPLACE METHOD FOR BAYESIAN LOGISTIC REGRESSION

Bayesian treatment of logistic regression can be effected by introducing a prior distribution to the parameter θ, say with a Gaussian prior $p(\theta) = \mathcal{N}(m_0, S_0)$, to obtain the posterior distribution [32]. The Laplace method aims to approximate a posteriori distribution at a local maximum of the likelihood by using Taylor expansion. Consider the a posteriori distribution $p(\theta|x_{1:N}, z_{1:N})$ and prior $p(\theta)$, it satisfies $p(\theta|x_{1:N}, z_{1:N}) \propto \exp\{\log\, p(\theta) + \sum_{n=1}^N \log\, p(z_n|\theta)\}$. Let us define $k(\theta) := \log\, p(\theta) + \sum_{n=1}^N \log\, p(z_n|\theta)$. By using second-order Taylor expansion at its maximum point $\hat{\theta}$, we have $p(\theta|x, z) \propto \exp\{k(\theta)\} \approx \exp\{\frac{1}{2}(\theta - \hat{\theta})^T \nabla^2 k(\hat{\theta})(\theta - \hat{\theta})\}$, as $\hat{\theta}$ is the maximum point with $\nabla k(\hat{\theta}) = 0$. Note that $\hat{\theta}$ is also the maximum a posteriori (MAP) estimator of the global variable. Subsequently, it yields $p(\theta|x, z) \approx \mathcal{N}(\hat{\theta}, -\nabla^2 k(\hat{\theta}))$. Therefore, the Laplace method converts the inference problems into a MAP estimation problem $\hat{\theta} = \mathrm{argmax}_{\hat{\theta}}\ k(\theta)$. A key step for finding the MAP estimator is the optimization process involving the determination of the gradient for search direction. However, since $k(\theta)$ contains the sum of log-likelihood of the samples, the evaluation of the exact gradient requires excessive computation especially for large number of samples. Here, we adopt the proposed ICuREG-SR1 for solving the MAP estimate, which amounts to solving $\hat{\theta} = \mathrm{argmin}_{\theta \in \mathbb{R}^d}\ -\frac{1}{N}k(\theta)$. Furthermore, the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2019.2935900, IEEE Access

NOG for Bayesian logistic regression can be calculated as follows:

$$\text{NOG}_{BLR} = \left\| \frac{1}{N}\{\sum_{n=1}^{N}[z_n - \sigma(\theta^T x_n)]x_n + S_0^{-1}(\theta - m_0)\} \right\|. \tag{109}$$

C. EXPERIMENTS WITH SYNTHETIC DATASET

We first study a synthetic dataset with dimension $d = 150$. For the two classification problems, we set the same initial optimization variable to be θ_0, which is generated from Gaussian distribution $\mathcal{N}(0, I)$. We generate $1,000$ synthetic training data points in the following manner. First, generate the feature vectors x_n, $i = 1, \ldots, N$ using the uniform distribution $[0, 1]^d$. Next, use a specified vector $\bar{\theta}$ generated from the uniform distribution $[-1, 1]^d$ to generate the corresponding label $z_n = \mathbb{I}(\bar{\theta}^T x_n > 0)$.

The Sd-REG-LBFGS [42], SdLBFGS [13] and SGD algorithms are implemented as comparison. The basic idea of Sd-REG-LBFGS is to add a regularization parameter γ to avoid singular matrix [42]. Moreover, it requires another parameter δ to satisfy $\delta > 1.25\gamma$ to ensure positive definiteness. The parameters $\gamma = 10^{-4}$ and $\delta = 1.25\gamma + 0.01$ are employed. Additionally, the parameter β which is used for initializing the LBFGS method is set to $\beta = 0.1$. Sd-REG-LBFGS also performs average of the iterate every L iterations, L is set to $L = 10$ as a trade off between performance and complexity. In the LBFGS process, the memory is set to $M = 10$. For SdLBFGS, the same values for β and M are used. Furthermore, the stepsize for SGD, SdLBFGS and Sd-REG-LBFGS is set to be $\eta_k = 7/k$. The parameters are tuned to exhibit as best performance of the three optimization scheme as possible.

1) Logistic Regression

Numerical experiments were performed on the generated synthetic dataset using classical logistic regression. Figure 1 shows the numerical results of the performance evaluation in terms of gradient magnitude, i.e., NOG. According to Algorithm 2, our proposed method updates one individual function at each iteration, which also corresponds to one specific data point. This indicates that our proposed method will update all the individual functions over the whole dataset every 1,000 iterations as there are 1,000 data points. Moreover, these 1,000 iterations are called local iterations for convenience. It can be seen from Figure 1 that our proposed method outperforms SGD after 50 passes through the whole dataset, which has a gradient magnitude of 0.065. Furthermore, ICuREG-SR1 started to outperform Sd-REG-LBFGS at the 120th pass. Additionally, it takes 370 passes for our proposed method to have better performance than SdLBFGS and the corresponding NOG is 0.01. ICuREG-SR1 continues to descend afterwards. After a total of 600 passes over the whole dataset, the magnitude of its gradient is as small as 0.006. A subplot is used to observe the detail in the local iterations of one pass. Specifically, the $500 \sim 501$ pass is depicted. The NOG first decreases and then increases again but overall, the NOG value decreases as the number of iterations increases.

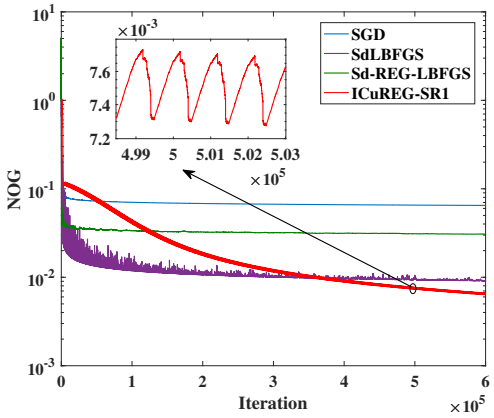

FIGURE 1: The NOG performance of different algorithms applied in solving logistic regression using *synthetic* dataset.

2) Bayesian Logistic Regression

Figure 2 presents the performance of various algorithms in terms of gradient magnitude for solving Bayesian logistic regression with the synthetic dataset. It can be seen that SGD and SdLBFGS exhibit similar performance. They both reach a gradient magnitude of 0.075 at convergence. Meanwhile, Sd-REG-LBFGS scheme has better performance than the above two methods. The NOG value is 0.038 at convergence, which is reduced by 49.3% compared with the two methods. For our proposed method, it performs the best as it shows the smallest NOG value for the specific iteration number. Similar to the scenario of logistic regression with synthetic dataset in Figure 1, the learning curve of our proposed method started to outperform both SGD and SdLBFGS at 60 passes over the whole synthetic dataset, and then has better NOG performance than Sd-REG-LBFGS after 115 passes. The subplot also shows the similar result to Figure 2 that the NOG first decreases and then increases within one pass.

D. NUMERICAL RESULTS WITH *SCENE* DATASET

In this subsection, we compare Sd-REG-LBFGS with SdLBFGS and SGD for a practical dataset called the *scene* dataset [51]. It contains 1,211 images in the training set and 1,196 in the testing set. Each image has 294 features and up to 6 scene labels: *beach*, *sunset*, *fall-foliage*, *field*, *mountain* and *urban*. To form the binary classification problem, we simply group *beach*, *sunset* and *fall-foliage* as a new category with label $z = 1$. The remain classes are grouped with label $z = 0$. The binary classification problem is to predict whether an image in the testing set is in the new category [31]. The parameters of various optimization schemes are the same as the numerical experiments for the synthetic dataset.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2019.2935900, IEEE Access

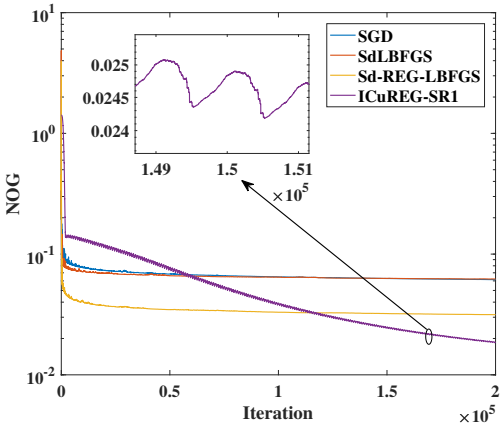

FIGURE 2: The NOG performance of different algorithms applied in solving Bayesian logistic regression using *synthetic* dataset.

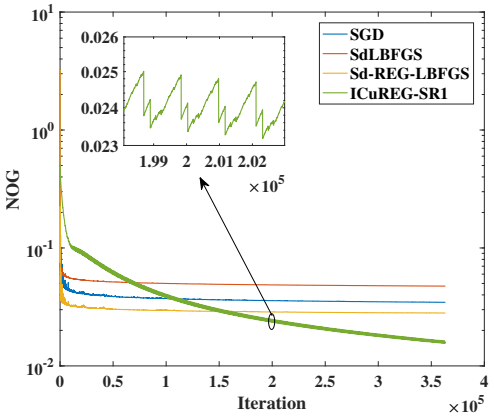

FIGURE 3: The NOG performance of different algorithms applied in solving logistic regression using *scene* dataset.

1) Logistic Regression

Figure 3 shows the performance comparison of the four algorithms in terms of the NOG. The result shows that our method generally outperforms the other methods as our method has the smallest NOG value, which indicates that our method is the closest to the critical point. Additionally, our proposed method exhibits a tendency of continuing to reduce the gradient magnitude while the other methods have converged. From the subplot, we can draw a similar conclusion that our proposed method exhibits an initial decreased gradient magnitude and then increased NOG value within one pass over the whole *scene* dataset.

2) Bayesian Logistic Regression

Figure 4 compares the performance of different methods when applying the Laplace scheme to Bayesian logistic regression. The *scene* dataset is used. Figure 4 generally shows that our proposed method performs the best in terms of the NOG. Similar to Figure 3, ICuREG-SR1 continues

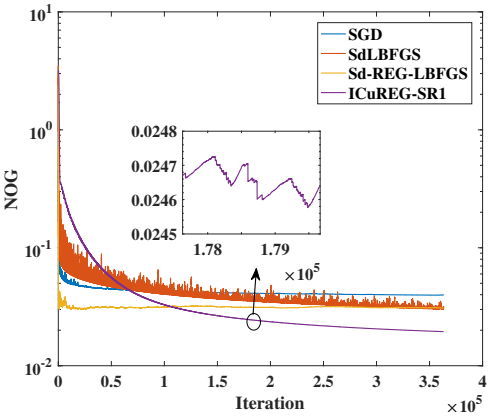

FIGURE 4: The NOG performance of different algorithms applied in solving Bayesian logistic regression using *scene* dataset.

to decrease the NOG value while the other methods have converged. Furthermore, our proposed method has improved the performance by more than 33.3% compared to Sd-REG-LBFGS in terms of gradient magnitude. In addition, within one pass over the whole dataset, the subplot shows that the NOG values exhibit certain variations with the iterations, but the NOG value is decreased as a whole as the number of iterations increases.

E. NUMERICAL RESULTS WITH *CIFAR-10* DATASET

In this subsection, we compare the proposed algorithm with the Sd-SEG-LBFGS, SdLBFGS and SGD algorithms with a large practical dataset: *CIFAR-10* dataset [50]. We randomly choose 10,000 data points from the CIFAR-10 dataset, where each feature vector has dimension 3072 and its label ranges from 0 to 9. In the paper, we consider our proposed method in the optimization of binary classification problems. Thus, we regroup the data points into two classes: if the label is less equal than 4, we set its new label to 0, i.e., $z_n = 0$; for other cases, $z_n = 1$. Moreover, as the dataset is large, updating a single function at each iteration is inefficient in terms of arithmetic complexity and memory cost. Hence, we group 100 single functions as a new individual function. For the regrouped objective function, there are 100 individual functions now. We set the iteration number to terminate the algorithm as 300 times of the number of individual functions. This has saved much memory and substantially reduced the iteration number required to reach a desired point, which is sufficiently close to the critical point. Moreover, we have used the effective regularization technique in (107) for numerical stability.

1) Logistic Regression

Figure 5 shows the performance of various approaches for solving the logistic regression using the *CIFAR-10* dataset. Our proposed method generally exibits a stable learning curve and the NOG gradually descends. However, it also os-

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2019.2935900, IEEE Access

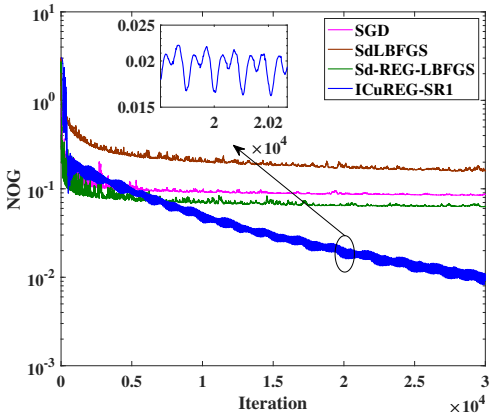

FIGURE 5: The NOG performance of different algorithms applied in solving logistic regression using *CIFAR-10* dataset.

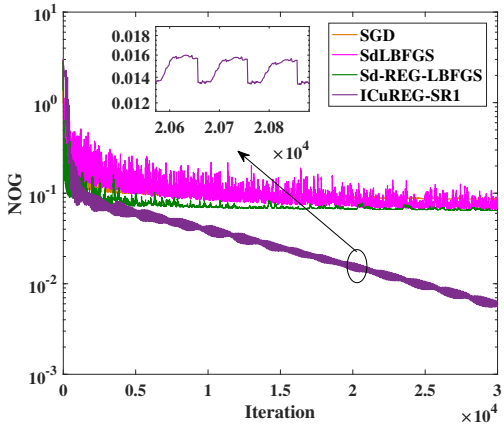

FIGURE 6: The NOG performance of different algorithms applied in solving Bayesian logistic regression using *CIFAR-10* dataset.

cillates within local iterations. Since there are 100 individual functions, the number of local iterations within one pass is 100. It can be seen that NOG generally decreases for one pass over the individual functions. Moreover, it has shown that our proposed method has the lowest gradient magnitude, and continues to descend while other methods have reached convergence.

2) Bayesian Logistic Regression

Figure 6 shows the performance of various methods for solving the Bayesian logistic regression using the CIFAR-10 dataset. It shows that our proposed method has the best performance in terms of NOG, which also means that the solution of our proposed method is closer to the critical point. Similarly, the NOG curve generally decreases in each pass over the individual functions.

VI. CONCLUSION

We have extensively studied the convergence properties of CuREG-SR1 and derived novel results. It should be noted that direct applying classical methods that analyze convergence properties of SR1 can be problematic. For example, in *Theorem 2*, the difference between the approximated Hessian and true Hessian is bounded. However for SR1, the approximated Hessian converges to the true Hessian. This is due to the reason that incorporating cubic regularized technique introduces the shift. Hence, we restric the regularized parameter, under which, for sufficiently large iteration numbers, we have proofed that there are $q - d$ superlinear steps in every $q \geq d + 1$ steps. Furthermore, we proposed a novel incremental optimization method based on SR1 and CuREG-SR1 and its efficient implementation to solve large scale problems. In the numerical experiments, we apply ICuREG-SR1 to logistic regression and its Bayesian treatment using artificial dataset and real world dataset respectively. It has shown that our proposed method is powerful and superior in terms of the gradient magnitude, compared to other three schemes.

REFERENCES

[1] C. G. Broyden, "Quasi-Newton methods and their application to function minimisation,"*Mathematics of Computation*, 21(99), 368 - 381, 1969.
[2] H. Benson and Y. Shanno, "Cubic regularization in symmetric rank-1 quasi-Newton methods. Mathematical Programming Computation,"*Mathematics of Computation*, 10(4), 457 - 486, 2018.
[3] H. Khalfan, R. Byrd and R. Schnabel, "A Theoretical and Experimental Study of the Symmetric Rank-One Update,"*SIAM J. Optim.*, vol. 3, no. 1, pp. 1 - 24, Feb. 1993.
[4] C. G. Broyden, J. E. Dennis and J. Moré, "On the Local and Superlinear Convergence of Quasi-Newton Methods ,"*IMA J. Appl. Math.*, vol. 13, no. 3, pp. 223 - 245, Dec. 1973.
[5] A. Conn, N. Gould and P. Toint, "Convergence of quasi-Newton matrices generated by the symmetric rank one update,"*Mathematical Programming*, vol. 50, no. 1-3, pp. 177 - 195, Mar. 1991.
[6] J. E. Dennis and J. Moré, "A Characterization of Superlinear Convergence and Its Application to Quasi-Newton Methods,"*Mathematics of Computation*, Vol. 28, No. 126, pp. 549 - 560, Apr., 1974.
[7] W. C. Davidon, "Variable metric method for minimization,"*SIAM J. Optim.*, Vol. 1, No. 1, pp. 1 - 17, 1991.
[8] C. G. Broyden, "The convergence of a class of double-rank minimization algorithms 2. the new algorithm ,"*IMA J. Appl. Math.*, vol. 6, no. 1, pp. 222 - 231, 1970.
[9] D. Xu, D. Jian and C. Zhang " Convergence of Quasi-Newton Method for Fully Complex-Valued Neural Networks , "*Neural Processing Letters*, vol. 46, no. 3, pp. 961 - 968, Dec. 2017.
[10] C. Cartis, N. Gould and P. Toint, "Adaptive cubic regularisation methods for unconstrained optimization. Part I: motivation, convergence and numerical results, "*Mathematical Programming*, vol. 127, no. 2, pp. 245 - 295, April. 2011.
[11] Y. Nesterov and B. Polyak "Cubic regularization of Newton method and its global performance, "*Mathematical Programming*, vol. 108, no. 1, pp. 177 - 205, Aug. 2006.
[12] A. Mokhtari and A. Ribeiro, "RES: Regularized Stochastic BFGS Algorithm,"*IEEE Trans. Signal Process.*, vol. 62, no. 23, pp. 6089 - 6104, Dec.1, 2014.
[13] 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.
[14] 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.
[15] R. H. Byrd, S. L. Hansen, Jorge Nocedal, and Y. Singer, "A Stochastic Quasi-Newton Method for Large-Scale Optimization,"*SIAM J. Optim.*, vol. 26, no. 2, pp. 1008 - 1031, 2016.

[16] H. Chen and W. H. Lam, "Training Based Two-Step Channel Estimation in Two-Way MIMO Relay Systems,"*IEEE Trans. on Veh. Tech.*, vol. 67, no. 3, 2193 - 2205, Mar.2018.

[17] H. Chen and W. H. Lam, "Training design and two-stage channel estimation for correlated two-way MIMO relay systems,"*International Conference on Ubiquitous and Future Networks (ICUFN)* , Vol. 67 , Vienna, June. 2016, pp. 307 - 312.

[18] L. Zhang, H. C. Wu, C. H. Ho and 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.*.

[19] 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.

[20] 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.

[21] M. J. D. Powell, "Algorithms for nonlinear constraints that use lagrangian functions,"*Math. Programming*, vol. 14, no. 1, pp. 224 - 248, Dec., 1978.

[22] R. Fletcher, "Practical Methods of Optimization,"*Wiley, Chichester*, 1987.

[23] J. Nocedal and S. J. Wright, *Numerical Optimization*, New York:Springer-Verlag, 1999.

[24] K. Levenberg, "A method for the solution of certain problems in least squares,"*Quart. Appl. Math.*, vol. 2, no. 2, pp. 164 - 168, July, 1944.

[25] D. Marquardt, "An algorithm for least-squares estimation of nonlinear parameters,"*SIAM J. Appl. Math.*, vol. 11, no. 2, pp. 431 - 441, June, 1963.

[26] D. Rosen, M. Kaess and J. Leonard, "RISE: An Incremental Trust-Region Method for Robust Online Sparse Least-Squares Estimation,"*IEEE Trans. on Robotics* , vol. 30, no. 5, 1091 - 1108, Oct.2014.

[27] Q. Shi and C. He, "A New Incremental Optimization Algorithm for ML-Based Source Localization in Sensor Networks ,"*IEEE Signal Processing Letters*, vol. 15, pp. 45 - 48, Jan., 2008.

[28] D. Blatt and A. Hero, "Energy-based sensor network source localization via projection onto convex sets ,"*IEEE Trans. Signal Process.*, vol. 54, no. 9, pp. 3614 - 3619, Aug., 2006.

[29] S. Babacan, S. Nakajima and M. Do, "Bayesian Group-Sparse Modeling and Variational Inference ,"*IEEE Trans. Signal Process.*, vol. 62, no. 11, pp. 2906 - 2921, June, 2014.

[30] J. Taghia and A. Leijon, "Variational Inference for Watson Mixture Model,"in *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 9, pp. 1886 - 1900, Sept., 2016.

[31] C. Wang and D. M. Blei, "Variational Inference in Nonconjugate Models,"*J. Mach. Learn. Res.*, vol. 14, no. 1, pp. 1005 - 1031, Jan., 2013.

[32] C. Bishop, *Pattern Recognition and Machine Learning*, Springer New York., 2006.

[33] 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.

[34] J. Paisley, D. M. Blei and M. I. Jordan, "Variational Bayesian Inference with Stochastic Search,"in *Proc. Int. Conf. Mach. Learn*, vol. 14, pp. 1363 - 1370, 2012.

[35] 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.

[36] H. Robbins and S. Monro, "A Stochastic Approximation Method,"*Ann. Math. Statist.*, vol. 22, no. 3, pp. 400 - 407, 1951.

[37] 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.

[38] 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.

[39] A. Nemirovski, A. Juditsky, G. Lan and A. Shapiro, "Robust Stochastic Approximation Approach to Stochastic Programming,"*SIAM J. Optim.*, vol. 19, no. 4, pp. 1574 - 1609, 2009.

[40] 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.

[41] 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.

[42] H.Chen, etc, "A Stochastic Quasi-Newton Method for Large-Scale Nonconvex Optimization with Applications, "submitted to *IEEE Trans. Neural Netw. Learn. Syst*.

[43] Z. Ghahramani, "Probabilistic machine learning and artificial intelligence,"*Nature*, vol. 521, no. 7553, pp. 452 - 459, 2013.

[44] Sun Yi, D. Wierstra, T. Schaul and J. Schmidhuber, "Stochastic search using the natural gradient,"in *Proc. Int. Conf. Mach. Learn*, vol. 382, pp. 1161 - 1168, 2009.

[45] 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.

[46] P. Moritz, 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.

[47] A. Honkela, T. Raiko, M. Kuusela, M. Torni and Juha Karhunen, "Approximate Riemannian Conjugate Gradient Learning for Fixed-Form Variational Bayes,"*J. Mach. Learn. Res.*, vol. 11, pp. 3235 - 3268, Dec., 2010.

[48] S. Amari, *Information Geometry and Its Applications*, Springer Japan, 2016.

[49] S. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge University Press New York., 2004.

[50] Alex Krizhevsky, *Learning Multiple Layers of Features from Tiny Images*, 2009.

[51] M.R. Boutell, J. Luo, X. Shen and C.M. Brown, "Learning multi-label scene classiffication,"*Pattern Recognition*, vol. 37, no. 9, pp. 1757-1771, 2004.

[52] D. Kingma and J. Ba, "Adam: A method for stochastic optimization", in *3rd International Conference for Learning Representations, 2015*.

[53] T. Tieleman and G. Hinton, "Lecture 6.5 - RMSProp, COURSERA: Neural Networks for Machine Learning", Technical report, 2012.