A DIAGONAL-AUGMENTED QUASI-NEWTON METHOD WITH APPLICATION TO FACTORIZATION MACHINES

Aryan Mokhtari[†⋆] *and Amir Ingber*[⋆]

[†]Department of Electrical and Systems Engineering, University of Pennsylvania, PA, USA
[⋆]Big-data Machine Learning Group, Yahoo!, Sunnyvale, CA, USA

ABSTRACT

We present a novel quasi-Newton method for convex optimization, in which the Hessian estimates are based not only on the gradients, but also on the diagonal part of the true Hessian matrix (which can often be obtained with reasonable complexity). The new algorithm is based on the well known Broyden-Fletcher-Goldfarb-Shanno (BFGS) algorithm and has similar complexity. The proposed Diagonal-Augmented BFGS (DA-BFGS) method is shown to be stable and achieves a super-linear convergence rate in a local neighborhood of the optimal argument. Numerical experiments on logistic regression and factorization machines problems showcase that DA-BFGS consistently outperforms the baseline BFGS and Newton algorithms.

Index Terms— Quasi-Newton methods, partial Hessian information, factorization machines

1. INTRODUCTION

The problem of minimizing a convex function arises in different aspects of machine learning. In particular, many machine learning problems such as support vector machines, logistic regression, least squares and factorization machines boil down to minimizing the average of a set of simple convex functions [1–3]. Consider the optimization variable $\mathbf{x} \in \mathbb{R}^p$ as the input of the function $f : \mathbb{R}^p \to \mathbb{R}$ where the function f can be written as the average of N convex functions $\{f_i\}_{i=1}^N$, i.e, $f(\mathbf{x}) := (1/N)\sum_{i=1}^N f_i(\mathbf{x})$. The goal is to find the optimal argument of the function f,

$$\mathbf{x}^* := \operatorname*{argmin}_{\mathbf{x}} f(\mathbf{x}) := \operatorname*{argmin}_{\mathbf{x}} \frac{1}{N}\sum_{i=1}^{N} f_i(\mathbf{x}). \tag{1}$$

Problems of this form also arise in control problems [4–6], and wireless communication [7–9].

The gradient descent (GD) method is well-known tool for solving convex optimization problems [10, 11]. It has relatively low computational complexity of order $\mathcal{O}(Np)$ per iteration. While GD achieves a linear convergence rate, the actual convergence can be very slow, especially when the function f is ill-conditioned. The accelerated version of gradient descent improves the convergence rate of the vanilla gradient descent, but still has a linear convergence rate that depends on the square root of the condition number of the function's Hessian [12, 13]. Newton's method arises as a natural solution for solving ill-conditioned problems. It improves the convergence speed of first-order methods by incorporating second-order information [10, 11], and achieves *quadratic* convergence (which is significantly faster than linear rate). However, the implementation of Newton's method requires computing the objective function Hessian and its inverse at each iteration. Thus, the overall computational complexity of Newton's method per iteration is of the order $\mathcal{O}(Np^2+p^3)$ which is not computationally affordable in large-scale problems.

Quasi-Newton methods such as Broyden's method, Davidon-Fletcher-Powell (DFP), and Broyden-Fletcher-Goldfarb-Shanno (BFGS) sit at the sweet spot of affordable computation complexity and fast convergence rate [14–16]. These methods try to approximate the Hessian of the function f using its first-order information (i.e., function gradients). Therefore, quasi-Newton methods do not require computation of the Hessian or its inverse, and their overall computational complexity is of the order $\mathcal{O}(Np + p^2)$ which is similar to gradient descent methods. In addition, they enjoy a fast super-linear convergence rate.

While the computation of the full Hessian matrix is costly, in some applications partial information of the Hessian $\nabla^2 f(\mathbf{x})$ is either available or easy to compute. This justifies the use of this partial Hessian information in the update of quasi-Newton methods. Such algorithms are generally termed *structured quasi-Newton methods* [17, 18]. They incorporate partial Hessian information in the update of quasi-Newton methods and achieve super-linear convergence. However, these methods are not globally convergent and only guaranteed to converge when the variable is close enough to the optimal solution, which limits their applicability.

In this paper we develop a novel quasi-Newton method called Diagonal-Augmented BFGS (DA-BFGS) which incorporates diagonal part of the Hessian in the Hessian inverse approximation of BFGS. The proposed DA-BFGS method is globally convergent with a linear rate and has a super-linear convergence rate in a local neighborhood of the optimal argument, while it has a low computational complexity per iteration of the order $\mathcal{O}(Np + p^2)$, like the BFGS algorithm. In the next sections, we first provide a summary of the BFGS method (Section 2). Then, we introduce the DA-BFGS method which uses the diagonal part of the Hessian matrix (Section 3). We show that DA-BFGS is globally convergent and has super-linear convergence rate in a neighborhood of the optimal argument (Section 4). Further, we evaluate the performance of DA-BFGS on logistic regression and factorization machine problems (Section 5). Finally, we close the paper by concluding remarks (Section 6). Proofs of results in this paper are available in [19].

2. BFGS METHOD

To reduce the computation time required for Newton's method, quasi-Newton (QN) methods such as Broyden's method, DFP and BFGS were developed. These methods are globally convergent and enjoy a fast super-linear convergence rate in a local neighborhood of the optimal argument. It has been shown that BFGS has the best performance among the quasi-Newton methods [20]. Thus, here we focus on BFGS and its variants.

The main idea of BFGS (and other QN methods) is to approximate the Hessian inverse of the objective function using the evaluated gradients. In particular, define k as the time index and $\mathbf{x}_k$ as the variable at iteration k. Then, the BFGS update at step k with step-size ϵ_k is given by

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \epsilon_k \mathbf{B}_k^{-1} \nabla f(\mathbf{x}_k), \tag{2}$$

where $\mathbf{B}_k$ is a positive definite matrix that approximates the Hessian $\nabla^2 f(\mathbf{x}_k)$ associated with the variable $\mathbf{x}_k$. Note that if we replace $\mathbf{B}_k$ in (2) by the Hessian $\nabla^2 f(\mathbf{x}_k)$ we recover the update of Newton's method.

To understand the rationale behind the QN Hessian approximation, first define $\mathbf{s}_k$ and $\mathbf{y}_k$ as the variable and gradient variations associated to the time index k which are explicitly given by

$$\mathbf{s}_k := \mathbf{x}_{k+1} - \mathbf{x}_k, \qquad \mathbf{y}_k := \nabla f(\mathbf{x}_{k+1}) - \nabla f(\mathbf{x}_k). \tag{3}$$

In the BFGS method, we use the fact that the Hessian $\nabla^2 f(\mathbf{x})$ satisfies the *secant condition* $\nabla^2 f(\mathbf{x}_{k+1})\mathbf{s}_k = \mathbf{y}_k$ when the variables $\mathbf{x}_k$ and

$\mathbf{x}_{k+1}$ are close to each other. Thus, the Hessian approximation matrix of BFGS is chosen such that it satisfies the secant condition, i.e., $\mathbf{B}_{k+1}\mathbf{s}_k = \mathbf{y}_k$. However, this condition does not lead to a unique solution. To resolve this issue we pick the matrix $\mathbf{B}_{k+1}$ in a way that the secant condition is satisfied and the matrix $\mathbf{B}_{k+1}$ is the closest matrix to the previous Hessian approximation $\mathbf{B}_k$, according to a certain distance measure [14]. This proximity condition in conjunction with the secant condition implies that the Hessian inverse approximation $\mathbf{B}_{k+1}^{-1}$ can be evaluated as

$$\mathbf{B}_{k+1}^{-1} = \mathbf{B}_k^{-1} + \frac{(\mathbf{s}_k - \mathbf{B}_k^{-1}\mathbf{y}_k)\mathbf{s}_k^T + \mathbf{s}_k(\mathbf{s}_k - \mathbf{B}_k^{-1}\mathbf{y}_k)^T}{\mathbf{s}_k^T\mathbf{y}_k} - \frac{\mathbf{y}_k^T(\mathbf{s}_k - \mathbf{B}_k^{-1}\mathbf{y}_k)\mathbf{s}_k\mathbf{s}_k^T}{(\mathbf{s}_k^T\mathbf{y}_k)^2}. \tag{4}$$

The update in (4) only utilizes first-order information, in the form of the gradient variation. The computation cost of the descent direction $\mathbf{d}_t = \mathbf{B}_k^{-1}\nabla f(\mathbf{x}_k)$ evaluation is of order $\mathcal{O}(Np + p^2)$ where Np corresponds to gradient evaluation and p^2 to operations on matrices of size $p \times p$.

Although BFGS is successful in solving large-scale optimization problems, it only depends on the first order information. In some applications, partial information of the Hessian is either available or cheap to compute. In the following section, we propose a QN method that tries to incorporate this partial information in the update of BFGS.

3. MAIN RESULT

In this section, we propose a variant of the BFGS method called Diagonal-Augmented BFGS (DA-BFGS). The DA-BFGS method tries to exploit the diagonal information about the objective function Hessian in the update of BFGS. In particular, consider the matrix $\mathbf{D}(\mathbf{x})$ as a diagonal matrix which contains the diagonal components of the Hessian $\nabla^2 f(\mathbf{x})$. We assume the Hessian inverse has the general form of $\nabla^2 f(\mathbf{x})^{-1} = \mathbf{D}(\mathbf{x})^{-1} + \mathbf{A}(\mathbf{x})$, where the matrix $\mathbf{A}(\mathbf{x})$ is unknown (and expensive to compute). Note for a given diagonal matrix $\mathbf{D}(\mathbf{x})$, the computational cost of the inversion $\mathbf{D}(\mathbf{x})^{-1}$ is of the order $\mathcal{O}(p)$.

Consider $\mathbf{x}_k$, the variable at step k. The associated inverse Hessian $\nabla^2 f(\mathbf{x}_k)^{-1}$ can be written as the sum of the diagonal matrix $\mathbf{D}_k^{-1} = \mathbf{D}(\mathbf{x}_k)^{-1}$ and the unknown matrix $\mathbf{A}(\mathbf{x}_k)$. If we define $\mathbf{A}_k$ as the approximation of the matrix $\mathbf{A}(\mathbf{x}_k)$, then we can define the Hessian inverse approximation matrix $\mathbf{B}_k^{-1}$ as

$$\mathbf{B}_k^{-1} = \mathbf{D}_k^{-1} + \mathbf{A}_k. \tag{5}$$

Note that the matrix $\mathbf{A}_k$ might have negative eigenvalues which may consequently lead to negative eigenvalues for the Hessian inverse approximation matrix $\mathbf{B}_k^{-1}$. This phenomenon may cause a major issue, since the vector $\mathbf{B}_k^{-1}\nabla f(\mathbf{x}_k)$ might not be a descent direction. The structured quasi-Newton methods in [17, 18] suffer from this issue, and for this reason they are not globally convergent [21]. To resolve this, one might suggest to evaluate the eigenvalues of the matrix $\mathbf{D}_k^{-1} + \mathbf{A}_k$ and check if they are all positive. However, obtaining the eigenvalues can be computationally expensive. Instead, we directly check the inner product $\nabla f(\mathbf{x}_k)^T(\mathbf{D}_k^{-1} + \mathbf{A}_k)\nabla f(\mathbf{x}_k)$. If this inner product is sufficiently larger than 0, we obtain that the direction $\mathbf{d}_k = -(\mathbf{D}_k^{-1} + \mathbf{A}_k)\nabla f(\mathbf{x}_k)$ is a valid descent direction and we can proceed. In particular, we check if

$$\frac{\nabla f(\mathbf{x}_k)^T(\mathbf{D}_k^{-1} + \mathbf{A}_k)\nabla f(\mathbf{x}_k)}{\|(\mathbf{D}_k^{-1} + \mathbf{A}_k)\nabla f(\mathbf{x}_k)\|^2} \geq \delta \tag{6}$$

holds, where δ can be chosen as an arbitrary small positive scalar. The condition in (6) guarantees that the direction $-(\mathbf{D}_k^{-1} + \mathbf{A}_k)\nabla f(\mathbf{x}_k)$ is a valid descent direction. Moreover, we check the ratio between the descent direction norm $\|(\mathbf{D}_k^{-1} + \mathbf{A}_k)\nabla f(\mathbf{x}_k)\|$ and the gradient norm $\|\nabla f(\mathbf{x}_k)\|$. We need to ensure that this ratio is bounded away from zero

Algorithm 1 Diagonal-Augmented BFGS (DA-BFGS)

Set $\mathbf{A}_0 = \mathbf{0}$, $k = 0$. Choose proper $0 < \beta, c_1 < 1$.
Compute $\mathbf{D}_0^{-1}$ and $\nabla f(\mathbf{x}_0)$.
while $\|\nabla f(\mathbf{x}_k)\| >$ tol **do**
  Compute descent direction $\mathbf{d}_k = -(\mathbf{D}_k^{-1} + \mathbf{A}_k)\nabla f(\mathbf{x}_k)$
  if $\frac{-\nabla f(\mathbf{x}_k)^T\mathbf{d}_k}{\|\mathbf{d}_k\|^2} < \delta$ or $\frac{\|\mathbf{d}_k\|}{\|\nabla f(\mathbf{x}_k)\|} < \delta'$
    Set $\mathbf{A}_k = \mathbf{0}$ and $\mathbf{d}_k = -\mathbf{D}_k^{-1}\nabla f(\mathbf{x}_k)$
  end
  Set stepsize $\epsilon_k = 1$.
  while $f(\mathbf{x}_k + \epsilon_k\mathbf{d}_k) > f(\mathbf{x}_k) + c_1\epsilon_k\nabla f(\mathbf{x}_k)^T\mathbf{d}_k$ **do**
    Update the stepsize $\epsilon_k \leftarrow \beta\epsilon_k$.
  end while
  Update the variable $\mathbf{x}_{k+1} = \mathbf{x}_k + \epsilon_k\mathbf{d}_k$
  Compute variable variation $\mathbf{s}_k = \mathbf{x}_{k+1} - \mathbf{x}_k$
  Compute $\nabla f(\mathbf{x}_{k+1})$ and $\mathbf{y}_k = \nabla f(\mathbf{x}_{k+1}) - \nabla f(\mathbf{x}_k)$
  Compute $\mathbf{D}_{k+1}^{-1} = \text{Diag}(\nabla^2 f(\mathbf{x}_{k+1}))^{-1}$
  Compute $\mathbf{s}_k^{\#} := \mathbf{s}_k - \mathbf{D}_{k+1}^{-1}\mathbf{y}_k$
  Compute the updated matrix $\mathbf{A}_{k+1}$ as in (11)
  Set $k \leftarrow k + 1$
end while

by checking the following condition

$$\frac{\|(\mathbf{D}_k^{-1} + \mathbf{A}_k)\nabla f(\mathbf{x}_k)\|}{\|\nabla f(\mathbf{x}_k)\|} \geq \delta'. \tag{7}$$

Note that the conditions in (6) and (7) are required to prove the global convergence of DA-BFGS method. If at least one of the conditions in (6) and (7) is not satisfied, we reset the non-structured matrix $\mathbf{A}_k = \mathbf{0}$ and use $\mathbf{B}_k^{-1} = \mathbf{D}_k^{-1}$ as the Hessian inverse approximation. Note that the vector $-\mathbf{D}_k^{-1}\nabla f(\mathbf{x}_k)$ is a valid descent direction, since the matrix $\mathbf{D}_k^{-1}$ is positive definite with bounded eigenvalues for any $\mathbf{x}$. Note that we show that the conditions in (6) and (7) are always satisfied in a local neighborhood of the optimal argument – see Proposition 1.

After computing the descent direction $\mathbf{d}_k = -\mathbf{B}_k^{-1}\nabla f(\mathbf{x}_k)$, we proceed to pick a proper choice of step-size ϵ_k which guarantees function decrements. Following the classic BFGS method, we choose the stepsize such that the new variable leads to a lower objective function value as

$$f(\mathbf{x}_k + \epsilon_k\mathbf{d}_k) \leq f(\mathbf{x}_k) + c_1\epsilon_k\nabla f(\mathbf{x}_k)^T\mathbf{d}_k, \tag{8}$$

where $0 < c_1 < 1$ is a given constant. To make sure that the condition in (8) holds, we start with the largest possible step-size $\epsilon_k = 1$ and check if the condition is satisfied. If the condition is not satisfied, we backtrack the step-size by multiplying that by a factor $\beta < 1$. Hence, the updated variable $\mathbf{x}_{k+1}$ can be computed as

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \epsilon_k\mathbf{d}_k. \tag{9}$$

To update the matrix $\mathbf{A}_k$ which is an approximation for the exact $\mathbf{A}(\mathbf{x}_k)$, we look for a matrix that satisfies

$$\mathbf{A}_{k+1}\mathbf{y}_k = \mathbf{s}_k^{\#} := \mathbf{s}_k - \mathbf{D}_{k+1}^{-1}\mathbf{y}_k, \tag{10}$$

where $\mathbf{s}_k^{\#} := \mathbf{s}_k - \mathbf{D}_{k+1}^{-1}\mathbf{y}_k$ is defined as the modified variable variation. The expression in (10) is designed such that the approximate Hessian inverse $\mathbf{B}_{k+1}^{-1} := \mathbf{D}_{k+1}^{-1} + \mathbf{A}_{k+1}$ satisfies the condition $\mathbf{B}_{k+1}^{-1}\mathbf{y}_k = \mathbf{s}_k$ which is equivalent to $\mathbf{B}_{k+1}\mathbf{s}_k = \mathbf{y}_k$. Similarly to the logic in BFGS, we require that $\mathbf{A}_{k+1}$ satisfies the condition in (10), and seek the closest matrix to the previous approximation $\mathbf{A}_k$. Based on the update of structured BFGS methods in [17], the update of $\mathbf{A}_{k+1}$ is given by

$$\mathbf{A}_{k+1} = \mathbf{A}_k + \frac{(\mathbf{s}_k^{\#} - \mathbf{A}_k\mathbf{y}_k)\mathbf{s}_k^T + \mathbf{s}_k(\mathbf{s}_k^{\#} - \mathbf{A}_k\mathbf{y}_k)^T}{\mathbf{s}_k^T\mathbf{y}_k} - \frac{\mathbf{y}_k^T(\mathbf{s}_k^{\#} - \mathbf{A}_k\mathbf{y}_k)\mathbf{s}_k\mathbf{s}_k^T}{(\mathbf{s}_k^T\mathbf{y}_k)^2}. \tag{11}$$

The steps of the proposed DA-BFGS method are summarized in Algorithm 1. In step 5 we check whether the descent direction $\mathbf{d}_k = (\mathbf{D}_k^{-1} + \mathbf{A}_k)\nabla f(\mathbf{x}_k)$, computed in Step 4, satisfies the conditions in (6) and (7) or not. If it passes the checkpoints we proceed, otherwise we reset the unstructured part $\mathbf{A}_k = \mathbf{0}$ and set $\mathbf{d}_k = \mathbf{D}_k^{-1}\nabla f(\mathbf{x}_k)$ as in Step 6. The operations in Steps 8-11 are devoted to the computation of the step-size ϵ_k. The step-size is initialized by 1. If the step-size does not satisfy (8), we backtrack the step-size by the factor $\beta < 1$. In Step 12, the new variable $\mathbf{x}_{k+1}$ is computed and is used to compute the variable variation $\mathbf{s}_k$ and gradient variation $\mathbf{y}_k$ in Steps 13 and 14, respectively. To update the non-structured matrix $\mathbf{A}_k$ in Step 17, the modified variable variation $\mathbf{s}_k^{\#}$ is computed in Step 16 which requires access to the inverse of the diagonal matrix $\mathbf{D}_k^{-1}$ evaluated in Step 15. The algorithm stops when the norm of the gradient is sufficiently small.

4. CONVERGENCE ANALYSIS

In this section we study convergence properties of the DA-BFGS method. In proving our results we assume the following conditions hold.

Assumption 1 *The objective function $f : \mathbb{R}^p \to \mathbb{R}$ is twice differentiable and strongly convex with constant $\mu > 0$. Moreover, the gradients ∇f are Lipschitz continuous with a bounded constant $L < \infty$, i.e., for all $\mathbf{x}, \hat{\mathbf{x}} \in \mathbb{R}^p$*

$$\|\nabla f(\mathbf{x}) - \nabla f(\hat{\mathbf{x}})\| \leq L\|\mathbf{x} - \hat{\mathbf{x}}\|. \tag{12}$$

It follows from Assumption 1 that the Hessian $\nabla^2 f(\mathbf{x})$ is well-defined for all $\mathbf{x} \in \mathbb{R}^p$ and the eigenvalues of the Hessian are strictly bounded below and above by the constants μ and L, respectively, i.e. $\mu\mathbf{I} \preceq \nabla^2 f(\mathbf{x}) \preceq L\mathbf{I}$. Using these bounds it can be shown that the diagonal components of the Hessian are lower and upper bounded by μ and L, respectively. Thus, we obtain that the eigenvalues of the matrix $\mathbf{D}(\mathbf{x})$, are lower and upper bounded by μ and L, respectively, i.e., $\mu\mathbf{I} \preceq \mathbf{D}(\mathbf{x}) \preceq L\mathbf{I}$, for all $\mathbf{x} \in \mathbb{R}^p$

Consider θ_k as the angle between the negative descent direction $-\mathbf{d}_k$ and the gradient $\nabla f(\mathbf{x}_k)$. In order to make sure that the descent direction of the DA-BFGS method is a proper descent direction that leads to a globally linear convergent algorithm, the $\cos(\theta_k)$ should be strictly larger than 0. In the following lemma we show that this condition is always satisfied for the descent direction of the DA-BFGS method.

Lemma 1 *Consider the DA-BFGS method introduced in Section 3. Further, recall the definition of θ_k as the angle between the negative descent direction $-\mathbf{d}_k$ and the gradient $\nabla f(\mathbf{x}_k)$. If the conditions in Assumption 1 are satisfied, then for all steps k we have*

$$\cos(\theta_k) \geq \min\left\{\delta\delta', \frac{\mu}{L}\right\}. \tag{13}$$

The result in Lemma 1 guarantees that the descent direction of DA-BFGS is a valid descent direction. In the following lemma we show that the number of backtracking steps to compute a valid step-size that satisfies the condition in (8) is bounded above.

Lemma 2 *Consider the DA-BFGS method introduced in Section 3. Further, define $\zeta := \min\{\delta, \mu\}$. If the conditions in Assumption 1 are satisfied, then the condition in* (8) *is satisfied for ϵ_k chosen from the interval*

$$\epsilon_k \in \left[0, \frac{2(1-c_1)\zeta}{L}\right]. \tag{14}$$

The results in Lemma 2 shows that the condition in (8) is satisfied for all the positive stepsize ϵ_k less than the threshold $2(1-c_1)\zeta/L$. Note that this is a lower bound and there could be cases that for step-size larger than $2(1-c_1)\zeta/L$ the condition in (8) is satisfied. We use the results from the lemmas above in order to prove global linear convergence of DA-BFGS in the following theorem.

Theorem 1 *Consider the DA-BFGS method introduced in Section 3. If the conditions in Assumption 1 are satisfied, then the sequence of objective function error $f(\mathbf{x}_k) - f(\mathbf{x}^*)$ converges linearly to null. In other words, there exits a constant $0 < \rho < 1$ such that for all steps k,*

$$f(\mathbf{x}_{k+1}) - f(\mathbf{x}^*) \leq \rho(f(\mathbf{x}_k) - f(\mathbf{x}^*)). \tag{15}$$

Theorem 1 shows global convergence of DA-BFGS at a linear rate. Now we proceed to prove super-linear convergence of DA-BFGS in a local-neighborhood of the optimal solution.

The analysis in [17] for general structured BFGS shows that these methods converge super-linearly when the variable $\mathbf{x}_k$ is in a local neighborhood of the optimal argument $\mathbf{x}^*$. However, to guarantee that the required conditions are satisfied we first require the following assumption.

Assumption 2 *The objective function Hessian $\nabla^2 f$ is Lipschitz continuous with a bounded constant $L' < \infty$, i.e., for all $\mathbf{x}, \hat{\mathbf{x}} \in \mathbb{R}^p$ we have*

$$\|\nabla^2 f(\mathbf{x}) - \nabla^2 f(\hat{\mathbf{x}})\| \leq L'\|\mathbf{x} - \hat{\mathbf{x}}\|. \tag{16}$$

Note that Assumption 2 is commonly made to prove quadratic convergence of Newton's method [11] and superlinear convergence of quasi-Newton methods [15, 16, 22].

To show the super-linear convergence of DA-BFGS we use Theorem 3.2 in [17]. This result holds for a large class of structured quasi-Newton methods which DA-BFGS hold in this class if we drop the conditions in (6) and (7) from the update of DA-BFGS.

Theorem 2 *[Theorem 3.2 in [17]] Consider the DA-BFGS method proposed in Section 3. Suppose that the conditions in Assumptions 1 and 2 are satisfied. Moreover, assume that the inequalities in* (6) *and* (7) *hold. If the sequence of variables $\mathbf{x}_k$ is convergent to $\mathbf{x}^*$, then there exist positive constants $\hat{\epsilon}$ and $\bar{\epsilon}$ s.t. for $\mathbf{x}_{k_0}, \mathbf{A}_{k_0}$ satisfying $\|\mathbf{x}_{k_0} - \mathbf{x}^*\| \leq \hat{\epsilon}$ and $\|\mathbf{A}_{k_0} - \mathbf{A}^*\| \leq \bar{\epsilon}$ where k_0 is a positive integer, the sequence of variables $\mathbf{x}_k$ generated by DA-BFGS is q-superlinearly convergent to $\mathbf{x}^*$.*

The result in Theorem 2 indicates that if the variable $\mathbf{x}_k$ is close to the optimal argument and the approximation matrix $\mathbf{A}_k$ is in a neighborhood of the optimal matrix $\mathbf{A}^*$, then the convergence rate is superlinear. However, note that the result in Theorem 2 holds for the case that there is no condition on the descent direction $-(\mathbf{D}_k^{-1} + \mathbf{A}_k)\nabla f(\mathbf{x}_k)$ and we never reset the matrix $\mathbf{A}_k$. In the following proposition we show that if the iterates are in a local neighborhood of the optimal argument such that $\|\mathbf{x}_{k_0} - \mathbf{x}^*\| \leq \hat{\epsilon}$ and $\|\mathbf{A}_{k_0} - \mathbf{A}^*\| \leq \bar{\epsilon}$, then the conditions in (6) and (7) are satisfied in this local neighborhood of the optimal solution.

Proposition 1 *Consider the DA-BFGS method introduced in Section 3. Suppose that the iterates are in a local neighborhood of the optimal argument such that $\|\mathbf{x}_{k_0} - \mathbf{x}^*\| \leq \hat{\epsilon}$ and $\|\mathbf{A}_{k_0} - \mathbf{A}^*\| \leq \bar{\epsilon}$ for some k_0. Then the inequalities in* (6) *and* (7) *hold true if the constants δ and δ' are chosen such that*

$$\delta < \left(\frac{1}{L} - \frac{L'}{\mu^2}\hat{\epsilon} - \bar{\epsilon}\right)^2, \quad \delta' < \frac{1}{L} - \frac{L'}{\mu^2}\hat{\epsilon} - \bar{\epsilon}. \tag{17}$$

The result in Proposition 1 shows that if the constant δ and δ' satisfy the conditions in (17), then in the local neighborhood of the optimal argument $\mathbf{x}^*$ characterized by $\|\mathbf{x}_k - \mathbf{x}^*\| \leq \hat{\epsilon}$ and $\|\mathbf{A}_k - \mathbf{A}^*\| \leq \bar{\epsilon}$, the inequalities in (6) and (7) are always satisfied. Thus, the super-linear convergence of DA-BFGS follows from Theorem 2.

5. APPLICATIONS

In this section, we study the performance of DA-BFGS in two different applications. First, we consider a logistic regression problem, then we apply DA-BFGS to factorization machines.

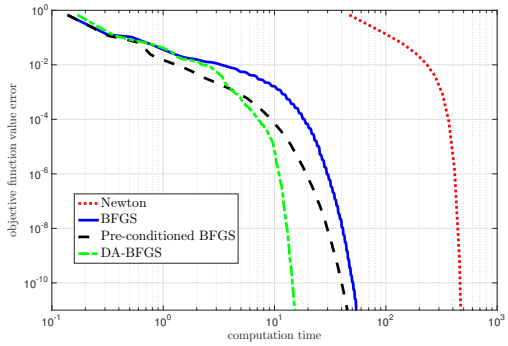

Fig. 1: Objective function value error $f(\mathbf{x}_k) - f(\mathbf{x}^*)$ versus computation time (sec). DA-BFGS outperforms other algorithms.

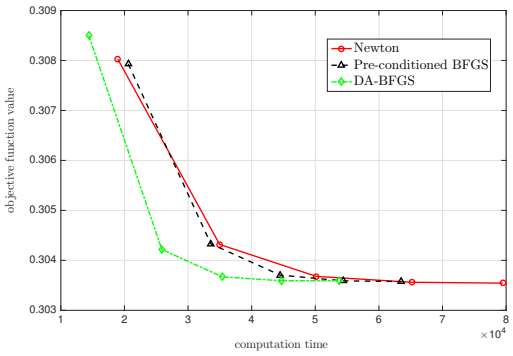

Fig. 2: Learning FM models (objective function value vs. time (m.s.)). DA-BFGS converges faster than other methods.

5.1. Logistic Regression

Consider the logistic regression (LR) problem where N samples $\{\mathbf{u}_i\}_{i=1}^N$ and their corresponding labels $\{l_i\}_{i=1}^N$ are given. The samples have dimension p, i.e., $\mathbf{u}_i \in \mathbb{R}^p$, and the labels l_i are either -1 or 1. The goal is to find the optimal classifier $\mathbf{x}^* \in \mathbb{R}^p$ that minimizes the regularized logistic loss which is given by

$$\min_{\mathbf{x}} f(\mathbf{x}) := \frac{1}{N}\sum_{i=1}^{N} \log\left(1+\exp\left(-l_i \mathbf{x}^T \mathbf{u}_i\right)\right) + \frac{\lambda}{2}\|\mathbf{x}\|^2. \quad (18)$$

Although the computation of the Hessian is not affordable, the diagonal components of the Hessian can be evaluated in an efficient way that has the total complexity of $\mathcal{O}(Np)$. To be more precise, if we define $\odot$ as an operation that computes component-wise product of two vectors and $\mathrm{Diag}(\cdot)$ as an operation that takes a vector and returns a diagonal matrix with the same entries, the matrix $\mathbf{D}(\mathbf{x})$ can be evaluated as

$$\mathbf{D}(\mathbf{x}) = \mathrm{Diag}\left(\frac{1}{N}\sum_{i=1}^{N} \frac{\mathbf{u}_i \odot \mathbf{u}_i \exp(l_i \mathbf{x}^T \mathbf{u}_i)}{(1+\exp(l_i \mathbf{x}^T \mathbf{u}_i))^2}\right) + \lambda \mathbf{I}. \quad (19)$$

The expression $\exp(l_i \mathbf{x}^T \mathbf{u}_i)$ which is required for the computation of $\mathbf{D}(\mathbf{x})$ has already been computed for the gradient evaluation. Hence, the only extra computation that the expression in (19) requires is computing the sum of N vectors with dimension p which requires $\mathcal{O}(Np)$ operations. Further, note that $\mathbf{D}(\mathbf{x}_k) = \mathbf{D}_k$ is diagonal, and the computation of its inverse has the computational complexity of the order $\mathcal{O}(p)$. Thus, the overall computation cost of DA-BFGS stays at the order $\mathcal{O}(Np)$.

For the problem in (18) we use the MNIST dataset [23]. We assign labels $l_i = 1$ and $l_i = -1$ to the samples corresponding to digits 8 and 0, respectively. We get a total of $11,774$ training examples, each of dimension 784. We compare DA-BFGS with three other algorithms: The first one is the BFGS method introduced in Section 2, initialized with the identity matrix. The second considered method is the same BFGS, initialized with a diagonal matrix of the true Hessian matrix $\mathbf{D}(\mathbf{x}_0)^{-1}$ (termed pre-conditioned BFGS). The third method is Newton's method.

The performance of these methods is compared in Fig. 1. The convergence paths in Fig. 1 showcase that Newton's method is almost impractical. Note that the dimension of the problem is $p = 784$ and if we increase the dimension p the gap between Newton's method and QN methods becomes more substantial. In addition, the pre-conditioned BFGS method (which uses partial Hessian information only for the first iteration) has a faster convergence rate relative to BFGS. Interestingly, the DA-BGFS method, which uses the partial Hessian information at *each iteration*, outperforms both BFGS and pre-conditioned BFGS.

5.2. Factorization Machines

Factorization machines (FM) extends generalized linear models [24]. In the FM model, for a feature vector $\mathbf{x} \in \mathbb{R}^n$, the predicted score $\widehat{y}(\mathbf{x})$ is

$$\widehat{y}(\mathbf{x}) = w_0 + \mathbf{w}^T \mathbf{x} + \sum_{i=1}^{n}\sum_{j=i+1}^{n} x_i x_j \mathbf{v}_i^T \mathbf{v}_j \quad (20)$$

where $w_0 \in \mathbb{R}$ is the global bias, $\mathbf{w} \in \mathcal{R}^n$ corresponds to the linear part of the model (also called feature biases), and the vectors $\mathbf{v}_i \in \mathbb{R}^k$ form the second-order part of the model. k is an important hyperparameter that controls the complexity of the interaction between every two features. Learning a FM model, then, entails learning the variables $\{w_0, \mathbf{w}, \{\mathbf{v}_i\}_{i=1}^n\}$. It is easy to see that the FM model equation is non-convex, a fact that makes learning FM a difficult task. Such models are learned either by alternating minimization or Markov-chain Monte Carlo (MCMC) [25]. We follow the alternating minimization approach.

For simplicity, we focus on learning only the latent vectors $\{\mathbf{v}_i\}$. We learn these vectors one after the other, every time regarding all the others as fixed. The algorithm shall make multiple passes over all features (termed "outer iterations"), until convergence is achieved. If we fix all the vectors except $\mathbf{v}_i$, we can write the FM prediction as

$$\widehat{y}(\mathbf{x}) = \theta_i(\mathbf{x}) + \mathbf{h}_i(\mathbf{x})^T \mathbf{v}_i, \quad (21)$$

where $\theta_i(\mathbf{x})$ is a scalar and $\mathbf{h}_i(\mathbf{x})$ is a vector, both independent of $\mathbf{v}_i$ (but dependent on $\mathbf{v}_j$ for $j \neq i$). The conclusion is that by fixing all the latent vectors but one, the FM model reduces to a linear model, which we have already discussed in the LR example. The value of k is typically small (note that k corresponds to the dimension of the optimization variable, denoted by p in the previous sections). Thus it is feasible to store the Hessian approximation matrix (100×100), but it might be expensive to invert it, or to compute the exact Hessian. This motivates the usage of BFGS-type methods, and, in particular, the proposed DA-BFGS method.

We use the `a9a` dataset [26], which has 123 features and 32,561 samples. We set consider $k = 64$ (the performance improvements were larger for larger values of k, but they have led to over-fitting). We compare Newton's method, pre-conditioned BFGS, and DA-BFGS. Fig. 2 shows the efficiency of DA-BFGS which converges after 35 seconds, compared to 45-50 seconds for the other methods. We are working on larger datasets, which should further showcase the impact of DA-BFGS.

6. CONCLUSIONS

In this paper we proposed DA-BFGS: a globally convergent structured BFGS method that incorporates the information on the diagonal components of objective function Hessian during the Hessian inverse approximation. DA-BFGS has a global linear convergence rate and a local superlinear convergence rate. Moreover, numerical results confirm that it outperforms BFGS, pre-conditioned BFGS, and Newton's method.

7. REFERENCES

[1] L. Bottou and Y. L. Cun, "On-line learning for very large datasets," in *Applied Stochastic Models in Business and Industry*, vol. 21. pp. 137-151, 2005.

[2] L. Bottou, "Large-scale machine learning with stochastic gradient descent," *In Proceedings of COMPSTAT'2010*, pp. 177–186, Physica-Verlag HD, 2010.

[3] S. Shalev-Shwartz and N. Srebro, "SVM optimization: inverse dependence on training set size," in *In Proceedings of the 25th international conference on Machine learning*. pp. 928-935, ACM 2008.

[4] F. Bullo, J. Cortés, and S. Martinez, *Distributed control of robotic networks: a mathematical approach to motion coordination algorithms*. Princeton University Press, 2009.

[5] Y. Cao, W. Yu, W. Ren, and G. Chen, "An overview of recent progress in the study of distributed multi-agent coordination," *IEEE Trans. on Industrial Informatics*, vol. 9, pp. 427–438, 2013.

[6] C. G. Lopes and A. H. Sayed, "Diffusion least-mean squares over adaptive networks: Formulation and performance analysis," *Signal Processing, IEEE Trans. on*, vol. 56, no. 7, pp. 3122–3136, 2008.

[7] I. D. Schizas, A. Ribeiro, and G. B. Giannakis, "Consensus in ad hoc WSNs with noisy links-part i: Distributed estimation of deterministic signals," *Signal Processing, IEEE Trans. on*, vol. 56, no. 1, pp. 350–364, 2008.

[8] A. Ribeiro, "Ergodic stochastic optimization algorithms for wireless communication and networking," *Signal Processing, IEEE Trans. on*, vol. 58, no. 12, pp. 6369–6386, 2010.

[9] ——, "Optimal resource allocation in wireless communication and networking," *EURASIP Journal on Wireless Communications and Networking*, vol. 2012, no. 1, pp. 1–19, 2012.

[10] Y. Nesterov, *Introductory lectures on convex optimization: A basic course*. Springer Science & Business Media, 2013, vol. 87.

[11] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

[12] Y. Nesterov, "A method of solving a convex programming problem with convergence rate $\mathcal{O}(1/k^2)$," in *Soviet Mathematics Doklady*, vol. 27, no. 2, 1983, pp. 372–376.

[13] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM journal on imaging sciences*, vol. 2, no. 1, pp. 183–202, 2009.

[14] J. Nocedal and S. Wright, *Numerical optimization*. Springer Science & Business Media, 2006.

[15] C. G. Broyden, J. E. D. Jr., Wang, and J. J. More, "On the local and superlinear convergence of quasi-Newton methods," *IMA J. Appl. Math*, vol. 12, no. 3, pp. 223–245, June 1973.

[16] J. J. E. Dennis and J. J. More, "A characterization of super linear convergence and its application to quasi-Newton methods," *Mathematics of computation*, vol. 28, no. 126, pp. 549–560, 1974.

[17] J. E. Dennis Jr, H. J. Martinez, and R. A. Tapia, "Convergence theory for the structured BFGS secant method with an application to nonlinear least squares," *Journal of Optimization Theory and Applications*, vol. 61, no. 2, pp. 161–178, 1989.

[18] L. Chen, N. Deng, and J. Zhang, "A modified quasi-Newton method for structured optimization with partial information on the Hessian," *Computational Optimization and Applications*, vol. 35, no. 1, pp. 5–18, 2006.

[19] A. Mokhtari and A. Ingber, "An improved quasi-Newton algorithm with application to factorization machines," *Technical Report*, 2016. [Online]. Available: http://www.seas.upenn.edu/~aryanm/wiki/ImprovedQuasiNewton.pdf

[20] R. H. Byrd, J. Nocedal, and Y.-X. Yuan, "Global convergence of a class of quasi-Newton methods on convex problems," *SIAM Journal on Numerical Analysis*, vol. 24, no. 5, pp. 1171–1190, 1987.

[21] W. Zhou and X. Chen, "Global convergence of a new hybrid Gauss-Newton structured BFGS method for nonlinear least squares problems," *SIAM Journal on optimization*, vol. 20, no. 5, pp. 2422–2441, 2010.

[22] M. J. D. Powell, *Some global convergence properties of a variable metric algorithm for minimization without exact line search*, 2nd ed. London, UK: Academic Press, 1971.

[23] Y. LeCun, C. Cortes, and C. J. Burges, "The mnist database of handwritten digits," 1998.

[24] S. Rendle, "Factorization machines," in *2010 IEEE International Conference on Data Mining*. IEEE, 2010, pp. 995–1000.

[25] ——, "Factorization machines with libfm," *ACM Trans. Intell. Syst. Technol.*, vol. 3, no. 3, pp. 57:1–57:22, May 2012. [Online]. Available: http://doi.acm.org/10.1145/2168752.2168771

[26] "`a9a` dataset," URL: https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/binary.html.