Stochastic Second-Order Method for Large-Scale Nonconvex Sparse Learning Models

Hongchang Gao, Heng Huang
Department of Electrical and Computer Engineering, University of Pittsburgh, USA
hongchanggao@gmail.com, heng.huang@pitt.edu

Abstract

Sparse learning models have shown promising performance in the high dimensional machine learning applications. The main challenge of sparse learning models is how to optimize it efficiently. Most existing methods solve this problem by relaxing it as a convex problem, incurring large estimation bias. Thus, the sparse learning model with nonconvex constraint has attracted much attention due to its better performance. But it is difficult to optimize due to the non-convexity. In this paper, we propose a linearly convergent stochastic second-order method to optimize this nonconvex problem for large-scale datasets. The proposed method incorporates the second-order information to improve the convergence speed. Theoretical analysis shows that our proposed method enjoys linear convergence rate and guarantees to converge to the underlying true model parameter. Experimental results have verified the efficiency and correctness of our proposed method.

1 Introduction

Sparse learning models, which play an important role in high dimensional machine learning applications [Lee *et al.*, 2006; Gao *et al.*, 2015; 2017], have attracted much attention in the past decade. Specifically, it assumes that only a few number of model parameters are responsible for the response. Thus, a straightforward way is to enforce sparsity on the model parameter by the ℓ_0-norm constraint, which restricts the number of non-zero entries in the model parameter. Due to the nonconvexity of ℓ_0-norm, most existing works [Tibshirani, 1996; Van de Geer, 2008; Yuan and Lin, 2006; Friedman *et al.*, 2008; Banerjee *et al.*, 2008] employ the relaxed ℓ_1-norm regularization to enforce the sparsity of model parameters, since it is easy to solve due to the convexity of ℓ_1-norm. For instance, the well-known Lasso [Tibshirani, 1996] solves a regression term to fit the data and an ℓ_1-norm regularization term to pursue a sparse model parameter. However, such a convex relaxation usually degenerates the performance of the model. Thus, it is necessary and challenging to solve the ℓ_0-norm constraint problem directly.

Formally, in this paper, we focus on the following sparsity-constrained optimization problem:

$$\begin{aligned} &\min_{\mathbf{w}} \mathcal{F}(\mathbf{w}) = \frac{1}{n}\sum_{i=1}^{n} f_i(\mathbf{w}) \\ &s.t. \|\mathbf{w}\|_0 \leq s \ , \end{aligned} \tag{1}$$

where $\mathcal{F}(\mathbf{w})$ is a smooth and convex function, which measures how well the model fits the input space. $\|\mathbf{w}\|_0 \leq s$ denotes the number of nonzero entries in $\mathbf{w}$ is not more than s, controlling the sparsity level of the model parameter. This model is very common in machine learning area. A representative case is the sparse linear regression problem, which is shown as follows:

$$\begin{aligned} &\min_{\mathbf{w}} \mathcal{F}(\mathbf{w}) = \frac{1}{n}\sum_{i=1}^{n} (y_i - \mathbf{x}_i^T\mathbf{w})^2 \\ &s.t. \|\mathbf{w}\|_0 \leq s \ , \end{aligned} \tag{2}$$

where $\mathbf{y} = [y_1, \cdots, y_n]^T \in \mathbb{R}^n$ is the response vector, $X = \{\mathbf{x}_1, \cdots, \mathbf{x}_n\} \in \mathbb{R}^{n\times d}$ is the design matrix, and $\mathbf{w} \in \mathbb{R}^d$ is the model parameter.

The challenge to solve Eq. (1) is the nonconvex sparse constraint, which makes Eq. (1) as an NP-hard problem. In the past few decades, a large family of algorithms [Mallat and Zhang, 1993; Needell and Tropp, 2009; Tropp and Gilbert, 2007; Zhang, 2011] have been proposed to solve Eq. (1). Among them, there has been much progress towards the gradient-based method, such as gradient hard thresholding pursuit (GraHTP) [Yuan *et al.*, 2014], iterative hard thresholding (IHT) [Blumensath and Davies, 2009], and so on. In particular, the gradient-based method updates the model parameter with the gradient descent method followed by Hard-Thresholding. However, with the development of large-scale data in recent years, these algorithms fail to handle large-scale datasets. The reason is that they need to compute the gradient with respect to all data points at each iteration, making it prohibitive for large-scale datasets. To address this problem, some researchers resort to stochastic algorithms to solve Eq. (1) in recent years, such as SGHT [Nguyen *et al.*, 2014], SVR-GHT [Li *et al.*, 2016], ASBCDHT [Chen and Gu, 2016], and so on. Unlike the full gradient descent method whose complexity of each iteration is $O(nd)$, stochastic methods have only $O(d)$ complexity in each iteration so that they are efficient for large-scale datasets.

Although the gradient-based method has achieved good performance when solving Eq. (1), yet it only considers the first-order information of the objective function, ignoring the second-order curvature information. As a result, its performance is far from satisfactory in some cases. For instance, when the condition number of the objective function in Eq. (1) is extremely large, the first-order method will converge very slowly. If incorporating the second-order curvature information of the objective function into the first-order gradient method, we can obtain a better searching direction in each iteration, making it converge fast. Thus, it is important to employ the second-order method to solve Eq. (1). In optimization community, there has been much progress towards the second-order method, such as [Byrd *et al.*, 2016; Gower *et al.*, 2016; Moritz *et al.*, 2016; Zhao *et al.*, 2017]. But most of them just focus on the convex problem. In [Yuan and Liu, 2014], a Newton greedy pursuit method was proposed to solve the nonconvex Eq. (1). However, it is not suitable for large-scale problems since it uses all data points in each iteration. Recently, some stochastic second-order methods have been proposed, such as [Moritz *et al.*, 2016; Gower *et al.*, 2016]. Although these algorithms enjoy a good converging property, the convergence analysis is based on the strongly convex condition, which is not applicable for the nonconvex Eq. (1). Thus, employing the second-order method to solve Eq. (1) is necessary and challenging.

To incorporate the second-order curvature information and address the scalability problem, we propose a stochastic L-BFGS method to solve the large-scale problem in Eq. (1). In particular, at each iteration, we first randomly sample a mini-batch of data points to evaluate the approximated inverse Hessian matrix and the gradient, updating the model parameter without considering the sparsity constraint and then performing Hard-Thresholding on the updated model parameter to get the sparse one. Additionally, due to the stochastic sampling, the introduced variance will slow down the convergence rate. To address this problem, we incorporate the variance reduction technique as [Moritz *et al.*, 2016]. One of the most important contributions of this paper is that we prove the linear convergence rate of the second-order stochastic L-BFGS for solving the nonconvex Eq. (1). As far as we know, this is the first work which proves stochastic L-BFGS has a linear convergence rate for the nonconvex problem with the sparsity constraint. Furthermore, the output estimator from our proposed method is guaranteed to converge to the unknown true model parameter, which is also the first work showing such a result for stochastic L-BFGS. The experiments on both synthetic and real-world datasets have shown the correctness and effectiveness of our proposed method.

Notation Throughout this paper, the matrix is represented by the uppercase letter, the vector is denoted by the bold lowercase letter, and the scalar is represented by by the unbold lowercase letter. In particular, $X = \{\mathbf{x}_1, \cdots, \mathbf{x}_n\} \in \mathbb{R}^{n \times d}$ denotes the design matrix, $\mathbf{y} = [y_1, \cdots, y_n]^T \in \mathbb{R}^n$ denotes the response vector, and $\mathbf{w} \in \mathbb{R}^d$ denotes the model parameter. For the vector $\mathbf{w} \in \mathbb{R}^d$, we define $\|\mathbf{w}\|_1 = \sum_{i=1}^d |w_i|$, $\|\mathbf{w}\|_2 = \sum_{i=1}^n w_i^2$, $\|\mathbf{w}\|_\infty = \max_i |w_i|$. Additionally, $supp(\mathbf{w})$ denotes the index of nonzero elements in $\mathbf{w}$, and $supp(\mathbf{w}, s)$ denotes the index of the top s elements of $\mathbf{w}$ in regard to magnitude. $\mathbf{w}^{(t)}$ denotes the vector in the t-th iteration.

2 Stochastic L-BFGS for Large-Scale Nonconvex Sparse Learning Models

In this section, we will present the detail of our proposed method for large-scale nonconvex sparse learning models.

The core idea is employing stochastic L-BFGS to solve Eq. (1). However, the naive stochastic L-BFGS converges slowly due to the introduced variance by random sampling. Inspired by [Moritz *et al.*, 2016], we employ the variance reduction technique [Johnson and Zhang, 2013] to accelerate it. Meanwhile, unlike the traditional stochastic L-BFGS [Moritz *et al.*, 2016] which is only applicable for strongly convex problems, our method can successfully handle the nonconvex problem with the sparsity constraint. The details of our proposed method are summarized in Algorithm 1.

In Algorithm 1, there are two nested loops. In the outer loop, we calculate the full gradient $\tilde{\boldsymbol{\mu}}$ in Line 8 such that we can use it to reduce the variance of the stochastic gradient. In the inner loop, our algorithm combines the variance reduced gradient $\mathbf{v}^{(t)}$ and the approximated inverse Hessian matrix $H^{(r)}$ to update the model parameter. More specifically, the gradient

$$\mathbf{v}^{(t)} = \nabla f_{\mathcal{B}}(\mathbf{w}^{(t)}) - \nabla f_{\mathcal{B}}(\tilde{\mathbf{w}}) + \tilde{\boldsymbol{\mu}} \tag{3}$$

is an unbiased estimation to $\nabla \mathcal{F}(\mathbf{w}^{(t)}) = \frac{1}{n}\sum_{i=1}^n \nabla f_i(\mathbf{w}^{(t)})$, where $\nabla f_{\mathcal{B}}(\mathbf{w}^{(t)}) = \frac{1}{|\mathcal{B}|}\sum_{i \in \mathcal{B}} \nabla f_i(\mathbf{w}^{(t)})$. After that, we update the model parameter without considering the sparsity constraint as follows:

$$\bar{\mathbf{w}}^{(t+1)} = \mathbf{w}^{(t)} - \eta H^{(r)} \mathbf{v}^{(t)} , \tag{4}$$

where η is the step size, and $\bar{\mathbf{w}}^{(t+1)}$ is the temporary model parameter. With this updating rule, the second-order curvature is incorporated by $H^{(r)}$. Thus, it will converge faster than the first order approach. In the following, the Hard-Thresholding is performed on $\bar{\mathbf{w}}^{(t+1)}$ to obtain the solution satisfying the sparsity constraint as follows:

$$\mathbf{w}^{(t+1)} = \mathcal{H}(\bar{\mathbf{w}}^{(t+1)}, s) , \tag{5}$$

where s is the sparsity level in Eq. (1), and the Hard-Thresholding operator $\mathcal{H}(\cdot, \cdot)$ is defined as follows:

$$[\mathcal{H}(\mathbf{w}, s)]_i = \begin{cases} w_i, & i \in supp(\mathbf{w}, s) \\ 0, & otherwise \end{cases} , \tag{6}$$

where $supp(\mathbf{w}, s)$ denotes the s largest non-zero values of the model parameter $\mathbf{w}$.

In Line 11-19 of Algorithm 1, after every L iterations, we update the approximated inverse Hessian matrix $H^{(r)}$ by the L-BFGS schema as follows:

$$\begin{aligned} H_i^{(r)} = & (\mathbf{I} - \rho^{(i)}\mathbf{s}^{(i)}\mathbf{y}^{(i)T})^T H_{i-1}^{(r)} (\mathbf{I} - \rho^{(i)}\mathbf{s}^{(i)}\mathbf{y}^{(i)T}) \\ & + \rho^{(i)}\mathbf{s}^{(i)}\mathbf{s}^{(i)T} , \end{aligned} \tag{7}$$

where $r - M + 1 \leq i \leq r$, M is the memory size, $\rho^{(i)} = \frac{1}{\mathbf{s}^{(i)T}\mathbf{y}^{(i)}}$, and $H_{r-M}^{(r)} = \frac{\mathbf{s}^{(r)T}\mathbf{y}^{(r)}}{\mathbf{y}^{(r)T}\mathbf{y}^{(r)}}\mathbf{I}$. Then, we set

$H^{(r)} = H_r^{(r)}$. Note that unlike traditional L-BFGS method, we update $\mathbf{y}^{(r)} = \nabla^2 f_{\mathcal{B}'}(\theta^{(r)})\mathbf{s}^{(r)}$ since it works better in the stochastic setting [Moritz *et al.*, 2016], where $\nabla^2 f_{\mathcal{B}'}(\boldsymbol{\theta}^{(r)}) = \frac{1}{|\mathcal{B}'|}\sum_{i\in\mathcal{B}'}\nabla^2 f_i(\boldsymbol{\theta}^{(r)})$.

Practical Acceleration In Algorithm 1, we need to compute the Hessian matrix $\nabla^2 f_{\mathcal{B}'}(\boldsymbol{\theta}^{(r)})$ and its inverse approximation $H^{(r)}$. Both of them require $O(d^2)$ storage, which is prohibitive for high dimensionality problems. Instead of constructing $H^{(r)}$ explicitly, we employ the two-loop recursion method [Nocedal and Wright, 2006] to directly compute $H^{(r)}\mathbf{v}^{(t)}$ based on the correction pairs $\{\mathbf{s}^{(i)}, \mathbf{y}^{(i)}\}_{i=r-M+1}^{r}$.

For the Hessian matrix, we assume it can be represented as follows:

$$\nabla^2 f_{\mathcal{B}'}(\boldsymbol{\theta}^{(r)}) = \frac{1}{|\mathcal{B}'|}\sum_{i\in\mathcal{B}'} A_i(\boldsymbol{\theta}^{(r)})A_i^T(\boldsymbol{\theta}^{(r)})\,. \tag{8}$$

Actually, it is very common in many machine learning problems. For example, $A_i(\boldsymbol{\theta}^{(r)})$ in Eq. (2) is $\mathbf{x}_i$ so that $\nabla^2 f_{\mathcal{B}'}(\boldsymbol{\theta}^{(r)}) = \frac{1}{|\mathcal{B}'|}\sum_{i\in\mathcal{B}'}\mathbf{x}_i\mathbf{x}_i^T$. Based on this representation, instead of computing $\nabla^2 f_{\mathcal{B}'}(\boldsymbol{\theta}^{(r)})$ explicitly, we can directly compute $\mathbf{y}^{(r)}$ as follows:

$$\mathbf{y}^{(r)} = \frac{1}{|\mathcal{B}'|}\sum_{i\in\mathcal{B}'} A_i(\boldsymbol{\theta}^{(r)})[A_i^T(\boldsymbol{\theta}^{(r)})\mathbf{s}^{(r)}]\,, \tag{9}$$

which will save much storage and computation since no explicit Hessian matrix needs to store.

Algorithm 1 Stochastic L-BFGS Algorithm for Solving Eq. (1).

```
Input: $X \in \mathbb{R}^{n\times d}$, $Y \in \mathbb{R}^n$, $s > 0$.
Output: $\mathbf{w} \in \mathbb{R}^d$
Initialize $r = 0$, $H_0 = I$
for $k = 0, 1, 2, \cdots$ do
    $\tilde{\mathbf{w}} = \tilde{\mathbf{w}}^{(k-1)}$, $\mathbf{w}^{(0)} = \tilde{\mathbf{w}}^{(k-1)}$
    $\tilde{\boldsymbol{\mu}} = \frac{1}{n}\sum_{i=1}^{n}\nabla f_i(\tilde{\mathbf{w}})$
    for $t = 0, 1, 2, \cdots, m-1$ do
        Randomly sample a subset $\mathcal{B}$ from $\{1, 2, \cdots, n\}$
        Compute gradient $\mathbf{v}^{(t)} = \nabla f_{\mathcal{B}}(\mathbf{w}^{(t)}) - \nabla f_{\mathcal{B}}(\tilde{\mathbf{w}}) + \tilde{\boldsymbol{\mu}}$
        Update $\bar{\mathbf{w}}^{(t+1)} = \mathbf{w}^{(t)} - \eta H^{(r)}\mathbf{v}^{(t)}$
        Hard-Thresholding $\mathbf{w}^{(t+1)} = \mathcal{H}(\bar{\mathbf{w}}^{(t+1)}, s)$
        if mod($t$, $L$)=0 then
            $r = r + 1$
            $\boldsymbol{\theta}^{(r)} = \frac{1}{L}\sum_{j=t-L}^{t-1}\mathbf{w}^{(j)}$
            Randomly sample a subset $\mathcal{B}'$ from $\{1, 2, \cdots, n\}$
            Compute $\nabla^2 f_{\mathcal{B}'}(\boldsymbol{\theta}^{(r)})$
            Compute $\mathbf{s}^{(r)} = \boldsymbol{\theta}^{(r)} - \boldsymbol{\theta}^{(r-1)}$
            Compute $\mathbf{y}^{(r)} = \nabla^2 f_{\mathcal{B}'}(\boldsymbol{\theta}^{(r)})\mathbf{s}^{(r)}$
            Compute $H^{(r)}$ with Eq. (7)
        end if
    end for
    Set $\tilde{\mathbf{w}}^{(k)}$ as the randomly selected $\mathbf{w}^{(i)}$ from $\{\mathbf{w}^{(0)}, \cdots, \mathbf{w}^{(m-1)}\}$
end for
return w
```

3 Convergence Analysis

In this section, we will present the convergence analysis about Algorithm 1. At first, we will introduce the following assumptions that our analysis depends on.

Assumption 1. *(Restricted Strong Convexity) Function $\mathcal{F}$ satisfies restricted $\lambda_{\tilde{s}}$-strong convexity condition at sparse level $\tilde{s}$. Formally, we have*

$$\mathcal{F}(\mathbf{w}) \geq \mathcal{F}(\mathbf{w}') + \nabla\mathcal{F}(\mathbf{w}')^T(\mathbf{w}-\mathbf{w}') + \frac{\lambda_{\tilde{s}}}{2}||\mathbf{w}-\mathbf{w}'||_2^2\,, \tag{10}$$

for all $\mathbf{w}$, $\mathbf{w}'$ *such that* $||\mathbf{w}-\mathbf{w}'||_0 \leq s$ *and* $\lambda_{\tilde{s}} > 0$.

Assumption 2. *(Restricted Strong Smoothness) Function f_i satisfies restricted $L_{\tilde{s}}$-strong smoothness condition at sparse level $\tilde{s}$. Formally, we have*

$$f_i(\mathbf{w}) \leq f_i(\mathbf{w}') + \nabla f_i(\mathbf{w}')^T(\mathbf{w}-\mathbf{w}') + \frac{L_{\tilde{s}}}{2}||\mathbf{w}-\mathbf{w}'||_2^2\,, \tag{11}$$

for all $\mathbf{w}$, $\mathbf{w}'$ *such that* $||\mathbf{w}-\mathbf{w}'||_0 \leq s$ *and* $L_{\tilde{s}} > 0$.

These two assumptions indicate that $\mathcal{F}(\mathbf{w})$ is strongly convex and $f_i(\mathbf{w})$ is smooth in the sparse subspace. Additionally, based on these two assumptions, we can define the restricted condition number as $\kappa_{\tilde{s}} = \frac{L_{\tilde{s}}}{\lambda_{\tilde{s}}}$.

Assumption 3. *The gradient is bounded as follows:*

$$E[||\nabla f_i(\mathbf{w})\|_2^2] \leq G^2. \tag{12}$$

In the following, we present three lemmas for proving the main theorem.

Lemma 1. *Suppose Assumption 1 and 2 satisfy with the sparsity level $\tilde{s} = 2s + s^*$. Then, for the sparse vector $\mathbf{w}^* \in \mathbb{R}^d$ such that $||\mathbf{w}^*||_0 \leq s^*$, and the sparse vector $\mathbf{w}^{(t)} \in \mathbb{R}^d$ such that $||\mathbf{w}^{(t)}||_0 \leq s$, we have*

$$\begin{aligned} E||\mathbf{v}_{\mathcal{S}}^{(t)}||_2^2 \leq\ & 12L_{\tilde{s}}[\mathcal{F}(\mathbf{w}^{(t)}) - \mathcal{F}(\mathbf{w}^*) + \mathcal{F}(\tilde{\mathbf{w}}) - \mathcal{F}(\mathbf{w}^*)] \\ & + 3||\nabla_{\mathcal{S}}\mathcal{F}(\mathbf{w}^*)||_2^2\,, \end{aligned} \tag{13}$$

where $\mathcal{S} \supseteq (supp(\mathbf{w}^*) \cup supp(\mathbf{w}^{(t)}))$.

This lemma bounds the variance of the stochastic gradient $\mathbf{v}_{\mathcal{S}}^{(t)}$. The proof can be found in Lemma 3.5 [Li *et al.*, 2016]. Thus, we do not include it due to the space limitation.

Lemma 2. *Given the sparse vector $\mathbf{w}^* \in \mathbb{R}^d$ such that $||\mathbf{w}^*||_0 \leq s^*$, for $s > s^*$ and any $\mathbf{w} \in \mathbb{R}^d$, we have*

$$\|\mathcal{H}(\mathbf{w}, s) - \mathbf{w}^*\|_2^2 \leq (1 + \frac{2\sqrt{s^*}}{\sqrt{s-s^*}})\|\mathbf{w}, \mathbf{w}^*\|_2^2\,. \tag{14}$$

This lemma actually presents the projection error bound for the Hard-Thresholding operator. The proof can be referred to Lemma 3.3 of [Li *et al.*, 2016].

Lemma 3. *If Assumption 1 and 2 hold, the estimation of inverse Hessian matrix $H^{(r)}$ (for all $r \geq 1$) is bounded by*

$$\gamma\mathbf{I} \preceq H^{(r)} \preceq \Gamma\mathbf{I}\,, \tag{15}$$

where $0 < \gamma \leq \Gamma$.

This is a common condition for stochastic L-BFGS method, and the detailed proof can be found from Lemma 4 in [Moritz *et al.*, 2016].

Based on these assumptions and lemmas, we turn to present the main result of our proposed method.

Theorem 1. *Suppose Assumption 1 and 2 satisfy with the sparsity level* $\tilde{s} = 2s + s^*$*. Denote* $\mathbf{w}^*$ *as the unknown true model parameter such that* $||\mathbf{w}^*||_0 \le s^*$ *and* $\tilde{\mathcal{S}} = supp(\mathcal{H}(\nabla \mathcal{F}(\mathbf{w}^*), 2s)) \cup supp(\mathbf{w}^*)$*. By choosing* $\frac{C_2}{\Gamma^2 L_{\tilde{s}}} \le \eta \le \frac{C_1}{\Gamma^2 L_{\tilde{s}}}$ *and* $s \ge (1 + 1/A^2)s^*$ *where* $A = \frac{C_2 - 3\kappa_{\tilde{s}}(1+\eta)\eta}{6\kappa_{\tilde{s}}(1+\eta)^2}$ *with valid constants* C_1*,* C_2*, and large* m*, such that*

$$\theta = \frac{6\eta\Gamma^2 L_{\tilde{s}}}{1 - 6\eta\Gamma^2 L_{\tilde{s}}} + \frac{\beta^m(1+\eta)(\beta - 1)}{\lambda_{\tilde{s}}\eta(1 - 6\eta\Gamma^2 L_{\tilde{s}})(\beta^m - 1)} < 1 \, ,$$

where $\beta = (1 + \frac{2\sqrt{s^*}}{\sqrt{s - s^*}})(1 + \eta)$*. Then, for all* $k > 0$*, we have*

$$E[\mathcal{F}(\tilde{\mathbf{w}}^{(k)}) - \mathcal{F}(\mathbf{w}^*)] \le \theta^k E[\mathcal{F}(\tilde{\mathbf{w}}^{(0)}) - \mathcal{F}(\mathbf{w}^*)] + \frac{3\eta\Gamma^2||\nabla_{\tilde{\mathcal{S}}}\mathcal{F}(\mathbf{w}^*)||_2^2 + \Gamma^2 G^2}{2(1 - 6\eta\Gamma^2 L_{\tilde{s}})(1 - \theta)} \, . \tag{16}$$

In the following, we present the detailed proof about Theorem 1.

Proof. Due to $\bar{\mathbf{w}}^{(t+1)} = \mathbf{w}^{(t)} - \eta H^{(r)}\mathbf{v}^{(t)}$, conditioning on $\mathbf{w}^{(t)}$, we have

$$\begin{aligned}
& E||\bar{\mathbf{w}}^{(t+1)} - \mathbf{w}^*||_2^2 \\
= & E||\mathbf{w}^{(t)} - \eta H^{(r)}\mathbf{v}^{(t)} - \mathbf{w}^*||_2^2 \\
= & E||\mathbf{w}^{(t)} - \mathbf{w}^*||_2^2 + \eta^2 E||H^{(r)}\mathbf{v}_{\mathcal{S}}^{(t)}||_2^2 \\
& + 2\eta\langle \mathbf{w}^* - \mathbf{w}^{(t)}, E[H^{(r)}\mathbf{v}_{\mathcal{S}}^{(t)}]\rangle \\
= & E||\mathbf{w}^{(t)} - \mathbf{w}^*||_2^2 + \eta^2 E||H^{(r)}\mathbf{v}_{\mathcal{S}}^{(t)}||_2^2 \\
& + 2\eta\langle \mathbf{w}^* - \mathbf{w}^{(t)}, \nabla_{\mathcal{S}}\mathcal{F}(\mathbf{w}^{(t)})\rangle \\
& + 2\eta\langle \mathbf{w}^* - \mathbf{w}^{(t)}, H^{(r)}\nabla_{\mathcal{S}}\mathcal{F}(\mathbf{w}^{(t)}) - \nabla_{\mathcal{S}}\mathcal{F}(\mathbf{w}^{(t)})\rangle \\
\le & (1+\eta)E||\mathbf{w}^{(t)} - \mathbf{w}^*||_2^2 + \eta^2\Gamma^2 E||\mathbf{v}_{\mathcal{S}}^{(t)}||_2^2 \\
& - 2\eta[\mathcal{F}(\mathbf{w}^{(t)}) - \mathcal{F}(\mathbf{w}^{(*)})] + \eta\Gamma^2||\nabla_{\mathcal{S}}\mathcal{F}(\mathbf{w}^{(t)})||_2^2 \\
\le & (1+\eta)E||\mathbf{w}^{(t)} - \mathbf{w}^*||_2^2 - 2\eta[\mathcal{F}(\mathbf{w}^{(t)}) - \mathcal{F}(\mathbf{w}^{(*)})] \\
& + 12\eta^2\Gamma^2 L_{\tilde{s}}[\mathcal{F}(\mathbf{w}^{(t)}) - \mathcal{F}(\mathbf{w}^*) + \mathcal{F}(\tilde{\mathbf{w}}) - \mathcal{F}(\mathbf{w}^*)] \\
& + 3\eta^2\Gamma^2||\nabla_{\mathcal{S}}\mathcal{F}(\mathbf{w}^*)||_2^2 + \eta\Gamma^2 G^2 \\
= & (1+\eta)E||\mathbf{w}^{(t)} - \mathbf{w}^*||_2^2 + 3\eta^2\Gamma^2||\nabla_{\mathcal{S}}\mathcal{F}(\mathbf{w}^*)||_2^2 \\
& - 2\eta(1 - 6\eta\Gamma^2 L_{\tilde{s}})[\mathcal{F}(\mathbf{w}^{(t)}) - \mathcal{F}(\mathbf{w}^*)] \\
& + 12\eta^2\Gamma^2 L_{\tilde{s}}[\mathcal{F}(\tilde{\mathbf{w}}) - \mathcal{F}(\mathbf{w}^*)] + \eta\Gamma^2 G^2 \, ,
\end{aligned} \tag{17}$$

where $\mathcal{S} = supp(\mathbf{w}^*) \cup supp(\mathbf{w}^{(t)}) \cup supp(\mathbf{w}^{(t+1)})$. The third step is due to $E[H^{(r)}\mathbf{v}_{\mathcal{S}}^{(t)}] = H^{(r)}\nabla_{\mathcal{S}}\mathcal{F}(\mathbf{w}^{(t)})$. The fourth step follows from Assumption 1, Lemma 3, and the fact $2ab \le a^2 + b^2$. The fifth step is due to Lemma 1.

According to Lemma 2, we have

$$\begin{aligned}
& E||\mathbf{w}^{(t+1)} - \mathbf{w}^*||_2^2 \\
\le & \alpha(1+\eta)E||\mathbf{w}^{(t)} - \mathbf{w}^*||_2^2 + 3\alpha\eta^2\Gamma^2||\nabla_{\mathcal{S}}\mathcal{F}(\mathbf{w}^*)||_2^2 \\
& - 2\alpha\eta(1 - 6\eta\Gamma^2 L_{\tilde{s}})[\mathcal{F}(\mathbf{w}^{(t)}) - \mathcal{F}(\mathbf{w}^*)] \\
& + 12\alpha\eta^2\Gamma^2 L_{\tilde{s}}[\mathcal{F}(\tilde{\mathbf{w}}) - \mathcal{F}(\mathbf{w}^*)] + \alpha\eta\Gamma^2 G^2 \\
\le & \beta E||\mathbf{w}^{(t)} - \mathbf{w}^*||_2^2 + \beta\frac{3\eta^2\Gamma^2}{1+\eta}||\nabla_{\mathcal{S}}\mathcal{F}(\mathbf{w}^*)||_2^2 \\
& - \beta\frac{2\eta(1 - 6\eta\Gamma^2 L_{\tilde{s}})}{1+\eta}[\mathcal{F}(\mathbf{w}^{(t)}) - \mathcal{F}(\mathbf{w}^*)] \\
& + \beta\frac{12\eta^2\Gamma^2 L_{\tilde{s}}}{1+\eta}[\mathcal{F}(\tilde{\mathbf{w}}) - \mathcal{F}(\mathbf{w}^*)] + \beta\frac{\eta\Gamma^2}{1+\eta}G^2 \, ,
\end{aligned} \tag{18}$$

where $\beta = \alpha(1+\eta)$ and $\alpha = 1 + \frac{2\sqrt{s^*}}{\sqrt{s - s^*}}$. By summing over $t = 0, \cdots, m-1$, we have

$$\begin{aligned}
& E||\mathbf{w}^{(m)} - \mathbf{w}^*||_2^2 \\
& + \frac{2\eta(1 - 6\eta\Gamma^2 L_{\tilde{s}})(\beta^m - 1)}{(1+\eta)(\beta - 1)}E[\mathcal{F}(\tilde{\mathbf{w}}^{(k)}) - \mathcal{F}(\mathbf{w}^*)] \\
\le & \beta^m E||\tilde{\mathbf{w}}^{(k-1)} - \mathbf{w}^*||_2^2 \\
& + \frac{12\eta^2\Gamma^2 L_{\tilde{s}}(\beta^m - 1)}{(1+\eta)(\beta - 1)}E[\mathcal{F}(\tilde{\mathbf{w}}^{(k-1)}) - \mathcal{F}(\mathbf{w}^*)] \\
& + \frac{3\eta^2\Gamma^2(\beta^m - 1)}{(1+\eta)(\beta - 1)}||\nabla_{\mathcal{S}}\mathcal{F}(\mathbf{w}^*)||_2^2 + \frac{\eta\Gamma^2(\beta^m - 1)}{(1+\eta)(\beta - 1)}G^2 \\
\le & \frac{2\beta^m}{\lambda_{\tilde{s}}}E[\mathcal{F}(\tilde{\mathbf{w}}^{(k-1)}) - \mathcal{F}(\mathbf{w}^*)] \\
& + \frac{12\eta^2\Gamma^2 L_{\tilde{s}}(\beta^m - 1)}{(1+\eta)(\beta - 1)}E[\mathcal{F}(\tilde{\mathbf{w}}^{(k-1)}) - \mathcal{F}(\mathbf{w}^*)] \\
& + \frac{3\eta^2\Gamma^2(\beta^m - 1)}{(1+\eta)(\beta - 1)}||\nabla_{\tilde{\mathcal{S}}}\mathcal{F}(\mathbf{w}^*)||_2^2 + \frac{\eta\Gamma^2(\beta^m - 1)}{(1+\eta)(\beta - 1)}G^2 \, .
\end{aligned} \tag{19}$$

where $\tilde{\mathbf{w}}^{(k-1)} = \mathbf{w}^{(0)} = \tilde{\mathbf{w}}$ and $\tilde{\mathcal{S}} = supp(\mathcal{H}(\nabla\mathcal{F}(\mathbf{w}^*), 2s)) \cup supp(\mathbf{w}^*)$. By rearranging the above inequality, we have

$$\begin{aligned}
& E[\mathcal{F}(\tilde{\mathbf{w}}^{(k)}) - \mathcal{F}(\mathbf{w}^*)] \\
\le & \theta E[\mathcal{F}(\tilde{\mathbf{w}}^{(k-1)}) - \mathcal{F}(\mathbf{w}^*)] + \frac{\Gamma^2}{2(1 - 6\eta\Gamma^2 L_{\tilde{s}})}G^2 \\
& + \frac{3\eta\Gamma^2}{2(1 - 6\eta\Gamma^2 L_{\tilde{s}})}||\nabla_{\tilde{\mathcal{S}}}\mathcal{F}(\mathbf{w}^*)||_2^2 \, .
\end{aligned} \tag{20}$$

Now, we need to verify that

$$\theta = \frac{6\eta\Gamma^2 L_{\tilde{s}}}{1 - 6\eta\Gamma^2 L_{\tilde{s}}} + \frac{\beta^m(1+\eta)(\beta - 1)}{\lambda_{\tilde{s}}\eta(1 - 6\eta\Gamma^2 L_{\tilde{s}})(\beta^m - 1)} < 1 \, . \tag{21}$$

At first, assume $\eta \le \frac{C_1}{\Gamma^2 L_{\tilde{s}}} < \frac{1}{18\Gamma^2 L_{\tilde{s}}}$, then

$$\frac{6\eta\Gamma^2 L_{\tilde{s}}}{1 - 6\eta\Gamma^2 L_{\tilde{s}}} < \frac{1}{2} \, . \tag{22}$$

Furthermore, assume $s \ge (1 + 1/A^2)s^*$ where $A = \frac{C_2 - 3\kappa_{\tilde{s}}(1+\eta)\eta}{6\kappa_{\tilde{s}}(1+\eta)^2}$, and $\eta \ge \frac{C_2}{\Gamma^2 L_{\tilde{s}}}$ where $C_2 \le C_1$, to guarantee

$$\frac{\beta^m(1+\eta)(\beta - 1)}{\lambda_{\tilde{s}}\eta(1 - 6\eta\Gamma^2 L_{\tilde{s}})(\beta^m - 1)} < \frac{1}{2} \, , \tag{23}$$

we get

$$\beta^m > \frac{C_2}{C_2 - 3\kappa_{\tilde{s}}(1+\eta)(\beta-1)} . \tag{24}$$

Hence, to guarantee $\theta < 1$, we should have

$$m > \frac{6\kappa_{\tilde{s}}(1+\eta)^2}{C_2 - 3\kappa_{\tilde{s}}(1+\eta)\eta} \log B \, , \tag{25}$$

where $B = \frac{C_2}{C_2 - 3\kappa_{\tilde{s}}(1+\eta)(\beta-1)}$. In this way, we have $\theta < 1$. By recursively applying Eq. (20), we can get the desired result as follows:

$$\begin{aligned} E[\mathcal{F}(\tilde{\mathbf{w}}^{(k)}) - \mathcal{F}(\mathbf{w}^*)] &\leq \theta^k E[\mathcal{F}(\tilde{\mathbf{w}}^{(0)}) - \mathcal{F}(\mathbf{w}^*)] \\ &+ \frac{3\eta\Gamma^2 ||\nabla_{\tilde{\mathcal{S}}}\mathcal{F}(\mathbf{w}^*)||_2^2 + \Gamma^2 G^2}{2(1-6\eta\Gamma^2 L_{\tilde{s}})(1-\theta)} \, , \end{aligned} \tag{26}$$

which completes the proof. □

Remark 1. *Theorem 1 indicates a linear convergence rate. In particular, to get a pre-defined accuracy $\epsilon > 0$ with respect to the function value gap, we need $O(\log(1/\epsilon))$ outer iterations. Additionally, to have linear convergence rate, m should be set sufficiently large as in Eq. (25).*

In the following, we present the approximation error bound for the estimator obtained from Algorithm 1.

Corollary 1. *With the same conditions as Theorem 1, for all $k > 0$, we have the error bound for the estimator $\tilde{\mathbf{w}}^{(k)}$ as follows:*

$$\begin{aligned} E||\tilde{\mathbf{w}}^{(k)} - \mathbf{w}^*||_2 &\leq \sqrt{\frac{2\theta^k[\mathcal{F}(\tilde{\mathbf{w}}^{(0)}) - \mathcal{F}(\mathbf{w}^*)]}{\lambda_{\tilde{s}}}} \\ &+ \sqrt{\frac{\Gamma^2}{\lambda_{\tilde{s}}(1-6\eta\Gamma^2 L_{\tilde{s}})(1-\theta))}} G \\ &+ \left(\frac{2}{\lambda_{\tilde{s}}} + \sqrt{\frac{3\eta\Gamma^2}{\lambda_{\tilde{s}}(1-6\eta\Gamma^2 L_{\tilde{s}})(1-\theta)}}\right)\sqrt{\tilde{s}}||\nabla\mathcal{F}(\mathbf{w}^*)||_\infty \, . \end{aligned} \tag{27}$$

Proof. Due to Assumption 1, we have

$$\begin{aligned} \mathcal{F}(\mathbf{w}^*) &\leq \mathcal{F}(\tilde{\mathbf{w}}^{(k)}) + \nabla\mathcal{F}(\mathbf{w}^*)^T(\mathbf{w}^* - \tilde{\mathbf{w}}^{(k)}) \\ &- \frac{\lambda_{\tilde{s}}}{2}||\mathbf{w}^* - \tilde{\mathbf{w}}^{(k)}||_2^2 \, . \end{aligned} \tag{28}$$

Furthermore, denote

$$\begin{aligned} \Sigma &= \theta^k[\mathcal{F}(\tilde{\mathbf{w}}^{(0)}) - \mathcal{F}(\mathbf{w}^*)] \\ &+ \frac{\Gamma^2 G^2 + 3\eta\Gamma^2||\nabla_{\tilde{\mathcal{S}}}\mathcal{F}(\mathbf{w}^*)||_2^2}{2(1-6\eta\Gamma^2 L_{\tilde{s}})(1-\theta)} \, , \end{aligned} \tag{29}$$

then based on Eq. (16), we have

$$\begin{aligned} E[\mathcal{F}(\tilde{\mathbf{w}}^{(k)}) - \Sigma] &\leq \mathcal{F}(\mathbf{w}^*) \leq E[\mathcal{F}(\tilde{\mathbf{w}}^{(k)}) \\ &+ \nabla\mathcal{F}(\mathbf{w}^*)^T(\mathbf{w}^* - \tilde{\mathbf{w}}^{(k)}) - \frac{\lambda_{\tilde{s}}}{2}||\mathbf{w}^* - \tilde{\mathbf{w}}^{(k)}||_2^2] \, . \end{aligned} \tag{30}$$

Furthermore, we have

$$\begin{aligned} &E[\nabla\mathcal{F}(\mathbf{w}^*)^T(\mathbf{w}^* - \tilde{\mathbf{w}}^{(k)})] \\ &\leq ||\nabla\mathcal{F}(\mathbf{w}^*)||_\infty E||\mathbf{w}^* - \tilde{\mathbf{w}}^{(k)}||_1 \\ &\leq \sqrt{\tilde{s}}||\nabla\mathcal{F}(\mathbf{w}^*)||_\infty E||\mathbf{w}^* - \tilde{\mathbf{w}}^{(k)}||_2 \, . \end{aligned} \tag{31}$$

Put Eq. (31) into Eq. (30), we have

$$\begin{aligned} &\frac{\lambda_{\tilde{s}}}{2}(E||\mathbf{w}^* - \tilde{\mathbf{w}}^{(k)}||_2)^2 \\ &\leq \sqrt{\tilde{s}}||\nabla\mathcal{F}(\mathbf{w}^*)||_\infty E||\mathbf{w}^* - \tilde{\mathbf{w}}^{(k)}||_2 + \Sigma \, . \end{aligned} \tag{32}$$

By solving this inequality with respect to $E||\mathbf{w}^* - \tilde{\mathbf{w}}^{(k)}||_2$, we can obtain the desired result. □

Remark 2. *This error bound for the estimator consists of three terms. The first term corresponds to the optimization error, the second term is a constant, and the third term corresponds to the statistical error. After sufficient iterations, the first term will approach to zero. Therefore, our algorithm can always converge to the unknown true parameter $\mathbf{w}^*$, up to the statistical error and a constant value.*

4 Experiments

In this section, we will present the performance of our proposed method on both synthetic and real-world datasets.

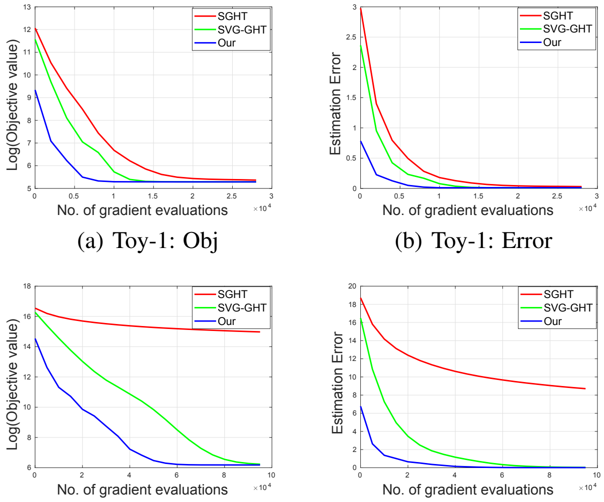
Throughout the experiments, we compare it with two state-of-the-art stochastic methods. They are SGHT [Nguyen *et al.*, 2014] and SVR-GHT [Li *et al.*, 2016]. Specifically,

- SGHT [Nguyen *et al.*, 2014]: It employs stochastic gradient to update the model parameter and then performs Hard-Thresholding on the obtained model parameter.
- SVR-GHT [Li *et al.*, 2016]: This method adopts the variance reduced gradient to update model parameter, accelerating the converging speed.

All of these methods belong to the stochastic method so that they are suitable for large-scale problems. Additionally, we set $L = 10$, $M = 10$, $|\mathcal{B}| = 10$, and $|\mathcal{B}'| = 50$. The step length of each method is chosen to achieve the best performance.

4.1 Synthetic Data

In this experiment, we focus on the sparse linear regression problem, just as shown in Eq. (2). For the synthetic data, each row of the design matrix $X \in \mathbb{R}^{n\times d}$ is independently generated from a multivariate Gaussian distribution $N(\mathbf{0}, \mathbf{\Sigma})$ where $\mathbf{\Sigma} \in \mathbb{R}^{d\times d}$. For the sparse regression coefficient $\mathbf{w}^*$, the nonzero entries are independently generated from a uniform distribution in $[-1, 1]$. The response vector is constructed by $\mathbf{y} = X\mathbf{w}^* + \boldsymbol{\epsilon}$, where the noise $\boldsymbol{\epsilon}$ is generated from a Gaussian distribution $N(\mathbf{0}, \sigma^2\mathbf{I})$, and we set $\sigma^2 = 0.01$. With these settings, we construct two synthetic datasets. Toy-1 is with $n = 20000$, $d = 2000$, $s^* = 100$, $s = 200$, $\mathbf{\Sigma} = \mathbf{I}$. Toy-2 is with $n = 50000$, $d = 5000$, $s^* = 500$, $s = 1000$, and diagonal entries of the covariance matrix $\mathbf{\Sigma}$ are set as 1, the other entries are set as 0.1.



(a) Toy-1: Obj

(b) Toy-1: Error

(c) Toy-2: Obj

(d) Toy-2: Error

Figure 1: The objective value and estimation error of the model parameter about two synthetic datasets.

In Figure 1, we show the logarithm of the objective function value and the estimation error with respect to the number of gradient evaluations. From Figure 1, we can find that our method converges faster than the state-of-the-art methods in terms of objective function value and the estimation error. In particular, although all these methods are theoretically shown with linear convergence rate for the sparse linear regression problem [Li *et al.*, 2016], our method and SVR-GHT have better performance than SGHT due to the incorporation of the variance reduction technique. Furthermore, our method incorporates the second-order information so that it converges faster than SVR-GHT, which is consistent with the motivation of our method.

4.2 Real-World Data

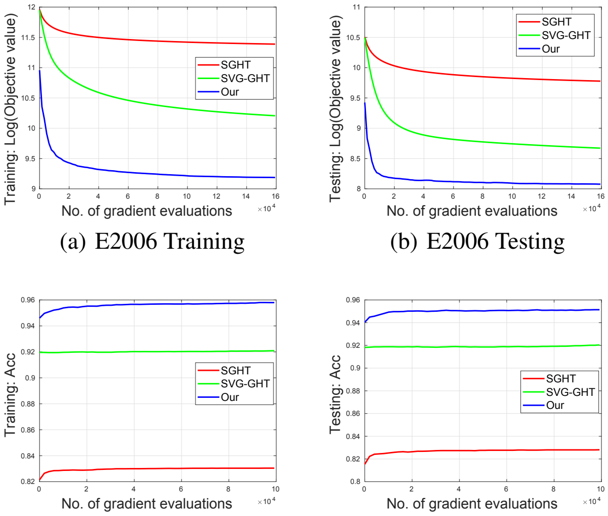
In this experiment, we will evaluate the performance of the sparse linear regression defined in Eq. (2) and the sparse logistic regression on the real-world data. Specifically, the sparse logistic regression is defined as follows:

$$\min \mathcal{F}(\mathbf{w}) = \frac{1}{n}\sum_{i=1}^{n}(-y_i\mathbf{x}_i^T\mathbf{w} + \log(1+\exp(\mathbf{x}_i^T\mathbf{w})))$$
$$s.t. \|\mathbf{w}\|_0 \leq s \;, \tag{33}$$

where $y_i \in \{0, 1\}$ is the class label.

For the sparse linear regression model, we evaluate its performance on E2006-TFIDF dataset , which includes 16,087 training data points, 3,308 testing data points, and 150,360 features. Instead of using all features, we randomly select 20,000 features for both training set and testing set. Additionally, the sparsity level s is set as 2000. For the sparse logistic regression model, we evaluate its classification performance on the RCV1-Binary dataset , which includes 20,242 training samples and 677,399 testing samples from two classes. Additionally, it has 47,236 features totally. Here, we choose 5000 samples from each class of the testing samples as our testing set. At last, we set the sparsity level s as 500. Note that both datasets are available at the LIBSVM website [1].

In Figure 2(a) and 2(b), we show the logarithm of the sparse linear regression's objective function value on the training set and testing set. We can find the similar result with the synthetic dataset. In particular, our method converges faster than the other state-of-the-art methods and has better performance on the testing set, which further confirms the effectiveness of our proposed method. In Figure 2(c) and 2(d), we show the classification accuracy of the sparse logistic regression on both training set and testing set. We can find that our method achieves the best classification result consistently during iterations. Additionally, the improvement is significant. In conclusion, our method converges fast and the obtained solution generalizes well.



(a) E2006 Training

(b) E2006 Testing

(c) RCV1 Traing

(d) RCV1 Testing

Figure 2: (a-b) show the logarithm of the sparse linear regression's objective function value. (c-d) show the classification accuracy of the sparse logistic regression.

5 Conclusion

In this paper, we propose a stochastic L-BFGS method for solving large-scale nonconvex sparse learning problems. By theoretical analysis, the proposed method has shown a linear convergence rate for this kind of nonconvex problems. Meanwhile, it can guarantee to converge the underlying true model parameters. The extensive experiments have verified the efficiency of the proposed method. Thus, it can be applied to the real-world nonconvex large-scale problems.

Acknowledgements

This work was partially supported by the following grants: NSF-IIS 1302675, NSF-IIS 1344152, NSF-DBI 1356628, NSF-IIS 1619308, NSF-IIS 1633753, NIH R01 AG049371.

[1] https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets

References

[Banerjee *et al.*, 2008] Onureena Banerjee, Laurent El Ghaoui, and Alexandre dAspremont. Model selection through sparse maximum likelihood estimation for multivariate gaussian or binary data. *Journal of Machine learning research*, 9(Mar):485–516, 2008.

[Blumensath and Davies, 2009] Thomas Blumensath and Mike E Davies. Iterative hard thresholding for compressed sensing. *Applied and computational harmonic analysis*, 27(3):265–274, 2009.

[Byrd *et al.*, 2016] Richard H Byrd, Samantha L Hansen, Jorge Nocedal, and Yoram Singer. A stochastic quasi-newton method for large-scale optimization. *SIAM Journal on Optimization*, 26(2):1008–1031, 2016.

[Chen and Gu, 2016] Jinghui Chen and Quanquan Gu. Accelerated stochastic block coordinate gradient descent for sparsity constrained nonconvex optimization. In *Conference on Uncertainty in Artificial Intelligence*, 2016.

[Friedman *et al.*, 2008] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, 9(3):432–441, 2008.

[Gao *et al.*, 2015] Hongchang Gao, Lin Yan, Weidong Cai, and Heng Huang. Anatomical annotations for drosophila gene expression patterns via multi-dimensional visual descriptors integration: Multi-dimensional feature learning. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 339–348. ACM, 2015.

[Gao *et al.*, 2017] Hongchang Gao, Feiping Nie, and Heng Huang. Local centroids structured non-negative matrix factorization. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

[Gower *et al.*, 2016] Robert Gower, Donald Goldfarb, and Peter Richtárik. Stochastic block bfgs: squeezing more curvature out of data. In *International Conference on Machine Learning*, pages 1869–1878, 2016.

[Johnson and Zhang, 2013] Rie Johnson and Tong Zhang. Accelerating stochastic gradient descent using predictive variance reduction. In *Advances in Neural Information Processing Systems*, pages 315–323, 2013.

[Lee *et al.*, 2006] Su-In Lee, Honglak Lee, Pieter Abbeel, and Andrew Y Ng. Efficient l˜ 1 regularized logistic regression. In *AAAI*, 2006.

[Li *et al.*, 2016] Xingguo Li, Raman Arora, Han Liu, Jarvis Haupt, and Tuo Zhao. Nonconvex sparse learning via stochastic optimization with progressive variance reduction. *arXiv preprint arXiv:1605.02711*, 2016.

[Mallat and Zhang, 1993] Stéphane G Mallat and Zhifeng Zhang. Matching pursuits with time-frequency dictionaries. *IEEE Transactions on signal processing*, 41(12):3397–3415, 1993.

[Moritz *et al.*, 2016] Philipp Moritz, Robert Nishihara, and Michael Jordan. A linearly-convergent stochastic l-bfgs algorithm. In *Artificial Intelligence and Statistics*, pages 249–258, 2016.

[Needell and Tropp, 2009] Deanna Needell and Joel A Tropp. Cosamp: Iterative signal recovery from incomplete and inaccurate samples. *Applied and Computational Harmonic Analysis*, 26(3):301–321, 2009.

[Nguyen *et al.*, 2014] Nam Nguyen, Deanna Needell, and Tina Woolf. Linear convergence of stochastic iterative greedy algorithms with sparse constraints. *arXiv preprint arXiv:1407.0088*, 2014.

[Nocedal and Wright, 2006] Jorge Nocedal and Stephen J Wright. Numerical optimization (2nd edition), 2006.

[Tibshirani, 1996] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 267–288, 1996.

[Tropp and Gilbert, 2007] Joel A Tropp and Anna C Gilbert. Signal recovery from random measurements via orthogonal matching pursuit. *IEEE Transactions on information theory*, 53(12):4655–4666, 2007.

[Van de Geer, 2008] Sara A Van de Geer. High-dimensional generalized linear models and the lasso. *The Annals of Statistics*, pages 614–645, 2008.

[Yuan and Lin, 2006] Ming Yuan and Yi Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(1):49–67, 2006.

[Yuan and Liu, 2014] Xiao-Tong Yuan and Qingshan Liu. Newton greedy pursuit: A quadratic approximation method for sparsity-constrained optimization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4122–4129, 2014.

[Yuan *et al.*, 2014] Xiaotong Yuan, Ping Li, and Tong Zhang. Gradient hard thresholding pursuit for sparsity-constrained optimization. In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, pages 127–135, 2014.

[Zhang, 2011] Tong Zhang. Adaptive forward-backward greedy algorithm for learning sparse representations. *IEEE transactions on information theory*, 57(7):4689–4708, 2011.

[Zhao *et al.*, 2017] Renbo Zhao, William B Haskell, and Vincent YF Tan. Stochastic l-bfgs revisited: Improved convergence rates and practical acceleration strategies. *arXiv preprint arXiv:1704.00116*, 2017.